Janne Liimatainen

Avoimen lähdekoodin alustojen sopivuus älykotiin

Tietotekniikan pro gradu -tutkielma

24. marraskuuta 2017

Jyväskylän yliopisto

Tietotekniikan laitos

Tekijä: Janne Liimatainen

Yhteystiedot: janne.s.liimatainen@jyu.fi

Ohjaaja: Timo Hämäläinen

Työn nimi: Avoimen lähdekoodin alustojen sopivuus älykotiin

Title in English: Suitablility of open source gateways in a Smart Home

Työ: Pro gradu -tutkielma

Suuntautumisvaihtoehto: Tietoliikenne, kyberturvallisuus?

Sivumäärä: 60+0

Tiivistelmä: Internet of Things kasvaa jatkuvasti ja sen eräs suurimmista sovelluskohteista on älykoti. IoT:n ongelmana on kuitenkin ollut keskimäärin erittäin huono tietoturva ja tämä on heijastunut myös älykoteihin esimerkiksi hakkeroituina kameroina ja itkuhälyttiminä. Toinen, älykodissa erityisen suuri ongelma on kerätyn tiedon yksityisyys. Tässä tutkielmassa kartoitetaan, kuinka hyvin avoimen lähdekoodin ohjelmistot soveltuvat toimivan ja turvallisen älykodin toteuttamiseen. Turvallisuuden osalta painopisteenä on erityisesti etäyhteys. Tutkimus toteutettiin konstruktiivisena tutkimuksena. Se ei käsittele yksittäisiä ohjelmistoja kovin syvällisesti, vaan sen tarkoituksena oli saada kattavahko yleiskuva tutkituista ohjelmistoista, minkä perusteella jatkotutkimusta voisi suunnitella.

Avainsanat: älykoti, kyberturvallisuus, Internet of Things

Abstract: Internet of Things is constantly growing and one if its biggest applications is Smart Home. IoT however has generally had very low security and this has reflected also on Smart Homes, for example in hacked cameras and baby monitors. Another big problem, especially in Smart Homes, is the privacy of the collected data. This thesis examines the feasibility of open source gateway softwares in implementing a working and secure smart home. Main focus regarding security is on remote access. The research was conducted as a constructive research. It does not delve deeply into any single software, rather its purpose was to get a broad overview of the investigated software, to guide follow-up research.

Keywords: Smart home, Cyber Security, Internet of Things

Termiluettelo

Älykoti	Älykodilla tarkoitetaan kotia, jossa erilaiset sensorit ja aktuaattorit huolehtivat käyttäjän asettamista toimenpiteistä, kuten esimerkiksi sammuttavat valoja tai lämmittävät taloa vain silloin, kun joku on kotona.
Internet of Things, IoT	Kaikkiin mahdollisiin laitteisiin ja tavaroihin lisätään mahdollisuus yhdistyä verkkoon tai ainakin niihin lisätään esimerkiksi RFID-tagi, jolloin niitä voidaan helposti seurata.

Sisältö

1 Johdanto

Maailma digitalisoituu jatkuvasti enemmän ja eräs tärkeimmistä sovelluksista tälle on älykodit. Älykodit voivat esimerkiksi säästää energiaa, lisätä asumismukavuutta ja turvallisuutta, sekä sallia vanhusten asua kotona pidempään. Varsinkin valmiisiin rakennuksiin kuluttajat toteuttavat älykodin useimmiten erikseen ostettavilla laitteilla, jotka voidaan useimmiten laskea kuuluvan IoT:seen, eli Internet of Thingsiin. Ne ovat käytännössä tavallisia laitteita (kuten hämäräkytkimiä, lamppuja, lämmityslaitteita, lukkoja, jne), joihin on lisätty sekä laskentakapasiteettia että mahdollisuus liittyä verkkoon. Nämä molemmat seikat tarkoittavat että näihin laitteisiin kohdistuu samoja tietoturvauhkia kuin perinteisempiinkin tietokoneisiin. Tietoturvauhkiin kuuluu lisäksi myös käyttäjien datan yksityisyys, sillä laitteet voivat kerätä hyvinkin sensitiivistä dataa, eikä käyttäjällä välttämättä ole valtaa vaikuttaa, mihin se menee ja miten sitä käytetään. Näiden kahden seikan lisäksi erityisesti kaupallisten älykotiratkaisujen heikkous on se, että ne useimmiten tukevat vain valmistajan omia laitteita ja samalla todennäköisesti vain yhtä tai kahta yhteysprotokollaa. Näiden puutteiden takia on kehitetty useita avoimen lähdekoodin kotiautomaatio-ohjelmistoja, jotka useimmiten pyrkivät varmistamaan tietoturvallisuuden, datan pysymisen käyttäjällä, sekä eri valmistajien laitteiden toimimisen keskenään.

Tämän tutkielman tarkoituksena oli kartoittaa, kuinka hyvin toimivan ja turvallisen älykodin rakentaminen onnistuu avoimen lähdekoodin ohjelmistoilla. Koska älykodissa voidaan käyttää erilaisia arkkitehtuureja, oli ensimmäinen tehtävä selvittää mikä näistä on turvallisin. Loppujen lopuksi niissä kaikissa on paljon samoja piirteitä, mutta tärkeimmät erot löytyvät siitä, missä laskenta tapahtuu, ts. pilvessä vai lokaalisti, ja jos lokaalisti, niin missä laitteessa. Lopulta päädyttiin gateway-arkkitehtuuriin, jossa laskenta, hallinta ja mahdollinen Internet-yhteys tapahtuvat yhdessä laitteessa. Tämä vertautuu hyvin myös kaupallisiin tuotteisiin, jotka useimmiten ovat samaa arkkitehtuuria, mutta vaativat pilven käyttöä etäyhteyteen ja yleensä myös jonkintasoiseen laskentaan.

Älykodin käyttö onnistuu pelkästään lokaalistikin, mutta useimmiten käyttäjät haluavat jonkinlaisen etähallintamahdollisuuden kodin ulkopuolelta. Tämä tarkoittaa, että älykodin gateway on tavalla tai toisella näkyvissä Internetiin, jolloin sen turvallisuudesta huolehtiminen

on erittäin tärkeää. Kaupalliset tuotteet lähes poikkeuksetta hoitavat etäyhteyden pilvipalvelun kautta, mikä luo huolta sekä pilven turvallisuudesta että käyttäjien tiedon yksityisyydestä. Erilaisia etäyhteystapoja on useita ja tutkielmassa niiden turvallisuutta vertaillaan myös hieman.

Tutkielma päätettiin toteuttaa konstruktiivisena tutkimuksena. Aluksi tarkasteltiin tarjolla olevia avoimen lähdekoodin älykoti-gateway-ohjelmistoja ja valittiin tarkempaan tutkimukseen niistä seitsemän. Nämä asennettiin ja niitä käyttämällä luotiin älykotiverkko, jossa oli muutama laite ja näiden laitteiden välillä jotain automaatiota. Tällä osuudella kartoitettiin ohjelmistojen käytön toimivuus ja käyttäjäystävällisyys. Samalla kartoitettiin ohjelmistojen turvallisuutta, erityisesti oletusasetusten suhteen. Erityisesti tutkittiin etäyhteyden ohjeita ja toteutusta, sekä autentikointia, sillä nämä ovat tärkeimpiä tekijöitä turvallisuuden kannalta. Myös datan säilytys- ja keräämisasetuksiin kiinnitettiin huomiota. Tutkimuskysymyksenä olivat ’Miten hyvin avoimen lähdekoodin gateway-ohjelmistot soveltuvat älykotiin, erityisesti turvallisuuden suhteen? Miten ne vertautuvat kaupallisiin tuotteisiin?’

Kappale 2 on teoriaosuus: se esittelee ja selittää älykodin ja IoT:n käsitteitä sekä käsittelee niiden turvallisuutta, uhkia ja haavoittuvuuksia. Siinä myös esitellään erilaiset älykodin arkkitehtuurit ja perustellaan, miksi tutkielma käsittelee juuri gateway-arkkitehtuuria. Lisäksi siinä esitellään erilaiset älykodin etäyhteystavat. Kappaleessa 3 esitellään tutkimustapa ja -laitteisto, sekä tutkittavaksi valitut ja pois jätetyt avoimen lähdekoodin ohjelmistot ja perustelut valinnoille. Lisäksi siinä on esitelty jokaisen valitun ohjelmiston tutkimus ja tulokset. Kappale 4 kokoaa tutkimusten tulokset yhteen ja pohtii niitä. Siinä myös annan joitain suosituksia ohjelmistojen suhteen. Kappale 5 on yhteenveto koko tutkielmasta, sen syistä, etenemisestä ja tuloksista.

2 Älykoti

Älykodiksi kutsutaan asuntoa, jossa erilaisten sensorien ja aktuaattorien avulla vastataan käyttäjien tarpeisiin. Bugeja, Jacobsson ja Davidsson (2016) jakavat nämä tarpeet eli älykodin sovellusalueet neljään kategoriaan: **viihde, energia, turvallisuus, ja terveydenhuolto.** Ne tarkoittavat, että älykoti

- pyrkii maksimoimaan käyttäjien mukavuustason ja tekemään heidän elämästään mahdollisimman helppoa, esimerkiksi sytyttämällä ja sammuttamalla valoja kulloisenkin aktiviteetin mukaan.
- pyrkii optimoimaan energiankäytön ja sen hallinnan, esimerkiksi säätämällä lämmitystä pienemmälle, kun kukaan ei ole kotona.
- tarjoaa uhkien monitorointia, havainnointia ja kontrollointia, esimerkiksi tarjoamalla mahdollisuuden katsoa kotiin asennetun kameran kuvaa etänä tai ilmoittamalla automaattisesti poliisille, jos talossa on liikettä omistajien ollessa lomalla.
- tarjoaa mobiiliterveyspalveluja ja fitness-tukea, esimerkiksi lääkärille etänä lähetettävät terveysmittaukset tai unen laadun seuranta sängyssä olevilla sensoreilla.

Älykoti voi myös mahdollistaa esimerkiksi vanhusten yksin asumisen kauemmin kuin 'normaali' asunto ja tehdä siitä turvallisempaa vaikkapa seuraamalla, että asukas ei ole ollut paikallaan liian kauaa ja ilmoittamalla mahdollisista vaaratilanteista lähiomaisille tai viranomaisille.

Wilson, Hargreaves ja Hauxwell-Baldwin (2015) löysivät tekemänsä kirjallisuuskatsauksen pohjalta yhteensä yhdeksän eri teemaa liittyen älykotiin. Nämä on jaettu kolmeen eri osa-alueeseen:

1. Älykodin näkemys, eli mitä älykoti on
2. Älykodin käyttäjät ja käyttö
3. Älykodin haasteet

Seuraavassa on eritelty tämän tutkielman kannalta olennaiset seikat.

Ensimmäinen näkemys älykodista on funktionaalinen: älykoti tekee elämisestä parempaa. Se lisää esimerkiksi mukavuutta, turvallisuutta ja energiansäästöä, sekä esimerkiksi vanhuksilla helpottaa yksin elämistä. Funktionaalisuus voidaan jakaa kolmeen osaan: elämäntyylin tukeminen, energiankäytön hallitseminen ja turvallisuus. Ylipäätään älykodin tarkoituksena on parantaa vanhoja laitteita ja vanhaa kotia, ei luoda täysin uutta. Pohjimmiltaan laitteet itsessään eivät ole mitenkään älykkäitä, mutta koska ne muodostavat verkon, ne pystyvät jakamaan informaatiota ja toimimaan sen mukaan, luoden näin 'älyä': pelkkä lamppu ja hämäräkytkin erillään eivät pysty lisäämään käyttäjän asumismukavuutta nykyisestä, mutta yhdistettynä kyllä. Schiefer (2015) määrittelee älykotilaitteen 'laitteeksi, jonka pääasiallista toiminnallisuutta laajennetaan verkkokyvykkyydellä ja näin luodaan uusi laite.'

Toinen näkemys on instrumentaalinen: älykoti säästää energiaa esimerkiksi erilaisten älymittareiden ja älypistorasioiden avulla. Käyttäjät voivat säästää rahaa seuraamalla reaaliaikaista tietoa sähkön hinnasta. Kolmas näkemys on sosio-tekninen: älykotia pidetään seuraavana askeleena teknologian ja yhteiskunnan suhteessa.

Wilson, Hargreaves ja Hauxwell-Baldwin (2015) mainitsemista älykodin käyttäjiin liittyvistä haasteista ensimmäinen on heille sopivien teknologioiden kehitys:

- Sensorien täytyy luotettavasti mitata asunnon tapahtumia ja algoritmien täytyy osata näistä mittaustuloksista tehdä oikeita päätöksiä.
- Älykodissa käytettyjen eri teknologioiden yhteensopivuus ja standardointi.
- Laitteiden luotettavuus ja hallittavuus.

Jos eri laitteet eivät toimi keskenään tai edes yksinään, on tuloksena vähintään käyttäjän ärsyyntyminen, pahimmillaan esimerkiksi vanhuksen sairaskohtaus ja kuolema. Tärkeää on myös se, että laitteet tulevat toimimaan tulevaisuudessakin.

Toisena haasteena on sopivien teknologioiden suunnittelu, mikä liittyy erityisesti sekä turvallisuuteen, yksityisyyteen ja luottamukseen, että käyttäjäystävällisyyteen. Käyttäjälle tulisi antaa mahdollisimman vapaat kädet päättää älykodin käytöstä ja datan keruusta. Lisäksi kerättävän tiedon suhteen pitäisi olla avoin. Kolmas haaste liittyy älykotiteknologioiden integroimiseen jokapäiväiseen elämään erilaisissa kodeissa. Myös tähän auttaa laitteiden käytön vapaus.

2.1 Älykodin turvallisuus

Vaikka älykoti tarjoaa useita etuja, on sillä myös haittoja ja uhkia, sekä teknisellä että sosiaalisella puolella. Suurimpina älykodin yleistymistä haittaavina tekijöinä ovat turvallisuuskysymykset ja kerätyn datan luottamuksellisuus sekä käyttöönoton ja käyttämisen vaikeus. Tässä tutkielmassa keskitytään tekniseen puoleen, erityisesti tietoturvaan ja siihen liittyviin seikkoihin.

Plachkinova, Vo ja Alluhaidan (2016) löysivät systemaattisella kirjallisuuskatsauksellaan viisi trendiä liittyen älykotien turvallisuuteen ja yksityisyyteen:

1. **Etänä tapahtuvien turvallisuusmurtojen mahdollisuus**: Esimerkkinä älykodille ominaisista turvallisuusongelmista mainitaan muun muassa asiantuntevan ylläpidon puute, vierailijoiden monimuotoisuus, sekä useampi ylläpitäjä, joilla on mahdollisesti erilaiset tarpeet. Nämä seikat voivat aiheuttaa sen, että käyttöoikeudet eivät ole turvallisesti asetettuja.
2. **Itse laitteiden riskit**: Erityisesti mainitaan älypuhelimet ja miten niiden turvallisuus on tärkeää, koska useimmiten niillä hallitaan älykotia etänä. Mobiilisovellukset saavat hyvin usein liikaa valtuuksia ja niitä tulisikin rajoittaa.
3. **Yksityisyysloukkaukset**: Laitteet keräävät valtavasti dataa, joka usein päätyy valmistajalle tai kolmansille osapuolille ilman, että käyttäjä tietää mitä tämä data sisältää tai että hän voisi sen lähettämistä estää. Kerättävä data voi olla enemmän tai vähemmän sensitiivistä: medikaalinen tieto enemmän, sisälämpötila vähemmän. Samoin jos kotiverkossa on turvaton laite, voidaan siihen murtautumalla mahdollisesti päästä käsiksi verkon koko liikenteeseen.
4. **Infrastruktuurin haavoittuvuudet**: Esimerkiksi IoT-protokollissa, kuten Zigbeessä, on haavoittuvuuksia, ja IoT-laitteiden pieni koko vaikeuttaa niiden kryptograafisia ominaisuuksia.
5. **Digitaalisen rikostekniikan haasteet**: Tämä liittyy digitaalisten rikosten tutkimiseen ja niiden ongelmiin älykodissa, eli esimerkiksi onko data luotettavaa, miten erotellaan kenen dataa se on, missä data on...

Turvallisuus liittyy myös käyttöönoton ja käyttämisen helppouteen. Jos älykodin laitteiden

asentaminen turvallisesti on yhtä helppoa tai jopa vaikeampaa kuin niiden asentaminen turvattomasti, on helppoa tehdä koko verkosta turvaton. Tämä pätee jo yhdenkin laitteen kohdalla, sillä jos kotiverkossa on turvaton laite, voidaan siihen murtautumalla useimmiten sekä päästä käsiksi verkon koko liikenteeseen sensorien datan osalta, että päästä mahdollisesti säätämään aktuaattoreita, esimerkiksi lämmitystä. Samoin jos käyttäminen vaikeutuu turvallisuuden lisääntyessä ja toisinpäin, on todennäköistä että käyttäjä suosii helppokäyttöisyyttä. Täten turvallisen asentamisen ja käyttämisen tulisi optimaalisessa tilanteessa olla helpompaa kuin turvattoman. Lisäksi sen tulisi olla oletuksena.

Schiefer (2015) mainitsevat, että suuria turvallisuudenrikkomistapauksia älykoteihin ei ole tiedossa. Syyksi he esittävät, että käytetyt laitteet ovat hyvin erilaisia ja tietyn valmistajan tiettyä laitetta on käytössä suhteellisen vähän, joten hyökkäys ei olisi kovinkaan laaja eikä näin ollen kiinnosta hyökkääjiä. Kuitenkin on olemassa nouseva trendi mahdollistaa eri valmistajien laitteiden yhdistäminen toisiinsa, mikä lisää niiden käyttäjiä ja hyökkääjien kiinnostusta.

