Eero Huusko

IoT-tietoturvaoppimisympäristön kehittäminen

Tietotekniikan
Pro gradu -tutkielma
27. huhtikuuta 2022

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Kokkolan yliopistokeskus Chydenius

Tekijä: Eero Huusko
Yhteystiedot: eero.a.huusko@gmail.com
Puhelinnumero: 0442608641
Ohjaaja: Risto T. Honkanen, Ismo Hakala, Mikko Myllymäki
Työn nimi: IoT-tietoturvaoppimisympäristön kehittäminen
in English: Development of IoT security learning environment
Työ: Tietotekniikan Pro gradu -tutkielma
Sivumäärä: 85+22
Tiivistelmä: Esineiden internet (IoT - Internet of Things) tarkoittaa "esineiden" yhdistämistä internetiin. Esineet voivat olla erilaisia laitteita kuten älykello, auto, robotti-imuri, IP-kamera, älykoti IoT-laitteet tai vaikkapa kahvinkeitin. Langattomat sensoriverkot (WSN) koostuvat erilaisista langattomista mittausyksiköistä, jotka mittaavat ja keräävät tietoa ympäristöstään. Vaikka IoT tarjoaa tehokkaita ja joustavia ratkaisuja moniin tosielämän haasteisiin, on tietoturvaan liittyviä ratkaisuja syytä kyseenalaistaa. Tilannetta pahentaa entisestään, kun yhdistettyjen laitteiden määrä kasvaa eksponentiaalisesti. Tämän seurauksena tietoturvasta ja yksityisyydestä on tullut merkittävä haaste IoT-järjestelmille. Tässä Pro gradu -tutkielmassa tutkittiin millainen on IoT-tietoturvaosaamisen tarvekartoituksen ja pedagogisen mallin mukaisesti toimiva ja ratkaisukeskeiseen oppimiseen parhaiten soveltuva oppimisympäristö. Painotus tutkimuksessa kohdistui IoT-järjestelmien ja erityisesti langattomien sensoriverkkojen tietoturva- ja yksityisyyshaasteisiin käytetyn teknologian ja arkkitehtuurin näkökulmaan. Tässä Pro gradu -tutkielmassa keskityttiin oppimisympäristön suunnitteluun ja toteuttamiseen kehittämistutkimuksen menetelmin. Kehitetyssä oppimisympäristössä on mahdollista toteuttaa opetusta langattomien sensoriverkkojen ja IoT-järjestelmien kohdistuviin haavoittuvuuksiin ja haavoittuvuuksien vaikutuksiin luottamuksellisuuteen, eheyteen ja saatavuuteen. Kehitystyön tuloksena toteutettiin useita IoT-laitteita ja Internet-verkon toimilaitteita sisältävä oppimisympäristö, joka voidaan tarvittaessa irrottaa julkisesta Internetistä. Oppimisympäristön kehityssykleissä suunniteltiin ja pilotoitiin useita erilaisia oppimistehtäviä. Kehityssykleissä analysoitiin myös oppimistehtävien soveltuvuutta IoT-järjestelmän tieto- ja kyberturvallisuuden testausmenetelmien opettamiseen. Parhaaksi vaihtoehdoksi todettiin IoT-järjestelmän tietoturvan opettaminen siten, että opetus toteutetaan noudattamalla penetraatiotestauksen menetelmiä. Didaktinen malli noudattaa Kajaanin ammattikorkeakoulussa käytössä olevaa sosiokonstruktivististä oppimiskäsitystä, jossa korostetaan tekemällä oppimista teoreettisen