2.1.1 Turvallisuusuhkien syyt

Bugeja, Jacobsson ja Davidsson (2016) mainitsevat, että älykoti sisältää sensitiivistä dataa ja laitteita, jotka teoriassa mahdollistavat videon tai äänen salakatselun/-kuuntelun, minkä takia turvallisuus on erittäin tärkeää. He mainitsevat kuusi ongelmaa älykodin turvallisuuteen liittyen: laiteongelmista laitteiden vähäiset resurssit, 'päätön' eli käyttöliittymätön luonne, sekä peukalointiresistanssin puute; kommunikointiongelmista heterogeeniset protokollat sekä dynaamiset luonteet; sekä huolto-ongelmista pitkäikäisyysodotukset.

Lin ja Bergmann (2016) esittävät haavoittuvuuksia älykodista:

1. **Laitteiden (mahdollinen) Internetiin kytkeytyneisyys**: Mistä tahansa päin maailmaa voidaan hyökätä älykotiin Internetin välityksellä.
2. **Fyysinen**: Talon ulkopuolelta voidaan fyysisesti liittyä sekä langattomiin että sähköverkkoihin, vaikka talo itsessään olisi lukittu.
3. **Laitteiden rajoitetut resurssit**: Laitteet ovat usein pieniä ja tehottomia ja toimivat paristoilla. Tästä johtuen niillä ei ole laskentakapasiteettia monimutkaisiin salausmene-

telmiin tai muihinkaan keinoihin estää väärinkäyttöä. Täten joudutaan aina tekemään kompromisseja laitteen koon, akunkeston, hinnan, sekä turvallisuuden välillä.

4. **Järjestelmien heterogeenisyys**: Laitteita on eri valmistajilta ja ne käyttävät eri teknologioita ja niiden dokumentaation ja laitteiston tiedot ovat vähäisiä tai puuttuvat kokonaan.
5. **Laitteiden päivitysmahdollisuuden puuttuminen**: Usein laitteita ei ole mahdollista päivittää tai valmistaja ei tarjoa päivityksiä, sillä parin euron laitteen päivittäminen ei maksane itseään takaisin.
6. **Standardien hidas käyttöönotto**: Useimmat laitteet eivät noudata mitään turvallisuusstandardeja.
7. **Älykotien kompleksisiin verkkoihin erikoistuneiden turvallisuusasiantuntijoiden puute.**

Lisäksi turvallisuuteen vaikuttavat ainakin seuraavat seikat:

- Valmistajat usein haluavat laitteensa nopeasti ja halvalla markkinoille eivätkä välitä turvallisuudesta kovinkaan paljoa.
- Laitteet ovat useimmiten langattomia, jolloin niiden liikenteeseen päästään käsiksi helpommin kuin langallisesti, sekä salakuuntelun että viestien lähettämisen suhteen. Ensimmäiset älykodit olivat yleensä langallisesti toteutettuja ja nykyäänkin se on mahdollista. Hyvänä puolena tässä ratkaisussa on tiedonsiirron luotettavuus; huonoina puolina asentamisen vaikeus sekä kiinteys. Lisäksi ne ovat usein kalliimpia kuin halvat pienet IoT-laitteet.

Laitteiden langattomuuden takia nousee esille myös kysymys niiden yhteysteknologian turvallisuudesta. WiFi:n turvallisuudesta on monia tutkimuksia ja ensimmäiset ja nykyäänkin tehokkaammat IoT-laitteet käyttävät sitä. Pienille ja teholtaan rajoitetuille laitteille se ei kuitenkaan sovi. Näille on kehitetty useita eri yhteysteknologioita, joista älykodeissa tunnetuimmat ovat Zigbee ja Z-Wave. Muita IoT-teknologioita ovat muun muassa 6LoWPAN ja LoRaWan. Näistä 6LoWPAN on kehitetty toimimaan IP-protokollan kanssa, jolloin laitteet ovat helposti yhdistettävissä Internetiin ilman välilaitteita, kuten gatewayta. Tämä on myös turvallisuusriski, sillä tällöin niihin pystytään hyökkäämään Internetin yli; muilla protokollilla hyökkäysvektori jää pienemmäksi.

Vaikka älykodin liikenne olisi salattua, voidaan sitä kaappaamalla silti saada paljonkin tietoa. Copos ym. (2016) esittävät menetelmän, jolla pystytään älykodin langattomasta liikenteestä päättelemään, onko ketään kotona ja eri puolilta asuntoa kerätty liikennedata voisi kertoa, missä huoneissa asukkaat ovat kulloinkin. Tarkimmat arvaukset voidaan tehdä, kun sekä aika- että paikkatiedot ovat liikenteestä selvillä, vaikka sisältö ei olisikaan. Tutkimuksessa he käyttivät laitteiden tilan päättelyyn paketin kohde-IP-osoitteita, lähetettyjen tavujen määrää, sekä pakettien 'purskeisuutta'. Tutkimus tehtiin salatuille paketeille, mutta salaamattomalle WiFi-liikenteelle. Sen salaus vaikeuttaisi, muttei kokonaan poistaisi, tällaista hyökkäystä. Liikenteen analysoinnin vaikeuttamiseen he esittävät esimerkiksi pakettien pehmustamisen (padding) samankokoisiksi, pakettien lähettämisen aina samalle serverille, sekä sen, että silloin tällöin lähetettäisiin valepaketteja. Purskeisuuteen nojaavia päättelyjä nämä eivät estä. Tällaisen tiedon salassapito lasketaan kontekstipohjaisen yksityisyyden alle (Islam, Shen ja Wang 2012).

Ulkopuolinen hyökkääjä ei pääse sisältöön käsiksi fyysisesti kaukaa, lähistöllä ollessa hän pystyy kyllä langattoman liikenteen kaappaamaan. Jos liikenne on salattua, ei sekään onnistu. Sisäverkkoon päässyt hyökkääjä, eli käytännössä laite, joka on saatu haltuun käyttäjän tietämättä, voi useimmiten päästä sisältöön käsiksi. Jos tämä laite on älykodin verkon reitityksessä sellainen, että se joutuu lähettämään eteenpäin viestejä, saa se haltuunsa myös niiden sisällön, sillä se tietää verkon avaimen. Samalla lailla se saa haltuunsa myös muut kantaman sisällä olevat viestit. Tämä voidaan estää end-to-end-salauksella, jolloin viesti on salattu avaimella, jonka tietävät vain keskustelevat laitteet, eivät välissä olevat reitittävät laitteet. Tällaisen tiedon salassapito lasketaan datapohjaisen yksityisyyden alle (Islam, Shen ja Wang 2012).

Siinä tapauksessa, että laitteet ovat 'vapaita' sisäverkossaan toisiinsa nähden, eikä autentikointia tai salausta ole, on koko älykotiverkon turvallisuus käytännössä kiinni sen heikoimmasta lenkistä. Tämän takia kaikkien laitteiden tulisi olla turvallisia, riippuen tietysti käytetystä arkkitehtuurista. Jos mahdollista, sallitaan esimerkiksi hämäräkytkimelle ainoaksi toiminnoksi liikettä-viestin lähetys. Mitään muita siltä tulevia viestejä ei gateway edes käsittelisi. Pienissä laitteissa ei kuitenkaan usein ole mahdollisuutta tällaista tehdä, eli laitteet itsessään ottaisivat todennäköisesti vastaan kaikki viestit. Useat aikaisimmista IoT-laitteista

luottivatkin siihen, että hyökkääjä ei pääse niiden kanssa samaan sisäverkkoon. Jos tällainen laite on suoraan näkyvissä Internetiin, se on täysin turvaton.

2.1.2 Älykodin tekniset vaatimukset

Korkeamman tason vaatimuksista voidaan määritellä esimerkiksi seuraavassa esitetyt.

Zhenhua (2016) määrittelee älykodin verkolle neljä funktiota:

1. Informaatiopalvelut, eli kotigatewayn kautta pääsee verkon palveluihin käsiksi; esimerkiksi selaamaan webbiä, katsomaan digi-tv:tä ja soittamaan videopuheluita.
2. Keskitetty kodin laitteiden hallinta; esimerkiksi valojen ja sisäympäristön säätö.
3. Kolmen mittarin tallennusjärjestelmä; kerää dataa veden, sähkön ja kaasun kulutuksesta ja lähettää ne kutakin hoitavalle yritykselle.
4. Turvallisuus ja tulipalohallinta; molempia varten asennetaan laitteistoja.

Zhou, Huang ja Zhao (2013) määrittelevät älykotijärjestelmälle kolme vaatimusta: Ensinnäkin sen tulee varmistaa asumisturvallisuus estämällä varkaudet, hälyttämällä tulipaloista jne. Toiseksi sen pitää hallita ja kontrolloida energiankäyttöä, saada sähkö- ja vesidata ja antaa käskyjä energiansäästöstrategian mukaisesti. Kolmanneksi sen tulee tukea eri kommunikaatio- ja applikaatio-protokollia.

Datan ja tiedonsiirron käsittelyn puolesta vaatimuksia voidaan ajatella esimerkiksi seuraavassa esitettyjen tasojen kautta.

IoT:n ja kyberturvallisuuden yleensäkin, tärkeimpinä teemoina pidetään usein CIA-kolmiota: Confidentiality, Integrity, Availability. *Confidentiality eli salassapito* viittaa siihen, että tiedon tulee pysyä salassa kaikilta muilta paitsi niiltä, joilla on siihen oikeus. *Integrity eli eheys* viittaa siihen, että tiedon pitää pysyä muuttumattomana ja eheänä elinkaarensa ajan. *Availability eli saatavuus* viittaa siihen, että tiedon tulee olla saatavilla kun sitä tarvitaan. Optimaalinen tilanne on se, että näistä saavutetaan kaikki, mutta joissain tapauksissa riittää yksi tai kaksi.

Lin ja Bergmann (2016) esittävät kolmena pääteemana *salassapidon, autentikaation ja pääsyn* (Confidentiality, Authentication ja Access). *Salassapito* on sama kuin CIA-kolmiossa.

Autentikaatio lisää *eheyteen* sen, että pitää pystyä verifioimaan että datan on lähettänyt se, kenen väitetään sen lähettäneen. *Pääsy* lisää *saatavuuteen* sen, että dataan ei pääse käsiksi kuin siihen oikeutetut. Tämä on osittain päällekkäin *salassapidon* kanssa.

Lin ja Bergmann (2016) esittävät joitain uhkia CAA-kolmionsa kautta: *Salassapidon* turvattomuus johtaa siihen, että dataa päätyy vääriin käsiin; esimerkiksi lämpötilatiedoista voi päätellä onko joku kotona. *Autentikaation* turvattomuus johtaa siihen, että kotiautomaatiojärjestelmälle voidaan antaa falsifioitua dataa ja näin saada esimerkiksi ovet aukeamaan oletetun tulipalon takia. *Pääsyn* turvattomuus johtaa pahimmillaan siihen, että älykodin gatewayhin päästään admin-oikeuksilla sisään. Jo pelkillä käyttäjän oikeuksilla voidaan saada DoS-hyökkäys aikaan ja esimerkiksi kuluttaa laitteiden paristoja loppuun.

Islam, Shen ja Wang (2012) mainitsevat turvallisuustavoitteina *salassapidon, eheyden, tuoreuden, saatavuuden ja autenttisuuden* (Confidentiality, Integrity, Freshness, Availability ja Authenticity). Näistä uutena *tuoreus*, joka tarkoittaa että datan ja salausavainten tulisi olla 'tuoreita'. Hyökkääjä ei saa pystyä lähettämään vanhoja viestejä eivätkä viestit saa jäädä välille odottelemaan. Erityisen tärkeää tämä on reaaliaikaisuutta vaativissa tehtävissä, esimerkkinä vaikkapa verensokerin mittaus, jolloin vanhentunut tieto voisi saada insuliinipumpun annostelemaan väärin.

2.1.3 Hyökkäykset

Erilaisia hyökkäyksiä Islam, Shen ja Wang (2012) esittelevät viisi kappaletta:

1. **Salakuuntelu**: Salakuuntelussa ulkopuolinen hyökkääjä vain kuuntelee älykodin verkon liikennettä ja tekee siitä päätelmiä. Hän voi myös aktiivisesti lähettää erilaisia viestejä, joiden tarkoituksena on kartoittaa verkkoa tarkemmin. Samaan kategoriaan kirjoittajat laskevat myös radioliikenteen häiritsemisen, jolloin pahimmassa tapauksessa mitkään oikeat viestit eivät pääse perille.

   Salakuuntelu voidaan estää tarpeeksi vahvalla salauksella sekä autentikoinnilla ja eheystarkistuksilla. Häirintää voidaan yrittää estää esimerkiksi hyppimällä taajuusalueelta toiselle mutta loppujen lopuksi kaikki nämä voidaan 'ohittaa', joten pohjimmiltaan häirinnän täydellinen esto on mahdotonta. Li ja Lin (2015) esittelevät arkkitehtuurin,

jossa sensorit lähettävät datansa langattomasti jokaisessa huoneessa olevalle päätelaitteelle, joka sitten lähettää datan eteenpäin gatewaylle. Tässä menetelmässä on tiedonsiirron luotettavuus suurempi, sillä päätelaitteet voivat huolehtia, että data pääsee perille. Tämä hieman vähentää häirinnän toimivuutta.

2. **Palvelunestohyökkäys**: Liittyy vahvasti edelliskohdassa mainittuun häirintään, erona se, että tämä voidaan toteuttaa myös muilla kerroksilla kuin fyysisellä radioliikenteellä. Tämän hyökkäyksen tarkoituksena on nimensä mukaisesti estää palvelua toimimasta, ja IoT-laitteiden tapauksessa usein pyritään myös kuluttamaan laitteiden akut loppuun pakottamalla ne jatkuvasti lähettämään viestejä uudelleen. Lisäksi hyökkäyksellä voidaan saada esimerkiksi reititystiedot sekaisin.

   Palvelunestohyökkäysten estoon Islam, Shen ja Wang (2012) mainitsevat erilaisia keinoja ja tutkimuksia: törmäyksiä voidaan vähentää satunnaisilla 'back-offeilla' tai rajoittamalla MAC-liikennettä sekä käyttämällä pieniä frame-kokoja; tutkimukset esittelevät keinoja identifioida huonosti käyttäytyvät solmut ja välttää niitä reitityksessä, tai käyttää virtuaalivaluuttaa liikenteen turvaamiseen.
3. **Solmun murtaminen (node compromise)**: Solmun murtamisessa joku verkon laite otetaan haltuun ja sen kautta tehdään erilaisia hyökkäyksiä verkkoon, esimerkiksi lähetetään väärää tietoa, luetaan muiden laitteiden lähettämää salattuakin tietoa, tai väitetään oikeaa laitetta saastuneeksi, jolloin sen viestejä ei hyväksyttäisi. Erityisen vakavaa tämä on gateway-arkkitehtuurissa, jos gateway saadaan saastutettua.

   Solmun murtamisen estoon Islam, Shen ja Wang (2012) esittävät koodin testausta, jossa solmun satunnaisista muistipaikoista lasketaan hashit ja niitä vertaillaan. Toinen esitetty keino olisi verrata solmun fyysistä sijaintia, oletuksella että murtaminen on tehty fyysisesti kaappaamalla solmu ja tuomalla se takaisin.
4. **Kuoppa- (sinkhole) ja madonreikä-hyökkäykset**: Kuoppahyökkäyksessä on tarkoituksena saada verkon reititys muuttumaan niin, että optimaalisessa tilanteessa kaikki laitteet reitittävät viestinsä saastuneen solmun kautta. Tällöin saastunut solmu saa kaikki viestit haltuunsa ja voi halutessaan joko pelkästään lukea niiden datan ja lähettää eteenpäin tai vaikkapa aiheuttaa palvelunestoa hukuttamalla kaikki paketit.

   Madonreikä-hyökkäyksen erona on se, että siinä ei tarvita laitetta välttämättä lainkaan,

vaan se voidaan tehdä kokonaan radioliikennetasolla. Tällöin hyökkääjä kaappaa laitteiden liikenteen yhdessä paikassa ja siirtää sen toiseen kaukaisempaan paikkaan jotain eri reittiä pitkin ja lähettää sen siellä ilmoille, samoilla tiedoilla kuin alkuperäinen viesti. Tällöin oikeasti fyysisesti etäällä olevat laitteet luulevat olevansa naapureita, eivätkä käytä minkäänlaista reititystä paketeilleen, vaan lähettävät ne suoraan toisilleen, ts. madonreikään. Myös tässä tapauksessa hyökkääjä saa kaikki viestit haltuunsa, (tai ainakin ne, jotka menevät reiän päästä toiseen) ja voi tehdä niillä mitä haluaa tai pystyy.

Näiden hyökkäysten estoon Islam, Shen ja Wang (2012) mainitsevat jokaisen laitteen ja tukiaseman/gatewayn välillä olevan uniikin avaimen. Muita vaihtoehtoja on käyttää reitityksessä esimerkiksi multi-path:ia tai laittaa solmut varmistamaan toistensa identiteetit. Loppujen lopuksi erityisesti madonreikä-hyökkäystä on lähes mahdotonta estää, sillä se ei näy korkeammille verkkokerroksille välttämättä lainkaan.

5. **Fyysinen hyökkäys**: Hyökkääjä voi päästä fyysisesti käsiksi laitteisiin, jolloin hän voi yksinkertaisimmillaan tuhota ne tai vaihtaa niiden paikkaa, varastaa ne, asentaa pahantahtoista koodia tai ottaa laitteesta ulos salausavaimia. Estokeinona jonkinlainen peukaloinnin estävä rauta, esimerkiksi liikutettaessa laitetta ilman lupaa pyyhitään muisti tyhjäksi ja poistetaan verkosta.

Geneiatakis ym. (2017) mainitsevat älykodin uhkamalleina ulkoisen ja sisäisen hyökkääjän, jotka voivat toimia passiivisesti tai aktiivisesti. Passiiviset hyökkääjät yrittävät kuuntelemalla liikennettä saada haltuunsa tietoja, joiden avulla voidaan joko päätellä jotain käyttäjistä tai niitä voidaan käyttää aktiivisen hyökkäyksen apuna. Aktiiviset hyökkääjät yrittävät saada laitteita tekemään jotain, muokata niiden tietoja jne. Verkkotason hyökkäysten lisäksi toimijat yrittävät hyökätä ohjelmistotasolle.

Erityisinä ongelmina Geneiatakis ym. (2017) mainitsevat neljä hyökkäystä: salakuuntelu, esiintyminen käyttäjinä, palvelunestohyökkäys, ja ohjelmistojen hyväksikäyttö. Erityisesti kaksi ensimmäistä ovat mahdollisia lähinnä vain silloin, kun hyökkääjä on jotenkin päässyt sisäverkkoon käsiksi. Ulkoverkosta palvelunestohyökkäykset onnistuvat todennäköisesti vain reitittimeen, lähempää langattomana langattomiin laitteisiin. Ohjelmistojen hyväksikäyttö vaatii joko käyttäjää asentamaan malwarea, tai haavoittuvuutta oikeassa ohjelmistos-

sa, jonka täytyy tällöin olla hyökkääjän tavoitettavissa. Salakuuntelun ratkaisuksi he mainitsevat salauksen, ja käyttäjänä esiintymisen ratkaisuksi eheyden tarkistamisen ja autentikoinnin. Palvelunestohyökkäyksiin ei heidän mukaansa ole kunnollisia ratkaisuja edes IP-verkoissa, saati sitten IoT-verkoissa. Ohjelmistojen hyväksikäyttön ratkaisuksi he esittävät, että käyttäjien tulisi käyttää vain tunnettuja reittejä pitkin tarjottuja sovelluksia ja palveluja. Ratkaisuna oletusasetusten turvattomuudelle he esittävät, että laitteita ei voisi lainkaan käyttää ennenkuin ne ovat kunnolla konfiguroituja: etähallinta mahdollista vain, kun salasana on tarpeeksi hyvä, portteja ei auki oletuksena, haavoittuvaiset protokollat poissa käytöstä, jne.

Erilaisia hyökkäyksiä ja niiden estokeinoja esittelevät myös muun muassa Farooq ym. (2015).

2.2 Älykoti-arkkitehtuurit

Lin ja Bergmann (2016) esittävät kolmeksi tärkeimmäksi ja suosituimmaksi älykotiarkkitehtuuriksi middleware-, pilvi-, ja gateway-arkkitehtuurit.

2.2.1 Middleware

Middleware on ohjelmistokerros, joka on matalan tason laitteistokerroksen ja korkean tason sovelluskerroksen välissä. Sen tarkoituksena on abstraktoida matalan tason yksityiskohdat ja tarjota standardoitu keskusteluprotokolla laitteille. Kun käyttäjä haluaa pyytää laitteelta dataa tai antaa sille käskyn, tekee hän tämän middlewaren kautta, jolloin hänen ei tarvitse tietää mitään laitteen matalan tason toiminnasta, vaan middleware hoitaa muunnokset. Samoin vastaus tulee aina samassa muodossa riippumatta laitteen sisäisen datan muodosta.

Useimmiten tällainen middleware on implementoitu laitteessa itsessään, mikä tarkoittaa että laitteen tulee olla suhteellisen tehokas ja laskentakykyinen. Lin ja Bergmann (2016) mukaan tämä sekä se, että middlewaren ohjelmointivirheet voivat altistaa laitteet turvallisuusuhille, tekevät middlewaresta sopimattoman useille IoT-laitteille.

2.2.2 Pilvi

Pilvi-arkkitehtuuri on eräs ratkaisu IoT-laitteiden yhteistoiminnan vaatimalle laskentatehole. Tällöin ei laitteiden lisäksi tarvita mitään erillistä gatewayta, mutta sen vastapainoksi Internet-yhteyden tulee olla nopea ja luotettava. Jos Internet-yhteys katkeaa, häviää kaikki pilven avulla realisoitunut älykodin toiminnallisuus. Samoin pilven etäisyydestä riippuen tulee toimintavasteesta suhteellisen pitkä.

Tällöin myös kaikki laitteet ovat yhteydessä Internetiin, mikä lisää hyökkäysmahdollisuuksia ja tarkoittaa sitä, että laitteilla pitää olla tarpeeksi tehoa jotta ne voivat toteuttaa tarvittavat suojausmekanismit ja salaukset. Parhaassa tapauksessa laitteet eivät näy ulospäin mitenkään, vaan niiden ainoa yhteys on itse avattu pilveen. Tässä tapauksessa suurin haavoittuvuusuhka on pilvi itse, jonka turvallisuuteen käyttäjien tulisi luottaa, erityisesti tallennetun datan osalta.

Lin ja Bergmann (2016) mukaan pilvi-arkkitehtuuri ei ole sopiva älykodin arkkitehtuuriksi, koska se vaatii ehdottomasti jatkuvan Internet-yhteyden sekä altistaa kaikki laitteet verkkohyökkäyksille.

2.2.3 Gateway

Gateway-arkkitehtuurissa[1] on älykotiverkon ytimenä suhteellisen tehokas laite, joka on samassa verkossa älykotilaitteiden kanssa. Sen tarkoituksena on yhdistää (eri valmistajien) älylaitteet ja koordinoida niiden välistä toimintaa. Tämän lisäksi se yleensä on jollain tavoin avoinna Internetiin, jotta älykodin verkkoa voi hallita etänä.

Tässä mallissa itse laitteiden ei tarvitse olla järin tehokkaita, vaan riittää että ne saavat datansa lähetettyä gatewaylle, joka sitten hoitaa kaiken laskennan. Gatewayn avulla saadaan siihen keskitettyä suurin osa turvallisuudesta, kuten autentikointi ja pääsyn hallinta. Samalla

1. Gatewayta nimityksenä käytetään sekä pelkästään ulkoverkkoyhteyden tarjoavista laitteista, että sellaisista, joissa sen lisäksi on mukana laskenta ja automatiikan hoito. Hubi-nimitystä käytetään useimmiten laitteista, jotka hoitavat laskennan ja automatiikan, mutta eivät välttämättä tarjoa ulkoverkkoihin yhteyttä. Tässä tutkielmassa gateway tarkoittaa laitetta, joka hoitaa älykodin laitteiden välisen tiedonsiirron ja laskennan sekä automatiikan, ja lisäksi tarjoaa etähallintamahdollisuuden eli on yhteydessä Internetiin, mutta ei välttämättä itse suoraan, vaan esimerkiksi reititinmodeemin kautta.

myös ainoa yhteys Internetiin on gatewaysta, jolloin hyökkäyspinta-ala pienenee. Gateway-ratkaisu mahdollistaa myös verkkohyökkäysten havaitsemisen verkkoliikenteestä, kuten esimerkiksi Singh, Sharma ja Park (2017), ja Pacheco ja Hariri (2016) esittävät.

Lin ja Bergmann (2016) mielestä gateway-arkkitehtuuri sopii älykotiin parhaiten: Siinä on tarpeeksi tehoa toteuttamaan monimutkaisetkin säännöt, se toimii myös ilman Internet-yhteyttä, se suojaa allaan olevia laitteita verkkohyökkäyksiltä ja se toimii ilman laitteissa olevaa middlewarea. Käytännössä gateway toteuttaa osaltaan middlewaren tehtävän, ainakin jos ajatellaan käyttäjän ja älykotilaitteen välistä kommunikaatiota.

Gateway-arkkitehtuurin huonona puolena, pilvi-arkkitehtuurin tavoin, on yksittäiseen pisteeseen kasaantuva toiminta, jolloin sen pettäminen kaataa koko verkon. Pilvi-arkkitehtuurissa se oli Internet-yhteys, tässä se on itse gatewayn toiminta. Jos gateway hajoaa, ilman että sen asetuksia on varmuuskopioitu, joudutaan koko älykotiverkko asentamaan uudestaan, mikä voi olla laajemman verkon tapauksessa suurikin toimenpide. Täten olisi hyvä, jos gateway automaattisesti varmuuskopioisi tietonsa.

Lin ja Bergmann (2016) mukaan gateway-arkkitehtuurin käyttöönoton yleistymisen haasteena on käytännössä kaksi asiaa: automaattinen konfigurointi ja ohjelmisto/firmware-päivitykset. Suurimman turvallisuuden aikaansaamiseksi tulisi älykodin laitteiden ja verkon asentamisen olla mahdollisimman helppoa, kuitenkin niin, että lopputulos on olosuhteiden puitteissa niin turvallinen kuin mahdollista. Lin ja Bergmann (2016) esittämässä arkkitehtuurissa gateway kysyy asennettavana olevan laitteen tiedot verkkopalvelusta eikä laitteelta itseltään. Tällöin tieto on ajantasaista ja laitteen vaatimukset vähenevät. Toinen vaatimus gatewaylle on laitteiden automaattinen päivitys. Päivitysten tulisi myös olla salattuja ja digitaalisesti allekirjoitettuja, jotta niitä ei ole mahdollista muunnella eikä laitteille voi asentaa muita kuin valmistajan sallimia firmwareja.

Son ym. (2015) esittävät middlewaren ja gatewayn yhdistelmää, jossa gateway on suhteellisen kevyt ja hoitaa vain laitteiden keskinäistä koordinointia kunkin taskin kohdalla; kaikki laskenta tapahtuu laitteissa itsessään. Tämä on kuitenkin toimimaton arkkitehtuuri vähänkään kevyempien älylaitteiden kohdalla. Heidän tutkimuksessaankin älylaitteet ovat suhtees-

sa erittäin tehokkaita[2].

Zheng, Wang ja Tan (2013) esittelevät mukautuvan gatewayn. Sitä pystyttäisiin päivittämään tarpeen mukaan ja näin ollen se ratkaisisi heterogeenisyys-ongelman.

Pishva ja Takeda (2006) esittävät älykodin laitteille erilaista turvallisuustarvetta sen mukaan, mikä on erilaisten uhkien todennäköisyys kyseiselle laitetyypille. He esittävät älykodin turvallisuusongelmien ratkaisuun seuraavaa: 1. Saada verkko-operaattori rakentamaan dedikoitu mutta ei-yksityisomistuksessa oleva (avoimen lähdekoodin) koti-gateway ja tulla suositelluksi luotettavaksi kolmanneksi osapuoleksi. 2. Saada älylaitteiden valmistajat kehittämään ajureita tälle gatewaylle laitteiden ohjaamisen mahdollistamiseksi. Pishva ja Takeda (2006) esittävät universaalin koti-gatewayn tarvittaviksi ominaisuuksiksi seuraavia:

1. Kotiverkon sisällä autentikoida käyttäjät esimerkiksi salasanalla tai muistikortti-ratkaisulla.
2. Toimia turvallisuusserverinä ja pitää huolta käyttäjien käyttöoikeuksista.
3. Tarjota turvallinen kommunikointi ja huolehtia turvallisuuskysymyksistä käyttäjien puolesta. Tarjota etäkäyttömahdollisuus. Sallia verkko-operaattorien autentikoida käyttäjät gatewayn avulla tai laskuttaa kolmannen osapuolen palveluista.
4. Sallia uusien laitteiden lisääminen ja pyytää käyttäjää asettamaan valmistajan tarjoama ajuriohjelmisto. Kuitenkin niin, että lisätään vain haluttaessa, ei täysin vapaana plug-and-play:na.
5. Automaattisesti poistaa käytöstä verkosta irrotettu laite ja toisaalta automaattisesti konfiguroida uudelleenlisättävä, aiemmin poistettu laite.
6. Sisältää palomuuri- ja virustorjunta-ohjelmistot.

Bregman ja Korman (2009) esittävät älykotiarkkitehtuurin, joka koostuu neljästä moduulista: Central Management Unit, User Interface, Home Equipment and Appliances Interface ja External Communication Interface. Näistä CMU on kaiken ytimenä, ja toimii käytännössä gatewayna, jota hallitaan User Interfacen kautta. HEAI:n kautta laitteet ovat yhteydessä CMU:hun. ECI:n kautta CMU voi olla yhteydessä useisiin eri yhteysteknologioihin.

Kim ja Keum (2017) esittelemä Trust Domain -malli on pohjimmiltaan gateway-ratkaisu. Siinä sallitaan älylaitteiden turvattomuus sillä, että ne ovat luotetun gatewayn takana ja yh-

2. Samsung Exynos4412 Prime Cortex-A9 Quad Core 1.7Ghz and 2GB LP-DDR2 Memory

teyden ottamiseen käytetään suoran IP-osoitteen sijasta ID:tä, joka on tiedossa vain luotetuilla toimijoilla.

Hosek ym. (2014) esittelevät gateway-mallin, johon on yhdistetty myös perinteinen reititin. Tässä tapauksessa ei siis tarvita kuin kyseinen gateway, joka hoitaa sekä älykotiverkon toiminnan että perinteisemmän Ethernet/WiFi-kotiverkon. Heidän esittämänsä, Home Gateway Initiativeen (HGI) perustuvat, vaatimukset ovat:

- Helppokäyttöisyys ja yksinkertaisuus - konfigurointi ilman toimenpiteitä ja plug-and-play.
- Turvallisuusfunktiot, sisältäen palomuurin, pääsyn hallinnan ja henkilökohtaisen monitoroinnin, jotta käyttäjä tuntee olonsa turvalliseksi Internetistä tulevia uhkia vastaan.
- Quality of Service (QoS), asiakkaan liikenteen priorisointi.
- Suorituskyvyn monitorointi ja diagnostiikka. Lisäksi ongelmiin puututaan vasta kun käyttäjä niin haluaa, eli reaktiivisesti, ei proaktiivisesti.

Kaikkia edellä kuvattuja gatewayn vaatimuksia mukaillen esitän älykodin gatewayn vaatimuksiksi seuraavia:

- **Mahdollistaa laitteiden välisen kommunikoinnin, niiden käyttämistä teknologioista riippumatta.** Tämä monipuolistaa mahdollisia laitteita eikä käyttäjän tarvitse rajoittua yhteen valmistajaan tai teknologiaan. Varsinkin kaupallisissa gatewayssa on usein ongelmana, että tuetaan vain saman valmistajan tuotteita.
- **Mahdollistaa laitteiden hallinnan sekä lokaalisti että haluttaessa myös etänä.** Vähintään lokaali hallinta täytyy olla, sillä muuten laitteita ei voida hallita ollenkaan. Etähallinta todennäköisesti lisää käyttömukavuutta, mutta lisää myös turvallisuusriskejä.
- **Mahdollistaa automaattisten toimintojen asettamisen sensoridatan tai esimerkiksi kellonajan perusteella.** Älykoti ei ole älykäs, ellei siinä ole jotain automaatiota, vaikka toisaalta on mahdollista myös vain kerätä dataa logeihin.
- **Mahdollistaa eritasoiset käyttäjät.** Eritasoisilla käyttäjillä voidaan parantaa sekä älykodin turvallisuutta että käyttömukavuutta. Vain osa käyttäjistä pystyy muokkaamaan laitteita tai sääntöjä ja kukin käyttäjä näkee vain ne laitteet, jotka hän haluaa nähdä.

- **Mahdollistaa käyttäjien autentikoinnin.** Turvallisuuden kannalta kenties keskeisin osa, jolla varmistetaan, että vain oikeat käyttäjät pääsevät gatewayta hallitsemaan.
- **Mahdollistaa laitteiden autentikoinnin.** Liittyy turvallisuuteen: Riippuen tavasta, jolla gateway yksilöi laitteet, voi olla mahdollista vaihtaa oikea laite 'rogue'-laitteeseen.
- **Mahdollistaa uusien laitteiden lisäämisen sekä vanhojen poistamisen.** Uusia laitteita pitää pystyä lisäämään ja poistamaan milloin tahansa, ilman että muiden laitteiden toiminta häiriintyy. Asennuksen pitää siis olla dynaaminen, ei 'kiinteä'.
- **Pitää itsensä päivitettynä.** Automaattinen päivitys, tai ainakin päivityksestä muistuttaminen, on tärkeää varsinkin turvallisuuden kannalta.
- **On mahdollista lisätä laitteiden lisäksi esimerkiksi uusia yhteysteknologioita, esimerkiksi usb-porttiin liitettävillä lisälaitteilla.** Jälleen dynaamisuuden ja 'future-proofingin' kannalta tärkeää.
- **Pitää huolta laitteiden päivityksistä, joko itse tai sallimalla laitteelle yhteyden Internetiin päivitysserverille.** Ensin mainittu olisi optimitilanteessa turvallinen ratkaisu, jolloin turvattoman laitteen haavoittuvuus ei vaarantaisi koko verkkoa, vaan gateway pitäisi huolta, että päivitykset ovat oikeita ja valmistajan tarjoamia. Se kuitenkin vaatisi paljon työtä ja valmistajien yhteistyötä, joten turvallisuuden kustannuksella on helpompaa sallia laitteiden päivittää itse itsensä. Tai jos niillä ei ole tarvetta päästä Internetiin, voidaan niiden yhteydet sinne sulkea ja periaatteessa näin turvatonkin laite voi olla älykodin osana.
- **Kerää logitietoja laitteiden toiminnasta. Toisaalta sallii kaiken tiedon keräämisen asettamisen pois päältä.** Sensoreiden logitietoja voidaan käyttää esimerkiksi energian käytön seurantaan ja ohjelmistojen logitietoja vikojen korjaamiseen. Toisaalta logitiedot voivat olla salassapidettäviä ja henkilökohtaisia, joten niiden kerääminen pitää myös olla mahdollista estää kokonaan.
- **Ilmoittaa käyttäjälle virheistä.** On tärkeää tietää, milloin ja miksi virheitä tapahtuu, erityisesti jos laitteen virhe vaikuttaa koko älykodin toimintaan, esimerkiksi lukon tai lämmityksen kohdalla.
- **Tutkii kotiverkon liikennettä ja havaitsee siitä hyökkäyksiä sekä suorituskykyongelmia.** Jotta älykoti pysyy turvallisena ja toimivana.
- **Tekee kaiken yllämainitun turvallisesti ja käyttäjäystävällisesti.**

Todennäköisesti yksikään gateway ei näitä kaikkia kuitenkaan toteuta. Siinä tapauksessa, että älykoti-gateway hoitaa Internet-yhteyden jonkin toisen laitteen kautta, siirtyy osa tehtävistä yleensä sille, erityisesti ulkoisten, sekä muihin kuin 'älylaitteisiin' (esimerkiksi pöytätietokoneet, läppärit ja älypuhelimet) kohdistuvien hyökkäysten havaitseminen. Nämä vaatimukset keskittyvät kuitenkin älykotilaitteiden verkkoon, jättäen perinteisemmän verkkotoiminnan huomioimatta.