tiedon pohjalta. Työn tuloksena saatiin myös ymmärrystä siitä, mitä osaamista opiskelijoilla tulee olla ennen IoT-tietoturvakurssin suorittamista. Merkittävin osaamisvaatimus liittyi tietoverkkoihin ja tietoliikenneteknologiaan osaamiseen.
Avainsanat: IoT, Esineiden Internet, Langattomat sensoriverkot, Tietoturva, Oppimisympäristö, IoT-järjestelmä, Penetraatiotestaus, Tietoliikennetekniikka, Internet verkot, Tietoverkot, Sosiokonstruktivismi
Abstract: Internet of Things (IoT) means the connection of "objects" to the Internet. The objects can be various devices such as a smart watch, a car and a robot vacuum cleaner, especially an IP camera, smart home devices or even a coffee maker. Wireless sensor networks (WSNs) consist of a variety of wireless measurement units that measure and collect information about their environment. While IoT provides effective and flexible solutions to many real-life challenges, security-related solutions are questionable. The situation is exacerbated as the number of connected devices increases exponentially. As a result, security and privacy have become a major challenge for IoT systems. This Master's thesis examined the learning environment that works in accordance with the needs assessment and pedagogical model of IoT security expertise and is best suited for solution-centered learning. The emphasis in the study was on the technology and architecture used to address the security and privacy challenges of IoT systems, and in particular wireless sensor networks. This Master's thesis focused on the design and implementation of a learning environment using development research methods. In a developed learning environment, it is possible to teach about the vulnerabilities of wireless sensor networks and IoT systems and the impact of the vulnerabilities on confidentiality, integrity and availability. As a result of the development work, a learning environment with several IoT devices and Internet network actuators was implemented, which can be disconnected from the public Internet if necessary. Several different learning tasks were planned and piloted in the development cycles of the learning environment. The development cycles also analyzed the suitability of learning tasks for teaching IoT network, datasecurity and cybersecurity testing methods. The best option was to teach the security of the IoT system by following the penetration testing methods. The didactic model follows the socio-constructivist concept of learning used at Kajaani University of Applied Sciences, which emphasizes learning by doing it on the basis of theoretical knowledge. The work also resulted in an understanding of what skills students should have before completing an IoT security course. The most significant competence requirement was related to competence in wireless networks

and telecommunications technology.

Keywords: IoT, Internet of Things, WSN, Wireless Sensor Network, Data Security, Learning environment, IoT-ecosystem, Penetration testing, Internet networks

Esipuhe

Tämän pro gradun kirjoittaminen on ollut lähes puolen vuoden projekti. Aihe on pysynyt sentään samana, mutta sisältö on elänyt sekä oman oppimisen myötä, mutta myös asiantuntevien ohjaajien myötävaikutuksella.

Sanasto

6LoWPAN	IPv6 over Low Power Wireless Personal Area Networks
Anomalia	Poikkeavuus
ARP	Address Resolution Protocol
IoT	Internet of Things (Esineiden Internet)
CoAP	Constrained Application Protocol
DDoS	Distributed Denial of Service
DDS	Data Distribution Servic
DTLS	Datagram Transport Layer Security
DREAD	Tietoturvariskien luokittelumenetelmä
EAL	Evaluation Assurance Level
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HW	Hardware
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
I/O	Input/Output
IoT	Internet of Things
MAC	Media Access Protocol
M2M	Machine-to-Machine
MD	Message Digest
MITM	Man-in-the-Middle

MQTT	Message Queue Telemetry Transport
OS	Operating System
OSI	Open Systems Interconnection Reference Model
POC	Proof of concept, soveltuvuusselvitys, jolla osoitetaan idea toteuttamiskelpoiseksi
RAM	Random Access Memory, Muistipiiri
RTSP	Real Time Streaming Protocol
SHA	Secure Hash Algorithm
SSH	Secure Socket Shell
STRIDE	Tietoturvariskien analysointimenetelmä
SoC	System-on-Chip, Järjestelmäpiiri
TRNG	True Random Number Generator
TBC	To Be Confirmed
TBD	To Be Determined
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TKI	Tuotekehitys ja innovaatio
T3P	Trusted Third Party
UDP	User Datagram Protocol
URL	Uniform Resource Locator
Wireshark	Verkkoprotokolla-analysaattori
WLAN	Wireless Local Area Network
WPS	Wi-Fi Protected Setup
WSN	Wireless Sensor Network

Sisällys

1 Johdanto

Opinnäytetyön aiheen taustalla on Kajaanin ammattikorkeakoulun Tieto- ja viestintätekniikan insinöörien opetussuunnitelmapäivitys. Opetussuunnitelmaan on päätetty lisätä tieto- ja kyberturvallisuuden opetusta. Kainuun ja lähialueiden yrityksiltä on myös tullut useita yhteydenottoja, joissa tietoturvan opetusta on toivottu opetukseen. Tieto- ja kyberturvallisuuden osaamistarve korostuu, sillä tietoturvariskit ja tietoturvahyökkäykset ovat yleistymässä kiihtyvään tahtiin. Aihe on myös ajankohtainen siksi, että IoT tulee muuttamaan asumista, liikkumista, vapaa-aikaa, kaupunkiympäristöä ja ihmisten välistä kommunikointia. Valmistuvien insinöörien on erityisen tärkeää ymmärtää se, että yksityisyys ja yksilön turvallisuus ovat vaarassa, kun otetaan huomioon, kuinka henkilökohtaisia ovat esimerkiksi kodin IoT-laitteet ja kuinka henkilökohtaista dataa laitteet lopulta keräävät.