2.3 Etäyhteys

Etäyhteys voidaan toteuttaa usealla eri tavalla, yleisimpien ollessa suora yhteys, käänteinen välityspalvelin (reverse proxy), VPN-yhteys, sekä pilviyhteys. Ne kaikki voivat olla turvallisia ja toisaalta turvattomia, riippuen asetuksista ja valitusta toteutuksesta. Niiden avulla on tarkoitus mahdollistaa turvallinen yhteys älykotiin sen ulkopuolelta. Siihen ei riitä pelkkä käyttäjän autentikointi, vaan liikenteen tulee olla salattua ja serverinkin aitous pitää tarkistaa. Yhdenkin puuttuminen vaarantaa turvallisuuden ja tietojen salaisuuden.

Näistä tavoista suora yhteys, käänteinen välityspalvelin, sekä VPN-yhteys vaativat optimaalisessa tilanteessa kaikki joko kiinteän IP-osoitteen tai domain-nimen, tai dynaamisen domain-nimen. Vaihtuvallakin IP-osoitteella yhteydet toimivat, mutta tällöin sen oikeellisuus pitää aina varmistaa ja jos se vaihtuu käyttäjän ollessa poissa kotoa, ei sitä pysty tarkistamaan. Kiinteä IP-osoite tai domain-nimi yleensä maksavat palveluntarjoajalta hankittuina, dynaamisen domain-nimen voi saada ilmaiseksi esimerkiksi DuckDNS:ltä.

Riippuen etäyhteyden konfiguroinnista, sen turvallisuus voi olla kiinni pelkästään hyvin valitusta salasanasta. Vaikka yhteys olisi salattu oikein, ei siitä ole käytännössä mitään hyötyä jos salasanaksi on valittu '1234'. Optimaalista olisi käyttää sertifikaatteja yhteyttä ottaville laitteille, mutta tämä vaatii enemmän teknistä osaamista, vaikkakin ohjeita löytyy. Laitteiden tunnistamiseen (fingerprinting) perustuvan autentikointimenetelmän esittelevät Jose, Malekian ja Ye (2016). Siinä laitteet yksilöitäisiin käyttämällä JavaScriptiä, Flashia ja HTML5:n Geo-Locationia. Näiden avulla saataisiin erilaisia tietoja, joiden avulla laitteet voidaan identifioida 97,93%:n tarkkuudella. Lisäksi vaaditaan käyttäjätunnus ja salasana.

2.3.1 Suora yhteys

Suora yhteys tarkoittaa sitä, että laite, johon yhteys otetaan, on suoraan näkyvissä Internetiin. Useimmiten tämä toteutetaan käskemällä reititintä ohjaamaan tiettyyn porttiinsa tuleva liikenne halutulle laitteelle ja portille sisäverkossa. Tällöin yhteys voidaan muodostaa, kun tiedetään reitittimen julkinen IP-osoite sekä ulkoinen portti, jonka kautta liikenne ohjataan; sisäporttia ei tarvitse tietää.

Hyvänä puolena tässä ratkaisussa on sen helppous ja yksinkertaisuus, sekä joissain tapauksissa suorityskyky. Tässä ratkaisussa ei tarvita välikäsiä yhteyden muodostamiseen eikä näin ollen tarvitse asentaa tai konfiguroida mitään muuta kuin porttiohjaus reitittimeen. Samoin ei tarvitse huolehtia välikäden turvallisuudesta.

Huonona puolena tässä on se, että laite tosiaan on käytännössä suoraan näkyvissä Internetiin, jolloin sen täytyy olla turvallinen. Tämä ei usein pidä paikkaansa, vaan ohjelmisto on suunniteltu olemaan jonkin turvallisen välikäden takana, tai sitä ei ole suunniteltu etäyhteyteen lainkaan. Näissä molemmissa tapauksissa suora yhteys on todennäköisesti täysin turvaton. Lisäksi koska on olemassa juuri tätä turvallisen välikäden roolia hoitavia ohjelmistoja, ei ole juurikaan syytä olla käyttämättä sellaista.

2.3.2 Käänteinen välityspalvelin (reverse proxy)

Käänteinen välityspalvelin toimii käytännössä välikätenä gateway-ohjelmiston ja etäyhteyttä haluavan laitteen välillä. Tällöin yhteys otetaan välityspalvelimeen, joka ohjaa liikenteen itse gateway-laitteelle. Näin perusmuodossaan ei välityspalvelimesta älykodin etäyhteydessä olisi mitään hyötyä, mutta sen avulla voidaan hoitaa liikenteen salaus ja autentikointi.

Hyviä puolia ovat turvallisuuden selvä lisäys suoraan yhteyteen verrattuna, jos välityspalvelin konfiguroidaan oikein. Tällöin liikenne on salattua, serverillä on sertifikaatti sitä varten, ja yhteydenotot autentikoidaan käyttäjätunnus/salasana-parilla, IP-osoitteella, ja/tai käyttäjäsertifikaatilla.

Huonona puolena on yhden lisäohjelmiston asennus ja konfigurointi, sekä mahdollisuus tehdä virheitä näissä, mikä pahimmassa tapauksessa ei lisää turvallisuutta lainkaan, mutta vä-

hentää käyttäjäystävällisyyttä. Varsinkin käyttäjäsertifikaattien käyttö lisää ainakin aluksi monimutkaisuutta.

2.3.3 VPN-yhteys

VPN (virtual private network) sallii etäyhteyden ottamisen niin, että etälaitteen ja kotiverkon välille muodostetaan tunneloimalla salattu yhteys, ja laitteet muodostavat virtuaalisen lähiverkon. Tällöin etälaite 'näkee' olevansa kotiverkossa ja voi täten ottaa yhteyden gatewayhin.

Hyvänä puolena VPN-yhteydessä on sen turvallisuus, sillä oikein asennettuna se takaa salassapidon, autentikaation ja eheyden. Erona käänteiseen välityspalvelimeen on se, että VPN tukee kaikkea IP-liikennettä, (useimmiten) pelkän HTTP/S-liikenteen sijaan. Samoin se salaa kaiken liikenteen, jolloin kaapatusta liikenteestä ei nähdä esimerkiksi metatietoja tai muuta HTTPS-informaatiota.

Huonona puolena on käänteistä välityspalvelintakin monimutkaisempi asennus, sekä se, että oletuksena etälaite näkee kaikki sisäverkon laitteet, eikä vain haluttua älykotigatewayta. Tämä voi olla haluttuakin, mutta jos ei, niin hyökkääjän päästessä VPN:ään voi hän hyökätä kaikkia sisäverkon koneita vastaan. Tämä on ongelmana erityisesti silloin, kun autentikointiin käytetään vain käyttäjätunnusta ja salasanaa eikä käyttäjäsertifikaatteja.

2.3.4 Pilviyhteys

Pilvipalvelun kautta etäyhteys toteutetaan niin, että gateway ottaa (ja pitää yllä) yhteyden pilveen, johon käyttäjä ottaa yhteyden etänä ja voi näin käskyttää gatewayta pilven kautta. Periaatteessa pilvi voi toimia pelkästään käänteisenä välityspalvelimena, mutta useimmiten gatewayn hallinta tapahtuu pilven tarjoaman käyttöliittymän kautta, joka voi olla sama kuin gatewayn itsensä tarjoama. Pilvipalvelua käyttävät varsinkin kaupalliset gateway-valmistajat, sillä heillä on mahdollisuus ja intressejä tarjota tällainen palvelu.

Hyvänä puolena pilvipalvelu voi tarjota esimerkiksi enemmän tilastoja ja datan hallintaa kuin gateway itse, suuremman laskenta- ja tallennuskapasiteetin vuoksi. Lisäksi sitä käytettäessä ei käyttäjän tarvitse huolehtia etäyhteydestä käyttäjätilin (ja mahdollisten sertifi-

kaattien) luontia enempää. Turvallisuus voi myös olla parempi, sillä ohjelmiston asennuksen ovat todennäköisesti tehneet asiantuntijat, joten käyttäjän ei tarvitse pelätä tekevänsä virheitä etäyhteyden asennuksessa.

Selkein huono puoli on se, että kerätty data on pilvessä eikä sen salassapitoa voi taata. Usein palvelun tarjoajat saattavat myös käyttää dataa omiin tarkoituksiinsa eikä datan hallinta ole käyttäjän käsissä, eli yksityisyys lähes varmasti huonontuu. Samoin saatavuus voi olla heikompi, mutta jos pilvi on vähänkään isomman toimijan tarjoama, niin tämän ei pitäisi olla vaarana. Huono puoli voi olla myös se, jos tarjotaan vain ja ainoastaan pilvessä oleva käyttöliittymä sekä älykodin hallinta ja laskenta. Tällöin menetettäessä Internet-yhteys menetetään myös koko älykodin toiminta ja hallinta.

Pilvipalvelu on myös mahdollista asentaa 'yksityisenä' ja itse, jos siihen tarjotaan avoin lähdekoodi. Tässäkin tapauksessa ongelmana on tiedon yksityisyys, eli käyttääkö pilven tarjoaja jotain tietoja hyväksi, mutta riski tähän vähenee huomattavasti.

3 Alustojen tietoturva-auditoinnit

Tässä kappaleessa esitellään tutkimuksessa käytetty laitteisto, tutkitut ohjelmistot, sekä tutkimuksen eteneminen ja tulokset. Kaikki tässä kappaleessa esitellyt ja tutkitut ohjelmistot ovat avointa lähdekoodia.

3.1 Tutkimuksen eteneminen

Jokainen ohjelmisto tutkittiin asentamalla se Raspberry Pi Model 3:een, lisäämällä siihen laitteita ja asettamalla niiden välille joitain automaatioita. Ohjelmiston asentaminen tehtiin siten, että pyrittiin valitsemaan aina oletusasetukset, eli 'näyttelemällä' peruskäyttäjää ja menemällä helpoimman kautta. Samalla huomioitiin joka kohdassa mahdolliset turvallisuusvalinnat ja se, oliko ne helppo huomata ja asettaa päälle.

Käytetyt laitteet olivat

- Danfoss Living Connect Z 014G0013 v1.01, Z-Wave
- D-Link DCH-Z110 Ovi-/ikkunasensori, Z-Wave
- D-Link DCH-Z510 Sireeni, Z-Wave
- Belkin WeMo Switch pistorasia, WiFi
- Philips Hue Bridge, Ethernet/Zigbee
- Philips Hue lamppu, Zigbee

Näistä selkeimmin sensori on ovi-/ikkunasensori, muut ovat aktuaattoreita. Sensorissa on lisäksi valoisuus- ja lämpötilamittaus. Laitteiden välille asetettiin mahdollisuuksien mukaan automaatioita, joissa kaikki laitteet olivat mukana. Näin niihin saatiin mukaan eri teknologioita ja niiden välinen toiminta varmistettiin. Alunperin tarkoituksena oli käyttää myös Centraliten Zigbee-vesisensoria, mutta käytössä ei ollut mitään laitetta, jota yksikään ohjelmisto olisi suoraan tukenut Zigbee-kontrollerina, joten se jätettiin pois kokonaan. Tämän takia Zigbee-kartoitus jäi vähäisemmäksi, mutta tarkoitus ei ollutkaan kartoittaa erityisesti mitään tiettyä teknologiaa, sillä niiden tuki vaihtelee huomattavasti.

Etähallinnan turvallisuutta kartoitettiin sekä hyökkäämällä suoraan porttiskannauksella saa-

tujen avoimien porttien takana oleviin palveluihin, että pintapuolisesti tutkimalla niiden lähdekoodia.

Kaikkien ohjelmistojen asennus tehtiin joko flashaamalla ohjelmiston sisältävä levykuva suoraan SD-kortille, tai asentamalla SD-kortille ensin Raspbian Stretch -käyttöjärjestelmä, jonka sisältä itse gateway-ohjelmisto sitten asennettiin. Nämä eivät varsinkaan ohjeita seuraamalla ole vaikeita toimenpiteitä, eikä helpompaa vaihtoehtoa oikeastaan ole olemassakaan, ellei asenna ohjelmistoa omalle kotikoneelleen.

Laitteista Philips Hue sekä Belkin Wemo vaativat, että ne asennetaan ensin kodin sisäverkkoon mobiilisovellustensa kautta. Varsinkin Wemon asennus tuotti ongelmia, sillä se vaati useita asennusyrityksiä ja virrankatkaisun jälkeen usein uudelleenasennuksen.

3.2 OpenHAB

OpenHab[1] on Java-pohjainen ohjelmisto, joka pyrkii tarjoamaan yhden eri kotiautomaatiojärjestelmät yhdistävän ratkaisun, joka sallii niiden välisen automatiikan ja tarjoaa yhtenäisen käyttöliittymän. Se on valmistaja-neutraali ja laitteisto- sekä protokolla-agnostinen.

OpenHABilla on vireä yhteisö ja paljon käyttäjiä. Lisäksi dokumentaatio on kattavaa ja ohjeita löytyy. Uusin versio on julkaistu 28.6.2017.

OpenHABin kehittäjät itsekin myöntävät, että se ei ole helppo asentaa ja sopii lähinnä asian harrastajille. Sen käyttöä sen sijaan kehutaan helpoksi, jolloin vain yhden asukkaista (tai esimerkiksi vuokranantajan puolelta olevan asentajan) täytyy hallita tekniikkapuoli.

Verkkosivujensa mukaan OpenHAB tukee yli 200 erilaista teknologiaa ja järjestelmää sekä tuhansia erilaisia laitteita. Eri yhteysteknologioista tuetaan esimerkiksi Bluetoothia, Zigbeetä ja Z-Wavea, mutta esimerkiksi Zigbee-laitteista toimiviksi mainitaan vain Philipsin Huelamput ja SmartThingsin pistorasia. Z-Wave näyttäisi olevan laajemmin tuettu, ja sen tuki on lähinnä kiinni tuetuista Command Classeista. Z-Waven turvallisuusominaisuudet ja esimerkiksi Security-luokka ovat vielä beta-vaiheessa, mutta ohjeet sen käyttämiseen ja toiminnan varmistamiseen löytyvät.

1. http://www.openhab.org/

OpenHAB toimii modulaarisesti: kun jokin uusi teknologia tai järjestelmä halutaan liittää osaksi OpenHAB-älykotia, tehdään se liittämällä siihen Binding. Esimerkiksi sosiaalisten medioiden, pikaviestien, tai IoT-pilvialustojen integroimiseen käytetään erillisiä add-oneja, nekin modulaarisesti lisättäviä.

OpenHABissa kaikki datan hallinta on käyttäjällä, eikä sitä tarvitse niin halutessa mihinkään lähettää tai edes tallentaa. Samoin etähallinta voidaan jättää pois käytöstä. Käyttäjä voi itse valita mitä haluaa logi-tiedostoihin tallentaa vai haluaako mitään.

Ramljak (2017) löysi OpenHABista useita turvallisuusongelmia, liittyen sekä Bindingeihin että turvattomaan oletuskonfiguraatioon. Erityisesti bindingeistä hän löysi staattisella koodianalyysillä suorituskykyongelmia, liian korkeita oikeuksia, validoimatonta käyttäjän syötettä, sekä OpenHAB REST -palvelun haavoittuvuuksia. Hän mainitsee, että sekä OpenHABista, että ainakaan Domoticz, OpenMotics ja Calaos -alustoista ei löydy aiempia turvallisuustutkimuksia, vaan ne ovat keskittyneet protokolliin ja laitteisiin, eivät koko systeemiin.

3.2.1 Asennus

OpenHABin asennus sujui ongelmitta, vaikka sivut välillä olivatkin sekavahkot. Kuitenkin jos tiesi, että haluaa asentaa Raspberry Pi:lle, niin pystyi löytämään tarvittavat ohjeet. Asennuksessa käytettiin suositeltua openHABian käyttöjärjestelmää, jossa on valmiina suurin osa tarvittavista ohjelmistoista, ja sen avulla on mahdollista asentaa etäyhteys suhteellisen automaattisesti.

Oletussalasanojen vaihtamiseen on ohjeet ja suositus openHABian-sivulla, mutta ne eivät ole erityisen silmille pomppaavia eikä normaalin asennuksen yhteydessä ollut mainintaa lainkaan.

Laitteiden asennus openHABiin sujui helposti ja hyvin. Lisätään aina oikea Binding, joka ainakin näissä testeissä löysi laitteet tämän jälkeen nopeasti ja oikein. Z-Wave-laitteet löytyivät myös suoraan; niissä tutkimuksesta jäi pois kontrolleriin lisäys, joka oli tehty jo aiemmin. Se kuitenkin on helppoa, painetaan vain tikun nappia ja sitten laitteen nappia. (Tai miten kukin laite nyt toteuttaakin lisäyksen.) Z-Wave-kontrollerin asennus on hieman vaativa. Sitä varten pitää selvittää käytetty Z-Wave-usb-tikun portti.

Z-Waven suhteen turvallisuus on huonosti toteutettu. Oletuksena vain 'Entry Control Devices' eli käytännössä lukot asennetaan salattuina. Kaikkien salausta ei edes suositella, vaan mainitaan sen vähentävän paristojen käyttöikää ja hidastavan kommunikointia; valideja pointteja toki. Vaihtamalla 'All Devices' salataan kaikki laitteet, jotka toteuttavat Security Command Class:in.

Kun laitteet ovat löytyneet, ne tulevat openHABin Paper UI:n things-valikkoon, josta kunkin laitteen halutut sensorit/aktuaattorit lisätään Itemeinä, jonka jälkeen ne tulevat Inboxiin. Kun ensimmäisen laitteen oppii lisäämään, tulevat muut helposti. Hieman monimutkaisuutta luo se, että laitteet pitää ensin löytää, sitten lisätä kokonaisena, ja sitten halutut Itemit. Toisaalta valinnanvapautta rajoittaisi, jos kaikki tulisivat automaattisesti; tämänkin saa kyllä päälle asetuksista.

Automaatiot luodaan tekstipohjaisesti, mistä miinusta käyttäjäystävällisyyteen. Syntaksi kuitenkin yksinkertaista, joten periaatteessa ovat suhteellisen loogisia tehdä, vaikka monimutkaisemmat säännöt vaativatkin ohjelmointia. Esimerkkisääntönä seuraava:

rule 'DoorOpenAllOn'
when
Item DCHZ110DoorWindowSensor_Status changed from Closed to Open
then
DCHZ510Siren_Switch.sendCommand(ON)
HueWhiteLamp1_Brightness.sendCommand(ON)
WeMoSwitch_Switch.sendCommand(ON)
end

Testeissä ovisensori ei kuitenkaan halunnut toimia, joten kokeiltiin korvata se 'puhelin on sisäverkossa'/'puhelin ei ole sisäverkossa' triggereillä, jotka toimivatkin oikein.

3.2.2 Etäyhteys ja turvallisuus

Etäyhteyteen on ohjeissa mainittu kaksi tapaa: SSH ja HTTPS. Näistä SSH on oletuksena vain localhostille, eli Raspberry Pi:stä itsestään pääsee. Tämä on eri yhteys kuin suora SSH, johon pääsee muiltakin koneilta, ainakin lähiverkosta. Tästä ei ole mainintaa ohjeis-

sa. SSH:n voisi ajatella sopivan hyvin etäyhteyden luontiin, mutta tällöin ei saada graafista käyttöliittymää ja laite on avoinna Internetiin, eli turvallisuus riippuu salasanan vahvuudesta tai sertifikaattien lujuudesta.

OpenHAB ei tarjoa oletuksena autentikointia HTTPS:n kautta, eli ei eri käyttäjiä. Aivan oletuksena ei myöskään edes toimi HTTPS portin 8443 kautta. Hallintapaneeli 8080 portilla eli HTTP:llä.

Ohjeissa kielletään avaamasta suoraan portteja Internetiin. Tällöin olisi vähintään yksi web-serveri-portti auki eikä niissä ole minkäänlaista suojausta. Asetuksista voi sallia vain tietyt IP-osoitteet, eli periaatteessa tällä tavalla voisi osittain suojata myös suoran Internet-yhteyden.

HTTPS:n sallivia etäyhteystapoja ohjeissa mainitaan kolme: VPN, openHAB Cloud -pilvipalvelu, sekä käänteinen välityspalvelin.

- VPN:stä mainitaan vain, että löytyy ohjeita Internetistä, mutta ilmeisesti suositelluin kun ensimmäisenä on.
- openHAB Cloud vaatii joko asennuksen itse omalle koneelle tai esimerkiksi Amazonin pilveen, tai sitten voi käyttää openHAB Foundationin tarjoamaa myopenHAB.org-palvelua. Oman serveri-instanssin asennus pilveen lienee käyttäjältä liikaa vaadittu, toki onnistuu ja ohjeet löytyy. MyopenHABia käyttämällä ei voi olla varma miten tietoja käsitellään jne. Tästä tutkielmasta pilven tutkiminen jäi pois seuraavista syistä: myopenHAB ei ole täysin käyttäjän hallinnassa, joten se jätettiin suoraan pois; kotikoneelle asennettuna ei hirveästi hyötyä liene, joten sekin ajatus pois; jäljelle jää asentaminen johonkin kolmannen osapuolen pilveen, mihin on kyllä hyvät ohjeet, mutta se maksaa ja on lisäksi suhteellisen teknistä. Jatkotutkimuskohteeksi kuitenkin erittäin hyvin soveltuva.
- Käänteisen välityspalvelimen asennukseen OpenHABian tarjoaa aktivointia vaille valmiin ’NGINX reverse proxyn’; aktivoitaessa asetetaan myös autentikaatio toimimaan sekä ’Let’s Encrypt’ -sertifikaatti. Tätä aktivointia varten on hyvät ohjeet. HTTPS:n käytössä on ehkä liikaa vaadittu että käyttäjällä olisi domain, mutta sekin onnistuu tietysti. Suoraan IP-osoitteellakin voi yhdistää, mutta tällöinkin pitää olla portti avoinna,

mitä ei toisaalta ohjeissa mainita erikseen. On kuitenkin tärkeää että käyttää HTTPS:ää, muuten turvallisuus on alhainen vaikka autentikaatio olisikin kunnossa. Asennetaan ohjeiden mukaisesti NGINX Reverse Proxy ja siihen OpenSSL, eli jätetään domain hankkimatta. Ohjeita seuraamalla onnistuu helposti. Käyttämällä OpenSSL:ää saadaan valitusta itse allekirjoitetusta sertifikaatista. Tämä voidaan kuitenkin jättää huomiotta, kun tiedetään että olemme sen itse tehneet. Hieman lisätyötä tuottaa kuitenkin.

Kun tämä on lisätty, voitaisiin avata reitittimestä Internetiin portit 80 ja 443, jotka ohjattaisiin NGINXille, joka sitten ohjaisi ne openHABille. Näin ollen turvallisuus olisi NGINXin turvallisuutta, joten jätetään se tutkimatta ja luotetaan siihen. Sisäverkosta päin openHABissa on kuitenkin auki portit 22,139,445,8080,8443 ja 9001, eli todella monta. Näistä portti 22 on SSH:lle, 139 ja 445 Samballe, 8080 ja 8443 ovat openHABin http-käyttöliittymille, sekä 9001 logien katselulle. Sisäverkosta näistä voidaan yhdistää kaikkiin ja HTTP- sekä logi-portit eivät vaadi mitään autentikointia. Nämä seikat vähentävät turvallisuutta. Esimerkiksi Samba ja SSH voidaan sammuttaa jos ei niitä käytetä, mutta HTTP-käyttöliittymää ei. Kannattaa siis luoda vieraille oma WiFi-verkko eikä päästää muita openHABin kanssa samaan. Tai optimaalisesti openHABille oma, johon pääsy vain NGINXin kautta. Tai helpommin sallitaan portteihin pääsy vain lokaalisti, eli käytännössä NGINXin kautta.

Ohjeiden avulla saatiin portti 8080 sallituksi vain localhostille, mutta 9001 on vieläkin auki, eli logit näkee. 8443 on samoin yhä auki kaikille. Sammuttamalla sisäinen https, eli portti 8443, saadaan aikaan tilanne, jossa enää vain 9001 on auki. Nyt sisäverkossakin liikenne menee NGINXin kautta, eli turvallisesti. Auki ovat enää portit 22, 80, 443 ja 9001. Vielä kun viimeinen saataisiin kiinni, niin turvallisuus paranisi. Oletusasetuksilla kuitenkin sisäverkon suhteen turvaton. 9001 portissa pyörii frontail, eli node.js sovellus joka striimaa logeja selaimelle. Siihenkin ilmeisesti saa turvallisuutta lisää, mutta ohjeissa ei mainintaa eikä välttämättä kovin helposti saa käyttäjä laitettua.

Olettaen, että käyttäjä osaa NGINXille avata oikean portin reitittimestä ja vain tämä on auki, on tällöin NGINXin turvallisuudesta kiinni asetelman turvallisuus. Autentikointi oli perus käyttäjänimi-salasana, minkä turvallisuudesta voi olla montaa mieltä. Brute forcea vastaan siihen saa IP-bannauksen, oletuksena sitä ei ole. Käyttämällä ulospäin

näkyvänä porttina jotain harvinaista porttia ja lisäksi käyttämällä hyvää salasanaa, ei tämä ole niin suuri ongelma, mutta palvelunestohyökkäys tietysti onnistuu.

3.3 HomeAssistant

HomeAssistant[2] on Python 3 -pohjainen ohjelmisto, joka sopii hyvin Raspberry Pi -pohjaiseksi kotiautomaatio-gatewayksi. Sillä on elinvoimainen yhteisö foorumin aktiivisuudesta päätellen; lisäksi sillä on mm. oma ala-Reddittinsä ja Discord-kanavansa. Sillä on parin viikon välein ilmestyvä podcast, jossa keskustellaan uusista ominaisuuksista yms.

HomeAssistantin uusin versio 0.55.1 on julkaistu 15.10.2017.

HomeAssistantilla on tuki 844 eri laitteelle. Se tukee mm. Zigbeetä, Z-Wavea ja Enoceania. Sen dokumentaatio on kattava ja ohjeita käyttöä varten löytyy paljon, esimerkiksi toisten käyttäjien tarjoamia malleja. Uusia laitteita on helppo lisätä, käytännössä vain kopioidaan valmis malli YAML-tiedostoon.

HomeAssistantissa on suora etähallintatuki, pilveä eivät kehittäjät tarjoa. Erilaisiin etähallintatapoihin ja niiden osiin on ohjeet; esimerkiksi suoraan yhteyteen porttiohjauksella, dynaamiseen DNS-palveluun, liikenteen salaamiseen Let's Encryptillä, self-signed sertifikaattiin, sekä Tor:in käyttöön etäyhteydessä.

Hass.io on käyttöjärjestelmä, joka hoitaa Home Assistantin asennuksen ja päivityksen; jonka avulla on helppoa hallita Home Assistantia; ja jota voidaan laajentaa lisäosilla.

HomeAssistant nojaa periaatteeseen, että älykodin hallinta ja data pysyvät käyttäjän käsissä, eivät pilvessä tai kolmansilla osapuolilla. Logi-tiedostoihin tallennettavat tiedot saa käyttäjä itse päättää.

3.3.1 Asennus

Asennettiin suosituksen mukaisesti tarkemmin ottaen hass.io, eli muun muassa Raspberry Pi:lle oleva valmis levykuva, jossa on Home Assistant ja sen konfiguroimiseen tarvitta-

2. https://home-assistant.io/

va web-käyttöliittymä. Asennettiin käyttöliittymän kautta SSH-serveri, johon suosittelevat public-keyn käyttöä.

WeMo ja Hue asentuvat itsestään ilman ongelmia. Z-Wave ei meinaa asentua suoraan ohjeiden mukaan, konfiguraatiosivukin ilmestyy yhtäkkiä. Uudelleenkäynnistys korjaa asian.

Automaation luonti vaatii useamman eri sivun välillä kikkailua. Hyvin yksinkertainen sääntö onnistui helpohkosti, tarvittiin vain tiedot haluttujen sensorien ID:istä sekä tieto, miten kaikki Switchit ja kaikki valot laitetaan päälle ja pois, sekä miten termostaatin lämpötilaa muutetaan. Voisi kuitenkin selkeästi olla helpompikin tapa, esimerkiksi saisi automaatiota luodessa valikosta valita halutun laitteen.

Z-Wavesta termostaatin poistaminen ei toimi kunnolla. Ei poistu, vaikka laite itse on sitä mieltä. Yritetään lisätä Securena, saman ID:n solmuja on lopulta 3. Onnistuu uudelleen lisäys, vanhat solmut vielä näkyvät. Sallii lisätä Securena vaikkei ole asetettu verkon salausavainta eikä laite edes tue salausta. Ei valita mitään eikä mainitse mitään, eli huono juttu. Logeja lukemalla saa jotain irti, mutta missään ei näy onko joku solmu lisätty turvallisena.

3.3.2 Etäyhteys ja turvallisuus

Home Assistantissa on tällä kokoonpanolla avoinna portit 22 (SSH), 8123 (WEB-käyttöliittymä), 8989 (WeMon joku), ja 22222 (toinen SSH, suoraan laitteelle, ei vain hass.io:lle, sisään vain public keyllä). Turvallisuutta lisää hieman se, ettei käytetä pelkästään yleisiä portteja, vaan tässä tapauksessa käyttöliittymä on portin 8123 takana. SSH kylläkin on 22.

Web-käyttöliittymän puolelta en löydä turvallisuusasetuksia.

Sanotaan, että kerätty data säilötään vain lokaaliin instanssiin.

Löytyy GitHubin Issueseista salasanan 'remember me':hen liittyvä turvallisuusriski, eli tallennetaan selkokielisenä localstorageen. Tähän auttaisi paljon jo se, että myös sisäverkossa olisi salattu yhteys käytössä.

Sivuilta löytyy Securing-osio, jossa on checklist turvallisuusjutuille:

- HTTP-hallinnan salasana, erittäin tärkeä.

- Isäntäkoneen itsensä turvallisuus.
- Verkkoyhteyksien rajoittaminen, erityisesti SSH:lta pois rootlogin ja avaimien käyttö salasanojen sijaan.
- Ei ajeta HA:ta roottina. Tähän ei voinut itse vaikuttaa Hass.io:n tapauksessa.

Etäyhteyden tapauksessa:

- Käyttää TLS/SSL
- Käyttää Tor
- Käyttää self-signed certificatea
- Käyttää proxya

Näistä kaikkiin on ohjeet, lukuunottamatta ssh-avainten käyttöön sekä ei-roottina-ajamiseen.

Etäyhteyteen annetaan muutama eri keino:

- Käyttämällä Tor:ia ja sen Hidden Serviceä. Voi olla peruskäyttäjältä liikaa vaadittu, vaikka ohjeet löytyykin. Vaatii myös jokaiselle laitteelle, joka haluaa yhdistää Home Assistantiin, että siihen asennetaan Tor-mahdollisuus, esimerkiksi Orbot.
- Avaamalla portti suoraan. Tämän turvallisuudesta mainitaan vain lyhyesti, että se ei ole sitä. Syitä ei anneta.
- Käyttämällä DuckDNS:ää ja Let's Encryptiä antamaan pysyvä domain ja HTTPS-yhteys. Näistä jälkimmäinen takaa kuitenkin vain yhteyden salauksen, ei autentikointia.
- Löytyy myös eri paikasta mainintaa NGINX:illä toteutettavasta proxysta.

Jos käyttäjä sitä haluaa ja etsii, niin löytyy paljon ohjeita ja resursseja turvallisen HA:n asentamisesta. Oletuksena jää suhteellisen turvattomaksi eikä selkeästi kerrota eri vaihtoehtojen turvallisuudesta. Ei mainita esimerkiksi sitä, että lähes kaikki esitetyistä ratkaisuista riippuvat lopulta itse Home Assistantin autentikoinnista, eli käytännössä valitusta salasanasta. Ei ole brute forcen estoa, vaan kirjautumista saa yrittää niin usein kuin haluaa. (Vaikka löytyy kyllä lähdekoodia, jossa bannataan IP-osoite, josta tulee liikaa yrityksiä, mutta ilmeisesti ei ole käytössä.) Sallii myös esimerkiksi salasanan 'aab' eikä valita siitä.

3.4 Domoticz

Domoticz[3] on kotiautomaatiojärjestelmä, joka on kehitetty yksinkertaisuutta silmällä pitäen. Se toimii useissa eri käyttöjärjestelmissä ja sen käyttöliittymä on HTML5:ta ja tukee kaikkia selaimia. Sitä voi hallita myös mobiilisovelluksilla.

Sen viimeisin Stable-julkaisu on versio 3.8153, joka on julkaistu 30.7.2017.

Tuettuja laitteita on paljon, myös esimerkiksi Z-Wavelle, mutta Zigbee-tuki vaikuttaa olevan huono.

Dokumentaatio vaikuttaa olevan kattava, samoin foorumi on aktiivinen. Siltä löytyy hyvä manuaali, josta tärkeimmille toiminnoille löytyy ohjeet.

Domoticz tarjoaa etähallinnan ja lisäksi se antaa mahdollisuuden jakaa vain tiettyjä laitteita ja vain tietyille käyttäjille, eli esimerkiksi lämpötilatiedon sekä pihavajan lukon hallinnan voi jakaa kavereilleen, mutta ei muuta. Etähallinta onnistuu HomeAssistantin tyyliin eri turvallisuustasoilla, eli salaamattomalla tai salatulla yhteydellä.

Domoticz kerää laitteiden tietoa ja esittää niiden logit graafisesti. Logi-tiedostojen sisältöön ei voi vaikuttaa, mutta halutessaan ne saa pois päältä.

3.4.1 Asennus

Domoticz ei enää salli tai tarjoa valmiita levykuvia asennukseen, vaan täytyy asentaa jonkin käyttöjärjestelmän päälle. Ei sinänsä lisää monimutkaisuutta, mutta työtä kyllä. Ohjeet hieman vanhentuneet, mutta vaihtamalla vanhemman Raspbianin tilalle uudempi, toimii kyllä. Ainut ongelma libssl1.0.0:n puuttuminen uudemman version repositorysta, joten piti ladata erikseen, eikä ohjeita ollut olemassa. Kokonaisuudessaan tämä kuitenkin on tietoturvauhka, sillä se ei ole uusin openSSL:n versio ja siinä on useita haavoittuvuuksia.