IoT-laitteiden kuluttaja-asiakkailla on alati kasvava tarve hankkia esimerkiksi kodinkoneita, viihdelaitteita ja kodin turvajärjestelmiä, joilla on pääsy internettiin. Näiden IoT-laitteiden ydintarkoituksena on parantaa ihmisten elämää ja hyvinvointia joko auttamalla ihmisiä tekemään parempia päätöksiä tai auttaa heitä hallitsemaan elin- tai asuinpiiriään. Kodin IoT-laitteilla pyritään myös helpottamaan arkea, mutta kuten kirjoittajat Bastos et al. [9] artikkelissaan toteavat, että tavoitteena on myös saada myönteisiä vaikutuksia yhteiskuntaan, talouteen ja ympäristöön. Kirjoittajat toteavat lisäksi, että IoT avaa monia mahdollisuuksia automatisoinnin ja kontekstuaalisen tiedon suhteen [9]. He toteavat, että nykyisin asuinrakennuksissa voidaan seurata lämpötilaa, kosteutta, valoja, kaihtimia, sähkön käyttöä, jääkaapissa olevan ruoan määrää, tunkeilijoita ja niin edelleen. Langattomat IoT-laitteet voivat auttaa seuraamaan liikkeessä olevien asiakkaiden määrää, kerätä tietoa mihin tuotteisiin asiakkaat keskittyvät ja kuinka kauan aikaa asiakkaat ovat kaupassa.

IoT-teknologiaa otetaan nykyään käyttöön monilla toimialoilla. Esimerkiksi Euroopassa Alliance for Internet of Things Innovation (AIOTI) [4] on suunnitellut joukon pilottiprojekteja, jotka keskittyvät esittelemään IoT:n todellisia käyttötapauksia. Muutama toiminnassa oleva esimerkkipilotti on kuvattu alla olevassa luettelossa. Pilottiprojektit osoittavat IoT:n laajat käyttö- ja sovellusmahdollisuudet. IoT on paljon muutakin kuin Internetiin yhdistettyjä kuluttajaleluja. IoT-järjestelmät etene-

vät kohti todellista muutosta, joilla pystytään parantamaan väestön hyvinvointia ja kasvattamaan tuottavuutta yritysympäristössä. Näitä ovat muun muassa seuraavat:

- Vanhusten älykäs elinympäristö, jossa IoT-järjestelmät tukevat elämänlaadun parantamista ja vähentävät ikääntyvän väestön hoitokustannuksia,
- Älykäs maatalous ja elintarvikehuolto, jossa IoT-järjestelmät mahdollistavat tarkemman viljely ja mahdollistavat samalla uusien menetelmien kehittämisen elintarviketurvallisuuden varmistamiseksi. Uudet autonomiset tekniikat vähentävät työmäärää ja vähentävät maatalouden ilmastokuormitusta,
- Puettavat vaatteet projektissa IoT-järjestelmät integroituvat jokapäiväiseen elämään puettavien laitteiden, älyvaatteiden, älykellojen ja vartaloon kiinnitettävien laitteiden kautta,
- Älykäs energia pilotissa IoT-järjestelmät tukevat energiankäytön optimointia mukaan lukien uusiutuvan energian käyttö, sähkönsiirtoverkkojen valvonta, voimalaitokset, kysyntään reagoivat sovellukset ja sähköajoneuvojen lataukset ja
- Älykkäät rakennukset ja arkkitehtuurit pilottiprojektissa IoT-järjestelmät muuttavat kiinteistöjen hallintaa keskittyen asukkaiden elämänlaatuun parantamalla valaistusta, mukavuutta, lämpötilaa, ilmanlaatua, vettä, ravintoa, fyysistä kuntoa ja energian käyttöä.