Oletuksena käyttöliittymässä on sekä HTTPS- että HTTPS-portit käytössä, 8080 ja 443. Portit voisivat olla turvallisemmatkin, nyt HTTPS on oletusportissa ja 8080 on hyvin yleinen HTTP-proxy-portti.

3. https://domoticz.com/

Laitteiden asennus sujuu helposti. Z-waven tapauksessa usb-tikun porttikin on oletuksena oikein. Samoin Hue asentuu ongelmitta. WeMo:n asentaminen on vaikeampaa, mutta siihen löytyy ohjeet. Ohjeita joutuu vähän soveltamaan, periaatteessa asennetaan dummy-switch, joka suorittaa bash-skriptin, joka ohjailee pistorasiaa. Tässä tapauksessa suhteellisen helppoa, koska ohjeet ja skripti ovat valmiina.

Laitteiden asennuksen jälkeen pitää erikseen aktivoida niiden halutut sensorit/aktuaattorit, jolloin esimerkiksi ovi-/ikkunasensorin lämpötilasensoria ei tarvitse pitää näkyvillä. Eli voidaan valita vain tarvittavat, jolloin niiden valinta sääntöjä luodessa ja muutenkin pysyy selkeämpänä. Pientä ongelmaa voi tuottaa oikean sensorin/aktuaattorin valinta, joka pitää käytännössä tehdä testaamalla ja katsomalla tilojen muuttumista.

Domoticzin eventit luodaan Blockly:llä, eli graafisesti suhteellisen yksinkertaisesti. Valitaan sopivat palikat ja niitä yhdistelemällä luodaan haluttu sääntö. Säännön luominen on helppoa jos laitteiden sensorit/aktuaattorit on oikein lisätty ja nimetty selkeästi. Virheitä nimissä tai säännöissä ei pitäisi pystyä tekemään. Monimutkaisemmat säännöt tehdään LUA:lla.

Koska Domoticzistä voi helposti ajaa skriptejä, pitäisi mitä tahansa mitä Raspista itsestään pystytään tekemään, tekemään myös sen kautta. Skriptit ajettaneen käyttäjän oikeuksilla, eli jos löytyy root-oikeudet, niin huolimaton kopioitsija tehnee suurtakin tuhoa koneelleen.

3.4.2 Etäyhteys ja turvallisuus

Myöskään Domoticz ei mainitse mitään laitteiden turvallisuudesta puoleen tai toiseen. Huesta ei kai mitään sanomista olisi, mutta WeMot ovat käytännössä turvattomia sisäverkossa. Z-Wave-laitteista ei mainita, ovatko asennettuja turvallisesti vai ei. Uusia lisättäessä on turvaton lisäys ykkösvaihtoehtona. Mitkään ohjelmistot tähän asti eivät tunnu välittävän laitteiden, varsinkaan Z-Waven, turvallisuudesta.

Oletuksena käyttöliittymässä ei ole autentikointia. Se on lisättävä asetuksista, jotka tyhmästi vaativat samalla myös sijaintitiedot. Asetuksista voidaan myös asettaa osoitteet, joista autentikointia ei tarvita, yleensä sisäverkko. Muutamasta väärästä kirjautumisyrityksestä ohjataan 'Last page of the internet' -sivulle. Ei kuitenkaan IP-bannia oletuksena.

Voidaan asettaa eritasoisia käyttäjiä: viewer, user, admin. Eri käyttäjien eri näkymät vain piilotetaan, eli saadaan helposti näkyviin muuttamalla CSS:ää tai muita ominaisuuksia. Vääriin paikkoihinkin pääsee näin viewer- ja user-käyttäjillä sisään, mutta muutoksia ei voi tehdä siltikään.

Tarjoavat MyDomoticz-pilvipalvelua, jonka lähdekoodi ei kuitenkaan ole vielä avointa. Käyttäjiä (tai Domoticz-instansseja) sillä on noin 2000. Samoin tarjoavat Android ja ios-sovellukset. Näistä Android-versio toimii lähiverkossa suoraan, mutta ei ole rahkeita sanoa sen turvallisuudesta muuta. Vaikuttaa kyllä aika raakileen näköiseltä, mutta toiminee.

Etäyhteydestä ei ole manuaalissa muuta kuin että NAT:iin portit auki. Ei mainintaa salasanoista tai tämän etäyhteystavan turvallisuudesta. Missään ohjeissa ei selkeästi mainita, että salasana tulee asettaa ja sen tulee olla hyvä. Tai että käyttöliittymä on kaikkien tavoitettavissa.

Wiki-sivuilla on linkit eri turvallisuusohjeisiin etäyhteyteen liittyen. NGINX-proxyn sivut ovat kuitenkin poistuneet, sillä tuki tällaiselle yhteydelle pitäisi olla natiivisti mukana, mutta itse en löytänyt tai käsittänyt mitä tarkoittavat. Löytyy myös ohjeet fail2ban:in käyttöön, eli bannataan IP-osoitteet, jotka yrittävät kirjautua sisään liian monesti.

Tässä tapauksessa tuntuvat ohjeiden perusteella luottavan omaan kirjautumissivuunsa.

Oletuksena päällä sekä HTTP että HTTPS, jälkimmäisen kohdalla oletuksena sertifikaatti Domoticziltä, mistä selain valittaa ja käyttäjät myös. Ohjeet myös Let's Encryptin käyttöön, johon tarvitaan oma domain. Ja onnistuu tietysti asentaa vaikkapa NGINX tai muu testattu välittäjä.

3.5 Fhem

Fhem[4] on perl-pohjainen serveri kotiautomaatioon. Sillä on statistiikkasivujen mukaan Saksassa 3991 käyttäjää, Itävallassa 231, Sveitsissä 84, Alankomaissa 40, muissa maissa hyvin vähän, Suomessakin 0. Sitä on kehitetty ainakin vuodesta 2005 lähtien, aktiivisia kehittäjiä on sivujen mukaan tällä hetkellä 96.

4. http://fhem.de/fhem.html

Tuettua laitemoduuleja on yli 430. Nämä voivat olla sekä web- tai sovellussiltoja, kuten Spotify tai säätietopalvelu, että laitteita, kuten Hue tai Z-Wave. Kaikkiin tuettuihin moduuleihin on kattavat, joskin tekniset ohjeet.

Kerätty data jää käyttäjälle, joko logitiedostoihin tai tietokantaan, ja käyttäjä saa valita mitä loggaa.

3.5.1 Asennus

Perusasennukseen löytyy ohjeet, mutta tämän jälkeen ei selkeää tutoriaalia ole.

Löytää Z-Wave-tikun suoraan. Laitteiden 'esiintuonti' suhteellisen helppoa, mutta ei loogista mielestäni. Create node ja noden nimi Node List:istä. Miksei suoraan tuo kaikkia tai näytä listalla? Sireenin yhteys toimii, eli saadaan päälle ja pois; samoin termostaatin lämpötilaa voidaan säätää.

Hue saadaan ohjeiden avulla asennettua. Wemo vaatii enemmän työtä ja pitää käyttää ulkopuolisia skriptejä, mihin löytyy saksaksi ohjeita.[5] Yritetään, vaatii paljon dependenssien asennuksia, ja lopulta, kuten foorumin uusin viesti sanoo, ei onnistu. Se siis jää asentamatta.

Sääntöjen luominen (ja muukin) on komentorivipohjaisempaa ja käyttöliittymä ei ole hirveän käyttäjäystävällinen. Tietoa on paljon, mutta ei helposti sulatettavassa muodossa. Säännöt esimerkiksi luodaan komennoilla tyyliin 'define <NAME> notify <REGEXP> <command>', esimerkiksi 'ovisensori kiinni -> lämpö 25C' onnistuu komennolla

define lampo25 notify ZWave_SENSOR_NOTIFICATION_26:alarm:.*closed set ZWave_THERMOSTAT_28 setpointHeating 25

eli kun regexp sattuu kohdalleen, tehdään mitä käsketään. Tämä pitää tehdä joka kohdalle erikseen, eli ei voi yhdistää käskyjä. Kun ensin on luotu 'pohja' tällä käskyllä, aukeaa 'change wizard' eli jonkin sortin käyttöliittymä sääntöjen luontiin. Siinä ei kuitenkaan pysty esimerkiksi asettamaan 'alarm:.*closed' regexpiä, vaan pelkän 'alarm:.*', joten sillä ei esimerkiksi ylläolevaa sääntöä voi luoda lainkaan.

5. https://forum.fhem.de/index.php/topic,18524.msg248437.html

Saatiin kuitenkin ’kun ovisensori auki, niin laitteet päälle ja lämpö 5; kun ovisensori kiinni, niin laitteet pois päältä ja lämpö 25’ toimimaan suhteellisen helposti. Ohjeet piti etsiä googlella, vaikka fhemin wikissä lopulta olivatkin. Ne vain ovat lähes täysin saksaksi, joten selaamalla ei oikein löydä mitään.

3.5.2 Etäyhteys ja turvallisuus

Turvallisuuspuolelta ei liiemmin vaadita mitään salauksia tai salasanoja tai muita. On kyllä ohjeet basicAuth:in ja HTTPS:n käyttöön web-käyttöliittymässä sekä salasanan ja SSL:n käyttöön Telnetissä. Ohjeet komentorivi/’web-käyttöliittymän komentorivi’ -pohjaisia, vaikka sinällään kai helppoja. Samoin saa asetettua sallitut IP-osoitteet, jolloin vain niistä pääsee sisään. Ei sinänsä mainita että tarvitsee portteja ohjailla tai että näkyy kaikille tässä tapauksessa. Mainitaan myös VPN:n käyttö, sekä Apachen käyttö välityspalvelimena. Telnetin kautta ei päästä kunnolla Raspiin sisään, mutta löytyy esimerkiksi sammutus-, laitteen poisto-, ja nukkumiskomennot, joilla saataisiin tuhoa aikaan. Oletuksena kuitenkin täysin turvaton, eikä ohjeet ihan hirveän selkeät ole.

Voidaan tehdä myös eri käyttäjiä ja niille eri oikeuksia, jälleen vaikeahkosti komentoriviä käyttäen. Ohjeet kuitenkin selkeät, mutta eri oikeuksista ei ole listaa tarjolla. Käyttäjä, jolla on vain get/set oikeudet, pystyy sammuttamaan ja asettamaan kaikki laitteet päälle regexpillä set .* off(/on). Samoin regexpeillä get/set .* näkee halutessaan laajan listan eri objekteja.

Web-käyttöliittymä mainitsee turvallisuusongelmista kyllä etusivulla, ei hirveän vaativasti tosin.

3.6 Pimatic

Pimatic[6] on kotiautomaatioframework, joka on toteutettu node.js:llä. Sitä on kehitetty ainakin vuodesta 2013 eteeenpäin.

Foorumien perusteella yhteisö on vireä, GitHubin committien perusteella kehitys on hidastunut reilusti viime vuosina. Viimeisin julkaisu 0.9.35 on 2.11.2016.

6. https://pimatic.org/

Laitetuki on omien sivujen perusteella suhteellisen suppea, muutama kymmenen eri valmistajaa. Plug-in:eja sen sijaan on yli 150, joista jokainen tukenee vähintään yhtä laitetta, eli oikeasti on laajempi tuki kuin sanovat.

Dokumentaatio vaikuttaa kattavalta, samoin asennustutoriaalit.

Logien sisällön saa käyttäjä päättää, samoin asettaa pois päältä halutessaan.

3.6.1 Asennus

Asennus onnistuu, mutta pienillä vaikeuksilla. Jälleen Z-Wave tuottaa eniten ongelmia: OpenZWave piti kääntää itse, valmiiden käännösten ollessa vanhentuneita; tikun portti piti tietää, ei löytänyt automaattisesti; lisäksi laitteet vaativat säätämistä, mihin ei ollut ohjeita, eli itse piti googletella ja soveltaa. Ei välttämättä onnistuisi peruskäyttäjältä. Yleisesti laitteet eivät löytyneet helposti, Raspberry Pi:n uudelleenkäynnistys auttoi asiaa.

Automaatioiden luonti on helppoa, tällä kertaa ongelmana vain testilaitteiden huono tukeminen, eli esimerkiksi ovi-/ikkunasensorin auki/kiinni-tietoa ei tueta. Tehdään kuitenkin toimivana sääntö ’kun WeMo on päällä/pois, menee Hue ja sireeni päälle/pois ja lämpötila termostaatissa on 25/5’.

3.6.2 Etäyhteys ja turvallisuus

Ohjeissa mainitaan, että etäyhteys olisi oltava aina HTTPS:llä; HTTP mainitaan turvattomaksi, samoin se, että vain se on oletuksena päällä. Etäyhteyteen mainitaan keinoksi vain dynaaminen DNS ja suoraan portti auki reitittimeen. Ei mainintaa esimerkiksi IP-osoitteella pääsystä, VPN:stä, tai välityspalvelimesta. Tarjoavat kuitenkin sertifikaattien luomista varten valmiin skriptin. Eli ohjeiden mukaan tehtynä kaikki turvallisuus on itse Pimaticista kiinni, erityisesti sen käyttöliittymästä. Asennuksen yhteydessä tuli valita oletusadminin salasana, mutta sen vahvuudesta ei ollut mitään mainintaa. On mahdollista luoda käyttäjiä, joilla on vain lukuoikeus Pimaticiin, mutta ei mainintaa siitä, voiko vain näiden yhdistämisen etänä sallia, eli että admin-oikeuksilla ei voisi etäyhteyttä ottaa.

Ohjeiden ulkopuolisesti voi toki asentaa VPN:n tai käänteisen välityspalvelimen, mutta oh-

jeista nämä puuttuvat.

3.7 OpenNetHome

OpenNetHome[7] painottaa helppokäyttöisyyttä: se välttää skriptauskielen käyttöä, käyttöliittymä on Web-pohjainen ja sen pitäisi tukea myös automaattista konfigurointia laitteilta saatujen signaalien perusteella. Samalla se pyrkii siirrettävyyteen, eli kotiautomaatioserverin alustalaitteen tulisi olla vaihdettavissa helposti.

OpenNetHomea käytetään HomeItems-moduulien avulla. Niitä on tällä hetkellä yli 100. Zigbeetä tuetaan eri valmistajien bridgejen kautta (esim Philips Hue), mutta Z-Wave-tuki on vasta rakenteilla, vain perustoiminnot ovat käytössä. WiFi ja 433MHz -laitteiden tuki on laajempaa.

NetHomeServerin uusin versio 3.0 on julkaistu 31.7.2017. Githubin ja foorumin perusteella OpenNetHomen kehitys on yhä aktiivista, mutta yhteisö ei ole kovinkaan laaja ja kehitys tapahtuu lähinnä yhden henkilön toimesta.

Asennukseen Raspberry Pi:hin löytyy ohjevideo, samoin muutamien eri laitteiden lisäämiseen. Yleisesti ottaen dokumentaatio on kuitenkin vähäistä ja wikiä ei ole päivitetty useaan vuoteen, mutta foorumilla on tuorettakin keskustelua.

Logitiedostot ovat lokaaleja, eli datan keruu on täysin käyttäjän hallinnassa, ja niiden sisällön saa käyttäjä valita. Mitään pilvipalvelua ei tarjota.

3.7.1 Asennus

Asennus onnistuu helposti, eli laitteet löytyvät automaattisesti. Z-Wave vaatii usb-portin asettamista, jonka jälkeen solmut löytyvät, joskin hitaasti. Niiden automaattiset tiedot eivät kuitenkaan ole oikeat, esimerkiksi sireeniä pidetään lamppuna ja ovisensorista saadaan vain valoisuusarvo ulos. Termostaatti ei toimi lainkaan, vaikka se solmuna löytyykin.

Sääntöjen luominen jää puolitiehen. Ajastimia ja niille toimintoja löytyy, mutta esimerkiksi

7. http://opennethome.org/

intervalliajastin ei toimi testeissäni. Start-komento menee, muut eivät. Tavallisia 'kun tämä, tee tämä' -sääntöjä en onnistu luomaan. Eräs triggeri olisi raja-arvojen seuraaminen ja toiminnon tekeminen näiden ylittyessä/alittuessa, mutta tätä en voi testata. Yleisesti automatiikka ei kovin yksinkertaiselta vaikuta. Muutenkin OpenNetHome vaikuttaa selvästi keskeneräiseltä ja tiettyihin tarpeisiin kehitetyltä.

3.7.2 Etäyhteys ja turvallisuus

Etäyhteyteen en löydä minkäänlaisia ohjeita. Peruskäyttöliittymä on HTTP:n takana, joten suoraan avattu portti ei ole turvallinen millään tavoin. Mitään autentikointia ei ole mahdollista asettaa. Jokin HomeCloud Connection löytyy, mutta ei tietoa mikä on.

Foorumilta löytyy kehittäjän kommentti[8], että tarkoituksellisesti ei ole lisännyt mitään turvallisuus- tai kirjautumisominaisuuksia, koska on vaikeaa saada niitä varmasti turvalliseksi. Ehdottaa sen sijaan reverse proxyn käyttöä, eli Apache/NGINX, joihin löytyykin samasta keskustelusta ohjeet.

3.8 ioBroker

IoBroker[9] on aktiivinen ja vaikuttaa toimivalta, varsinkin visuaaliselta puolelta, mutta yhteisö ja ohjeet painottuvat Saksaan. Sitä on kehitetty ainakin vuodesta 2014 asti. Foorumin ja GitHubin committien perusteella vireä yhteisö.

Dokumentaatio ja tutoriaalit vaikuttavat ainakin saksankielisinä kattavilta, mutta englanninkielinen puoli on selvästi vähäisempi, mutta sitäkin löytyy. Se on muutenkin reilusti saksalainen, statistiikkasivujen mukaan 7077 saksankielistä, 469 englanninkielistä ja 329 venäjänkielistä käyttäjää. Käyttäjämäärissä isoimpana lukuna 9083, mitattuna 10.11.2017.

Todella paljon eri adaptereita, jopa muille gateway-ohjelmistoille, kuten openHABille ja pimaticille, eli on oikeastaan kokonainen IoT-platform. Tällä hetkellä adaptereita on 174.

Loggaa tietoja, mutta muokattavuudesta ja pois päältä asettamisesta ei löydy infoa.

8. http://forum.opennethome.org/viewtopic.php?p=462
9. http://www.iobroker.net/

3.8.1 Asennus

Asennus sujuu ohjeita noudattamalla, mutta sivujen kieli vaihtuu aina välillä saksaan, joten hieman joutuu näkemään vaivaa löytääkseen oikeat ohjeet. Englanninkieliset ohjeet ovat lisäksi vanhentuneita, mutta asennus toimii silti samalla periaatteella. Asennuksen jälkeen ei suoraan toimi, pitää ajaa sudona tai uudelleenkäynnistää Raspberry Pi.

Automaattinen laitteiden etsintä löytää laitteista vain Huen. Z-Wave-adapterin asennus vie hyvin kauan eikä ilmoita edistymisestään mitään. Samaa voisi olettaa muiltakin adaptereilta. Jälleen pitää asettaa Z-Wave-tikun usb-portti ja tällä kertaa myös ajaa lisäkomentoja, jotka löytyivät ohjeista. WeMo:lle ei löydy adapteria lainkaan.

Käyttöliittymä täynnä tavaraa, alussa jopa tukahduttavan paljon. Käyttöliittymiä voi rakentaa erilaisia, ja esimerkkejä ja valmiita pohjia on paljon.

Automaation luontiin tarjotaan Blockly, kuten Domoticzissa. Itse en kuitenkaan onnistu tekemään mitään oikeasti toimivaa. Laitteiden tilat haetaan eri tavalla kuin vaikkapa domoticzissa, ollen paljon teknisempää (ja ilmeisesti toimimattomampaa). Asettamani sääntö toimi kerran, sitten ei enää. Eikä yksinkertaista 'aseta laitteen tila tähän' palikkaa löytynyt.

Selvänä ongelmana ohjeiden painottuneisuus saksaan ja englanninkieliset ohjeet ovat joko vajavaiset, vanhentuneet tai molempia. Muuten vaikuttaa toimivalta ja laajalta.

3.8.2 Etäyhteys ja turvallisuus

Tarjoavat jonkinlaista pilvipalvelua, mutta ei ole avointa lähdekoodia, joten unohdetaan se.

Käyttäjiä ja käyttäjäryhmiä voi lisätä ja niille tarjotaan sisäänkirjautuminen, jossa on bruteforcen testaus. Oletuksena ei ole mitään käyttäjänhallintaa eikä sisäänkirjautumista. Se asetetaan Admin-adapterin kautta, mistä saadaan myös HTTPS päälle, mutta vaatii toki sertifikaattien väsäilyä (oletuksena löytyy defaultCert, mutta ei kannattane käyttää). Käyttäjävalikko pitää lisätä näkyviin, jonka jälkeen käyttäjiä ja heidän roolejaan on helppo luoda ja muokata. Erikoisesti autentikointia ei voi asettaa ilman HTTPS:ää, mutta toisaalta ymmärrettävää. Kirjautunut 'user' näkee aika paljon tietoja, oletuksena jopa enemmän kuin admin, mutta ei todennäköisesti suurinta osaa voi säätää. Testeissä ei onnistunut asetusten vaihto

ainakaan.

Etäyhteyteen en löydä ohjeita englanninkieliseltä puolelta, enkä käännöspalveluiden avulla saksankieliseltäkään puolelta. Ainut maininta on ioBroker Cloudista, mutta yllä olevien syiden takia jätän väliin. Ilmeisesti Cloudilla ei edes voi säätää mitään, näkee vain visuaaliset jutut. Toki etäyhteyden pitäisi onnistua käänteisellä välityspalvelimella tai muulla etäyhteystekniikalla, mutta ohjeet olisivat plussaa.

3.9 Muita ohjelmistoja

Löydettiin myös muita kotiautomaatio-ohjelmistoja, mutta ne päätettiin alkukarsia tutkimuksesta muun muassa ei-aktiivisen kehityksen ja yhteisön, liian teknisyyden, ja/tai vähäisen laite-/protokollatuen vuoksi. Varsinkin aktiiviset, mutta esimerkiksi huonon dokumentaation omaavat ohjelmistot olisivat hyvää jatkotutkimusaihetta, riippuen niiden kehityksestä ja tutkimuksen painotuksesta. Koska tässä tutkimuksessa painotus oli käyttäjäystävällisyydessä ja turvallisuudessa, jäi pois varmasti teknisesti toimivia ohjelmistoja, jotka kuitenkaan eivät ole helppoja käyttää.

- Open Source Automation[10]: Windowsille suunniteltu, joten ei sovellu open source -ajatukseen niin hyvin.
- OpenSmartHub[11]: Ei päivitetty yli kahteen vuoteen; lisäksi suunniteltu pilvipohjaiseksi ja Azurea käyttäväksi, vaikka toimii myös ilman niitä lokaalisti. Myös laitteistotuki on suhteellisen vähäinen.
- The Thing System[12]: Ei aktiivinen. Viimeisin tarjottava versio 1.10 julkaistu 25.8.2014, GitHubin viimeisin päivitys yli vuoden vanha.
- DomotiGa[13]: Ei aktiivinen. Viimeisin repository päivitys 30.12.2016. Keskustelupalsta ei myöskään aktiivinen.

10. https://github.com/opensourceautomation/Open-Source-Automation
11. https://github.com/OpenSmartHub/OpenSmartHub
12. http://thethingsystem.com/index.html
13. https://www.domotiga.nl/projects/domotiga/wiki/Home

- MisterHouse[14]: Ei kovin aktiivinen. Viimeisin kotisivuilta löytyvä Stable-versio 3.1 on julkaistu 31.3.2014, GitHubista löytyy v.4.2 julkaistuna 7.3.2017. Wikiä ei ole päivitetty hetkeen. Lisäksi ohjelmisto vaikuttaa suhteellisen tekniseltä, sen konfiguroimisen perustuessa pitkälti Perl-koodin kirjoittamiseen. Näin ollen se ei ole erityisen käyttäjäystävällinen, vaikka toisaalta dokumentaatiota on paljon.
- Calaos[15]: Calaos on sivujensa mukaan täysimuotoinen ratkaisu kotiautomaatioon. Sen kehitti alunperin samanniminen ranskalainen firma, mutta firman sulkeuduttua vuonna 2013, sen lähdekoodi annettin avoimeen käyttöön GPL-lisenssillä, minkä jälkeen se on jatkanut kasvua yhteisön avulla. Foorumi on lähes täysin ranskankielistä keskustelua ja sen aktiivisuus ei ole kovin korkea. Tästä päätellen käyttäjämäärät ovat vähäiset tai vähenemässä. Viimeisin Stable-julkaisu on yli 2 vuotta vanha (20150215) v2.0, mutta GitHubista löytyy v3.0 alpha-versio, joka on julkaistu 21.9.2017. Calaos on suunniteltu erityisesti uuteen taloon tai asuntoon, jossa kaikki laitteet voidaan kytkeä langallisesti WAGO PLC-laitteeseen. Sillä on myös tukea muille protokollille, kuten X10:lle, sekä langattomillekin laitteille, mutta se on paljon rajoitetumpaa. Esimerkiksi Zigbeetä ja Z-Wavea ei ilmeisestikään tueta. Tuetuista laitteista mainitaan vain Premoboard, Cubieboard, Raspberry Pi, sekä Intel 32- ja 64-bittiset laitteet. Dokumentaatiota ja tutoriaaleja on jonkun verran, mutta suhteellisen vähän varsinkin englanniksi. Foorumit ovat lähes kokonaan ranskaksi. Koko järjestelmästä on vaikeaa löytää tietoa, etähallinta onnistunee avaamalla Calaos:ia pyörittävään koneeseen suora yhteys, mutta sen mahdollisesti tukemista salauksista tai muista en löytänyt tietoa. Kuitenkin avoimuutensa takia Calaos kerää tietoa vain käyttäjän haluamalla tavalla.
- AgoControl[16]: Ollut aktiivinen vielä jokin aika sitten, mutta esimerkiksi foorumilla on vuodelta 2017 vain muutamia keskusteluja. Toisaalta nettisivujen blogin mukaan tämä johtuisi vain ohjelmiston toimivuudesta ja valmiudesta. GitHubin committeja on vuodelta 2017.
- IoTOne: Gyory ja Chuah (2017) esittelemässä IoTOne-alustassa on yhdistetty OpenHAB- ja Samsung Smartthings -yhteensopivuus yhden alustan alle. Kirjoittajat sanovat sen lisäksi tukevan kaikkia IoT-valmistajia jotka tukevat avoimen lähdekoodin ympäris-

14. http://misterhouse.sourceforge.net/
15. https://calaos.fr/en/
16. https://www.agocontrol.com/

töjä tai kolmannen osapuolen applikaatioita. SmartThings ei kuitenkaan ole avointa lähdekoodia, mikä jo itsessään rajaa IoTOne:n pois.

4 Pohdinta ja suositukset

Kaikki tutkitut ohjelmistot toimivat ainakin jollain tasolla: laitteista ainakin osa saatiin asennettua ja jonkinlaisia automaatioita luotua. Laitteiden lisäämisen helppous vaihteli: suurin osa löysi ainakin jotain automaattisesti (useimmiten Hue-laitteet), usein jotain joutui asentamaan manuaalisesti. Osa ohjelmistoista toi näkyviin automaattisesti kaikki löydetyt sensorit ja aktuaattorit yms, osassa haluamansa piti vielä lisätä erikseen niin, että niitä pystyi käyttämään automaatioissa ja että ne tulivat näkyviin käyttöliittymään. Tämän suhteen en voi antaa suosituksia, sillä jotkut pitävät ensin mainitusta tavasta, jotkut jälkimmäisestä.

Ohjelmistoja on mahdotonta laittaa paremmuusjärjestykseen tuettujen laitteiden suhteen, sillä tutkimuksessa oli mukana vain pieni määrä laitteita, eikä niiden tuettavuudesta voida päätellä mitään yleisemmästä laitetuesta. Tutkimuksen laiteasennuksista voidaan lähinnä vain päätellä yleinen laitteiden asennuksen helppous ja toimivuus. Hyvin pitkälti laitteiden ja gatewayn valinta riippuvat siis toisistaan: jos laitteita on valmiina, kannattaa valita ohjelmisto joka niitä varmasti tukee; jos ei ole valmiina, voi gatewayn valinnan tehdä muiden seikkojen perusteella, mutta tämän jälkeen laitteet tulee valita siihen sopivasti.

Zigbee jäi tutkimatta kokonaan, tutkimuslaitteiden ollessa tukemattomia ohjelmistoissa. Tämä ei sinänsä ollut ongelma, sillä modulaarisuutta voitiin testata Z-Wave:llakin yhtä lailla.

Käyttäjäystävällisyyden testauksen suhteen on vaikeaa olla objektiivinen: vaikka omasta mielestäni asennus onnistui helposti ohjeita seuraamalla, saattoi siinä silti olla ongelmakohtia, joita minun oli vaikea huomata. Suurimmalta osin uskoisin kuitenkin asennuskokemuksen olevan laajennettavissa. Käyttöliittymän helppouden suhteen lähinnä fhem:iä en lainkaan suosittele: se on selkeästi teknisempi kuin muut, eikä ulkoasua ole oletettavasti suunniteltu helppous edellä.

Turvallisuus laitteiden osalta oli kaikissa ohjelmistoissa olematon. Esimerkiksi kenen tahansa mahdollisuudesta ohjata WeMo:n pistorasiaa sisäverkossa ei ollut mainintaa, vaikka toisaalta se ei ilman tätä toiminnallisuutta toimisi lainkaan. Myöskään siitä, olivatko Z-Wave-laitteet asennettu turvallisesti vai ei, ei yksikään ohjelmisto maininnut mitään. Samoin salausta tukemattoman laitteen asentaminen salattuna onnistui ilman virheilmoituksia tai mai-

nintaa siitä, että laite ei oikeasti ole turvallinen. Täten voisi olettaa, että salausta tukevankin laitteen asennuksessa tulevasta virheestä ei mainittaisi mitään ja se asentuisi turvattomana, käyttäjän luullessa toisin. Tämä saattaa olla itse protokollan tai mahdollisesti OpenZWave-ohjelmiston puute. Mikään ohjelmisto ei maininnut mitään Z-Wave-liikenteen salauksesta, salausavaimen asettamisesta tai turvallisuudesta muutenkaan. Avaimen pystyi kyllä asettamaan ja jos turvallista Z-Wave-liikennettä halusi käyttää, löytyi siihen useimmiten ohjeet, mutta selkeämpää käyttäjälle ilmoittamista pitäisin järkevänä. Voitaneen olettaa, että muidenkin protokollien suhteen turvallisuus on tällä tavalla huonosti hoidettua.

Voidaanko tätä kuitenkaan pitää itse ohjelmiston puutteena? Onko sen tehtävänä pitää huolta, että käyttäjä tietää laitteiden tietoturvariskit, varsinkaan jokaisen yksittäisen laitteen kohdalla? Jonkinlainen automaattinen haavoittuvuushaku voisi olla mahdollinen, mutta oikea haavoittuvuustestaus ei tietenkään voi olla gateway-kehittäjien harteilla.

Useat ohjelmistoista mainitsevat kehittämisen perustaksi sekä eri valmistajien ja protokollien yhteistoiminnan, että datan pysymisen käyttäjän hallinnassa. Kaupallisiin tuotteisiin verrattuna nämä täyttyvätkin erittäin hyvin, ja tältä osin avoimen lähdekoodin gatewayta on erittäin helppoa suositella. Käytännössä ainoat kompromissit, mitä kaupallisiin tuotteisiin verrattuna joutuu tekemään, ovat vaihtelevan suuruinen asennuksen ja käytön vaikeuden kasvu, sekä mahdollisesti tuen puute. Samaan aikaan tuettujen laitteiden määrä, datan hallinta, yksityisyys, turvallisuus, sekä avoimuus kasvavat huomattavasti.

Aiemmin esittämäni älykodin gatewayn vaatimukset täyttyivät seuraavilla tavoilla:

- **Mahdollistaa laitteiden välisen kommunikoinnin, niiden käyttämistä teknologioista riippumatta.** Tämä toimi kaikissa tutkituissa ohjelmistoissa siltä osin, kuin ne eri teknologioita ja laitteita tukivat. Vaikka tutkimuksessa käytetyt laitteet eivät kaikissa ohjelmistoissa toimineet kunnolla tai ollenkaan, ei sitä voida suorilta käsin pitää miinuksena, sillä ne saattavat tukea jotain toisia laitteita, joita taas toiset ohjelmistot eivät tue. Tämän suhteen ainoa suositus on valita ohjelmisto, joka tukee käytettäviä laitteita.
- **Mahdollistaa laitteiden hallinnan sekä lokaalisti että haluttaessa myös etänä.** Kaikki ohjelmistot sallivat lokaalin hallinnan. Etäyhteyteen annettujen ohjeiden suhteen on eroja: Vain openHAB ja Home Assistant tarjoavat hyvät ja monipuoliset ohjeet. Myös

Fhem tarjoaa ohjeet jos niitä osaa etsiä, mutta vaativat hieman enemmän itse tekemistä. Muissa on puutteita: IoBrokerille en löytänyt ohjeita lainkaan; OpenNetHomen ainoat ohjeet löytyvät foorumilta; Pimatic ohjaa käyttämään suoraa yhteyttä dynaamisella DNS:llä; Domoticz ohjeistaa manuaalissa käyttämään suoraa yhteyttä, wikissä muitakin ohjeita, mutta vanhentuneet tai poistuneet. Eräs selkeä puute kaikissa on se, etteivät ne korosta sitä, että loppujen lopuksi etäyhteyden turvallisuus on salasanan vahvuudesta kiinni. Tämä ei toki päde sertifikaateilla yhdistämiseen. Kuitenkin riippumatta ohjelmistosta on etäyhteys mahdollista tehdä turvalliseksi esimerkiksi VPN:llä, eli siinä mielessä mikään näistä puutteista ei ole este kyseisen ohjelmiston käytölle.

Erilaisia pilvipalveluita tarjosivat openHAB, Domoticz, sekä ioBroker. Näistä vain openHAB:in pilvi oli avointa lähdekoodia ja sen voisi asentaa myös omalle palvelimelleen.

Ohjelmistojen mobiilisovelluksia ei tässä tutkimuksessa lopulta huomioitu. Niitä löytyi vaihtelevan laatuisina ja sekä ilmaisina että maksullisina, mutta ne jätettiin ajan puutteen ja suhteellisen vaikean tutkittavuuden vuoksi pois. Älypuhelimella pystyy kuitenkin käyttämään samaa Web-käyttöliittymää kuin tavallisella tietokoneellakin, eli sovellus ei ole pakollinen.

Tämän suhteen suositusta on vaikeampi antaa: Jos tietää mitä tekee, kaikkiin saa turvallisen etäyhteyden; jos osaa vain seurata ohjeita, on järkevintä valita openHAB tai Home Assistant. openHABin pilvenkin asennukseen on hyvät ohjeet ja toisaalta sen voisi myös tarjota käyttäjille esimerkiksi vuokranantajan kautta.

- **Mahdollistaa automaattisten toimintojen asettamisen sensoridatan tai esimerkiksi kellonajan perusteella.** Automaatioiden asettaminen onnistui lähes kaikilla ohjelmistoilla, hieman vaihtelevalla helppoudella ja monipuolisuudella. openHAB, Home Assistant ja fhem hoitavat automaatioiden luonnin enemmän tekstipohjaisesti; pimatic, Domoticz, OpenNetHome ja ioBroker tarjoavat graafisemmat välineet. Tekstipohjaisuus ei välttämättä tarkoita vaikeampaa luontia.

  OpenNetHomessa ja ioBrokerissa en saanut automaatioita toimimaan kunnolla tai lainkaan. Tämä ei tarkoita, etteikö niitä voisi saada toimimaan, mutta ainakin selkeästi ne

toimivat huonommin ja olivat vaikeampia tehdä kuin muissa. Myös fhemissä luonti oli vaikeaa, mutta siinä ne lopulta toimivat.

Suosituksena openHAB, Home Assistant, Domoticz ja pimatic, riippuen siitä millainen sääntöjen luontitapa käyttäjälle tuntuu luontevimmalta.

- **Mahdollistaa eritasoiset käyttäjät.** Käyttäjähallintaa ei openHABissa, Home Assistantissa, eikä OpenNetHomessa ollut, mikä ei kuitenkaan välttämättä tarkoita, että kaikki käyttäjät voivat tehdä aina mitä tahansa, sillä web-käyttöliittymässä voi olla rajoitteita ja esimerkiksi admin-toimenpiteet pitää joskus tehdä suoraan laitteessa itsessään. Mutta silti käyttäjänhallinnan puute on miinusta näille ohjelmistoille.

  Domoticz, ioBroker, Pimatic ja fhem tarjoavat jokainen käyttäjähallinnan ja eritasoisia käyttäjäprofiileja, joten näiltä osin ne ovat suositeltuja.
- **Mahdollistaa käyttäjien autentikoinnin.** openHABia ja OpenNetHomea lukuunottamatta kaikissa ohjelmistoissa on valmiina jonkinlainen autentikointi. On kuitenkin todennäköisesti turvallisempaa käyttää etäyhteyden kohdalla jotain muuta autentikointimenetelmää, kuten käänteisen välityspalvelimen autentikointia tai VPN:ää. Ja jos gatewayn käyttöliittymään saa sisäverkosta suoraan yhteyden, eivät nämä autentikoinnit tällöin auta mitään, jolloin tarvitaan ohjelmiston omaakin autentikointimenetelmää. Näin ollen openHAB ja openNetHome ovat ainoat, joita ei voi suositella tämän kohdan perusteella.
- **Mahdollistaa laitteiden autentikoinnin.** Tätä ei yksikään ohjelmisto toteuta tai ainakaan ei mainitse siitä. Tämä on täten mahdollinen haavoittuva osa kaikissa.
- **Mahdollistaa uusien laitteiden lisäämisen sekä vanhojen poistamisen.** Tämän toteuttavat kaikki. On mahdotonta sanoa onko joidenkin yksittäisten laitteiden kohdalla ongelmia, mutta yleisesti tässä ei ole puutteita.
- **Pitää itsensä päivitettynä.** Yksikään ohjelmisto ei päivitä itseään automaattisesti, mutta päivittäminen on kaikissa käytännössä yhden komennon päässä. Tästä syystä suositusta ei voi antaa.
- **On mahdollista lisätä laitteiden lisäksi esimerkiksi uusia yhteysteknologioita, esimerkiksi usb-porttiin liitettävillä lisälaitteilla.** Kaikki ohjelmistot tukivat tätä, ainakin Z-Waven osalta. Muutenkin ne ovat modulaarisia ja niihin saa lisättyä erilaisia lisä-

laitteita. Suositusta vaikea antaa, valinta tulisi tehdä tarvittavien yhteysteknologioiden ja niiden tuen perusteella.

- **Pitää huolta laitteiden päivityksistä, joko itse tai sallimalla laitteelle yhteyden Internetiin päivitysserverille.** Tämä jäi testaamatta, sillä kaikki tutkimuksen laitteet olivat päivitettyjä, eikä näin ollen päivitystarvettakaan ollut. Todennäköisesti ne eivät kuitenkaan laitteiden päivityksestä huolehdi, vaan antavat laitteiden itse tehdä sen.
- **Kerää logitietoja laitteiden toiminnasta. Toisaalta sallii kaiken tiedon keräämisen asettamisen pois päältä.** Kaikki ohjelmistot sallivat logien sisällön valitsemisen ja pois päältä asettamisen, lukuunottamatta ioBrokeria, joka ei niitä implisiittisesti mainitse, mutta tukenee kuitenkin; sekä Domoticzia, joka ei salli sisällön valitsemista.
- **Ilmoittaa käyttäjälle virheistä.** Lyhyen testausajan takia on mahdotonta sanoa tämän kohdan toteutumisesta. Useimmiten virheet löytyivät vain logeista eikä niitä tuotu käyttäjälle esiin, mikä toisaalta on hyväkin asia, ainakin jos virheet eivät ole kriittisiä.
- **Tutkii kotiverkon liikennettä ja havaitsee siitä hyökkäyksiä sekä suorituskykyongelmia.** Yksikään ohjelmisto ei toteuttanut tätä, mikä oli odotettua.
- **Tekee kaiken yllämainitun turvallisesti ja käyttäjäystävällisesti.** Käyttäjäystävällisyys vaihteli ohjelmistojen välillä, ja tietyn pisteen jälkeen sen arviointi itsessäänkin on vaikeaa. Ainoastaan fhem:iä en suosittelisi käyttäjäystävällisyyden suhteen lainkaan, muissa taso vaihtelee.

  Turvallisuudesta langattoman liikenteen osalta on mahdotonta sanoa varmasti mitään, eikä se useimmiten edes ole mitenkään gatewayn hallinnassa, vaan riippuu käytetyistä laitteista ja niiden teknologioista. Joissain niistä ei ole minkäänlaisia turvallisuusratkaisuja edes olemassa, joissain ne ovat käyttäjän käsissä, joissain harvoissa ne ovat täysin automaattisia ja turvallisia. Ja loppujen lopuksi langattoman liikenteen voi aina kaapata ja protokollissa voi olla haavoittuvuuksia.

  Uusien laitteiden suhteen mikään ohjelmisto ei automaattisesti lisännyt uutta laitetta verkkoon, mikä on oikea ratkaisu.

  Eräs haavoittuvuusmahdollisuus on ohjelmistoihin lisättävät lisäosat, käyttäjien jakoon antamat valmiit konfiguraatiot, yms. Tietysti avoimen lähdekoodin ohjelmistossa on pienempi todennäköisyys siihen, että nämä mitään haitallista tekisivät ja jonkinlai-

nen tarkistus kaikille varmaan tehdään, mutta mitä isompi osanen, niin sitä vaikeampi on varmistua sen turvallisuudesta. Myös itse ohjelmistojen päivitysten aitouden varmistaminen on turvallisuuden kannalta tärkeää.

Täysin varauksetonta suositusta ohjelmistosta on vaikea antaa, sillä kaikissa on puutteita ja eroavaisuuksia, ja kaikki vaativat ainakin vähän teknistä osaamista, erityisesti asentamisvaiheessa. Helpointa on kuitenkin kallistua openHABin ja Home Assistantin, sekä osittain myös Domoticzin ja Pimaticin suuntaan. openHAB ja Home Assistant sisältävät molemmat kattavan dokumentaation ja paljon ohjeita, ne tukevat suurta määrää protokollia ja laitteita, ja niiden yhteisöt ovat vireitä ja kehittäminen aktiivista, joten apua ja tukea löytyy. Lisäksi openHABista löytyy mahdollisuus asentaa avoimen lähdekoodin pilvipalvelu (esimerkiksi kaupungin toimesta), mikä lisäisi etäyhteyden helppoutta käyttäjille huomattavasti. Selkeänä heikkoutena näissä molemmissa on tällä hetkellä käyttäjänhallinnan puute, mutta sen lisäämistä voinee pitää todennäköisenä tulevaisuudessa. Domoticz ja vähäisemmässä määrin Pimatic saavat pisteitä samoissa kategorioissa, mutta vähemmän. Niiden selkeä etu on kuitenkin käyttäjänhallinnan olemassaolo.

Loppujen lopuksi täysin turvallista kotiautomaatiojärjestelmää on erittäin vaikea tai jopa mahdoton rakentaa. Etäyhteyden turvallisuus on tärkeimpiä huolehdittavia asioita, mutta esimerkiksi eri laitteiden rajaaminen eri verkkoihin (esim VLAN:eihin) on myös hyödyksi, jotta automaatioon kuulumattomiin laitteisiin, kuten esimerkiksi pöytätietokoneisiin, ei kohdistu uhkia jos jokin älylaite sisältää haavoittuvuuksia, joita jotenkin päästään hyödyntämään. Toisaalta tämä toimii myös toisinpäin, eli ehkä jopa todennäköisin kanava gatewayn takana olevien älylaitteiden saastumiseen on juuri samassa verkossa olevat saastuneet älypuhelimet tai PC:t. Tämänkin takia olisi hyvä pitää kotiautomaatioverkko erillisenä, vaikka gatewayn itsessään tuleekin olla tällöin kahdessa verkossa.

5 Yhteenveto

Tutkimuskysymyksenä oli ’Miten hyvin avoimen lähdekoodin gateway-ohjelmistot soveltuvat älykotiin, erityisesti turvallisuuden suhteen? Miten ne vertautuvat kaupallisiin tuotteisiin?’. Älykotiin soveltuvuus todettiin vaihtelevan tasoiseksi, mutta keskimäärin hyväksi tai erittäin hyväksi. Vähintäänkin pientä teknistä osaamista ohjelmistoista vaativat kaikki, sekä asennuksen että vähäisemmässä määrin käytön suhteen. Kaupallisiin tuotteisiin verrattuna niiden toiminnallisuus oli monipuolisempi, sillä kaikilla tutkituilla ohjelmistoilla pystyttiin käyttämään eri valmistajien ja eri protokollien laitteita, ja yksityisyyden osalta pystyttiin valitsemaan miten kerättyä tietoa käytettiin ja tallennettiin. Näiden osalta useimmat kaupalliset tuotteet ovat enemmän tai vähemmän rajoitettuja.

Turvallisuuden puolelta kiinnitettiin erityisesti huomiota oletusasetuksiin ja etäyhteyden toteutukseen. Oletusasetukset olivat kaikissa enemmän tai vähemmän turvattomat: hallintapaneelin liikenne oli salaamatonta ja ei-autentikoitua, eli sisäverkon suhteen täysin turvatonta, eikä tästä ollut selkeää mainintaa yhdessäkään ohjelmistossa. Pitkälti oletusasetukset olivat suunniteltu luotettuun sisäverkkoon asennukseen, mihin ne suhteellisen hyvin sopivatkin.

Ohjeet etäyhteyden muodostamiseen löytyivät lähes kaikista ohjelmistoista, mutta niiden laatu ja lopputulos vaihtelivat erittäin turvallisesta minimiturvallisuuteen, suurin osa oli kuitenkin turvallisemmasta päästä. Kaupallisten tuotteiden suosimaa ratkaisua, eli pilvipalvelua tarjosi kolme seitsemästä ohjelmistosta, mutta vain yksi niistä oli avointa lähdekoodia. Mielestäni selkeä turvallisuuspuute kaikissa ohjelmistoissa oli se, että ne eivät painottaneet salasanan vahvuuden merkitystä etäyhteyden turvallisuudessa.

Jatkotutkimusmahdollisuuksia tutkimus toi esiin useita: Erityisesti kunkin ohjelmiston syvällisempi tutkimus, sisältäen ainakin päivitysten hakemisen, lisäosien ja modulaaristen palikoiden, etäkäyttöliittymien, sekä käyttäjänhallinnan turvallisuuden tutkimisen. Lisäksi langattomien protokollien (erityisesti muiden kuin WiFi) hyökkäysten tutkimus ja testaaminen, sekä sisäverkosta toteutettavat hyökkäykset, jotka liittyvät esimerkiksi käyttäjien tai laitteiden autentikoinnin puutteisiin ja heikkouksiin.

Lähteet

Bregman, David, ja Arik Korman. 2009. “A Universal Implementation Model for the Smart Home”. *International Journal of Smart Home.*

Bugeja, J., A. Jacobsson ja P. Davidsson. 2016. “On Privacy and Security Challenges in Smart Connected Homes”. Teoksessa *2016 European Intelligence and Security Informatics Conference (EISIC),* 172–175. Elokuu. doi:10.1109/EISIC.2016.044.

Copos, B., K. Levitt, M. Bishop ja J. Rowe. 2016. “Is Anybody Home? Inferring Activity From Smart Home Network Traffic”. Teoksessa *2016 IEEE Security and Privacy Workshops (SPW),* 245–251. Toukokuu. doi:10.1109/SPW.2016.48.

Farooq, M.u., Muhammad Waseem, Anjum Khairi ja Sadia Mazhar. 2015. “Article: A Critical Analysis on the Security Concerns of Internet of Things (IoT)”. Full text available, *International Journal of Computer Applications* 111, numero 7 (helmikuu): 1–6.

Geneiatakis, D., I. Kounelis, R. Neisse, I. Nai-Fovino, G. Steri ja G. Baldini. 2017. “Security and privacy issues for an IoT based smart home”. Teoksessa *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO),* 1292–1297. Toukokuu. doi:10.23919/MIPRO.2017.7973622.

Gyory, Nathaniel, ja M Chuah. 2017. “IoTOne: Integrated platform for heterogeneous IoT devices”. Teoksessa *Computing, Networking and Communications (ICNC), 2017 International Conference on,* 783–787. IEEE.

Hosek, Jiri, Pavel Masek, Dominik Kovac, Michal Ries ja Franz Kröpfl. 2014. “IP home gateway as universal multi-purpose enabler for smart home services”. *e & i Elektrotechnik und Informationstechnik* 131 (4-5): 123–128.

Islam, K., W. Shen ja X. Wang. 2012. “Security and privacy considerations for Wireless Sensor Networks in smart home environments”. Teoksessa *Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD),* 626–633. Toukokuu. doi:10.1109/CSCWD.2012.6221884.

Jose, A. C., R. Malekian ja N. Ye. 2016. “Improving Home Automation Security; Integrating Device Fingerprinting Into Smart Home”. *IEEE Access* 4:5776–5787. ISSN: 2169-3536. doi:10.1109/ACCESS.2016.2606478.

Kim, E., ja C. Keum. 2017. “Trustworthy gateway system providing IoT trust domain of smart home”. Teoksessa *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN),* 551–553. Heinäkuu. doi:10.1109/ICUFN.2017.7993848.

Li, Mingfu, ja Hung-Ju Lin. 2015. “Design and implementation of smart home control systems based on wireless sensor networks and power line communications”. *IEEE Transactions on Industrial Electronics* 62 (7): 4430–4442.

Lin, Huichen, ja Neil W Bergmann. 2016. “IoT privacy and security challenges for smart home environments”. *Information* 7 (3): 44.

Pacheco, J., ja S. Hariri. 2016. “IoT Security Framework for Smart Cyber Infrastructures”. Teoksessa *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W),* 242–247. Syyskuu. doi:10.1109/FAS-W.2016.58.

Pishva, D., ja K. Takeda. 2006. “A Product Based Security Model for Smart Home Appliances”. Teoksessa *Proceedings 40th Annual 2006 International Carnahan Conference on Security Technology,* 234–242. Lokakuu. doi:10.1109/CCST.2006.313456.

Plachkinova, Miloslava, Au Vo ja Ala Alluhaidan. 2016. “Emerging Trends in Smart Home Security, Privacy, and Digital Forensics”.

Ramljak, Milan. 2017. “Security analysis of Open Home Automation Bus system”. Teoksessa *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2017 40th International Convention on,* 1245–1250. IEEE.

Schiefer, M. 2015. “Smart Home Definition and Security Threats”. Teoksessa *2015 Ninth International Conference on IT Security Incident Management IT Forensics,* 114–118. Toukokuu. doi:10.1109/IMF.2015.17.

Singh, Saurabh, Pradip Kumar Sharma ja Jong Hyuk Park. 2017. “SH-SecNet: An Enhanced Secure Network Architecture for the Diagnosis of Security Threats in a Smart Home”. *Sustainability* 9 (4). ISSN: 2071-1050. doi:10.3390/su9040513. http://www.mdpi.com/2071-1050/9/4/513.

Son, Heesuk, Bjorn Tegelund, Taehun Kim, Dongman Lee, Soong J. Hyun, Junsung Lim ja Hyunseok Lee. 2015. “A Distributed Middleware for a Smart Home with Autonomous Appliances”. Heinäkuu.

Wilson, Charlie, Tom Hargreaves ja Richard Hauxwell-Baldwin. 2015. “Smart homes and their users: a systematic analysis and key challenges”. *Personal and Ubiquitous Computing* 19, numero 2 (helmikuu): 463–476. ISSN: 1617-4917. doi:10.1007/s00779-014-0813-0. https://doi.org/10.1007/s00779-014-0813-0.

Zheng, J. H., Y. Wang ja W. R. Tan. 2013. “An Adaptive Gateway for Smart Home”. Teoksessa *2013 International Conference on Computational and Information Sciences,* 1729–1732. Kesäkuu. doi:10.1109/ICCIS.2013.451.

Zhenhua, X. 2016. “Design and implementation of intelligent gateway for smart home”. Teoksessa *2016 Chinese Control and Decision Conference (CCDC),* 4713–4718. Toukokuu. doi:10.1109/CCDC.2016.7531836.

Zhou, C., W. Huang ja X. Zhao. 2013. “Study on architecture of smart home management system and key devices”. Teoksessa *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology,* 1255–1258. Lokakuu. doi:10.1109/ICCSNT.2013.6967330.