Pro gradu -tutkielman tutkimusmenetelmänä on kehittämistutkimus, jonka kahdessa syklissä kehitettiin IoT-tietoturvan opetukseen soveltuva, skaalautuva fyysinen oppimisymäristö. Ensimmäisen tutkimuskysymyksen selvitettiin ensinnäkin se, millaista osaamista valmistuvilta opiskelijoilta odotetaan ja millainen tarve IoT-tietoturvaosaamiselle on tutkimushetkellä. Toisessa tutkimuskysymyksessä selvitettiin sitä, millaisilla oppimistehtävillä voidaan saavuttaa tavoitteiden mukaiset oppimistulokset. Kolmannessa tutkimuskysymyksessä haettiin vastausta siihen, mitä laitteisto- ja järjestelmävaatimuksia oppimisympäristön suunnittelussa tulee ottaa huomioon. Teoreettisessa ja empiirisessä osuudessa tutkittiin oppimistehtävien lisäksi laite- ja järjestelmävaatimuksia, joita oppimisoppimisympäristön suunnittelussa tulee ottaa huomioon. Kehittämistutkimuksen ensimmäisessä syklissä toteutettiin teoreettisen ja osin empiirisen tutkimuksen tuloksena oppimisympäristö, jota pilotoitiin muutaman opiskelijan työharjoittelussa. Havaittujen puutteiden pohjalta

kehitettiin toisessa kehittämissyksissä oppimisympäristö, jossa toteutetiin kahden eri kurssin IoT-turvakurssin opetus. Oppimistehtävien tavoitteeksi asetettu käytännönläheisen opetuksen toteuttaminen saavutettiin. Oppimistehtävien kehittämistä, testaamista ja pilotointia tehtiin samanaikaisesti oppimisympäristön kehittämisen kanssa. Toisen kehityssyklin lopputuloksena olevassa oppimisympäristössä voidaan toteuttaa oppimistehtäviä ja oppia perusteet tieto- ja kyberturvallisuudesta sekä syventävää osaamista IoT-järjestelmien tietoturvan analysoinnissa. Oppimistehtävissä toteutuvat myös asetetut pedagogiset tavoitteet. Tutkimuskysymyksillä haettiin myös vastausta siihen, millaiset oppimistehtävät ja millainen oppimisympäristö mahdollistaa tekemällä oppimisen sekä käänteisen oppimisen.

Luvussa 2 esitellään oppimisympäristön asetettavat tavoitteet pedagogisesta näkökulmasta katsottuna. Pedagogisen mallin mukaisesti osa tehtävistä harjoituksista toteutetaan opettajan ohjauksessa oppilaitoksen oppimisympäristössä ja osa itsenäisesti opiskelijan omalla tietokoneella. Oppilaitokseen rakennettavan oppimisympäristön tulee olla mahdollisimimman helposti käyttöön otettava, jotta käytännön harjoitukset voidaan aloittaa mahdollisimman vaivattomasti. Tavoitteeksi asetettiin myös se, että opiskelijoilla on mahdollisuus suorittaa osa harjoitustehtävistä itsenäisesti ennakkoon virtuaalisessa oppimisympäristössä. Tämä edesauttaa laajempien oppimistehtävien suorittamisen oppilaitosympäristössä. Tämä oppimismenetelmä on ns. Flipped learning -malli. Flipped learning tarkoittaa käänteistä opetusta, jolloin oppimisympäristössä toteutettavilla oppitunnilla jää enemmän aikaa syventävien harjoitustehtävien tekemiseen [35].

Luvussa 3 esitellään tyypillisimmät IoT-järjestelmien arkkitehtuurimallit. Opinnäytetyön tavoitteena on suunnitella ja luoda oppimisympäristö, joka mallintaa mahdollisimman realistisesti tyypillisiä sekä langallisia tietoliikenneratkaisuja että langattomien sensoriverkkojen toteutuksissa käytettyjä tietoliikenne- ja tietoverkkoratkaisuja. Oppimisympäristön tulee olla skaalautuva, jolloin on mahdollista liittää uusia verkon toimilaitteita tai langattomia sensorinoodeja oppimisympäristöön.

Luvussa 4 keskitytään erityisesti sensoriverkkojen (*engl. wireless sensor network, WSN*) ja esineiden internetin eli IoT-verkkojen (*engl. Internet of Things network*) tietoturvaan IoT-järjestelmien erityispiirteet huomioiden. Langattomiin sensoriverkkoihin ja IoT-verkkoihin koskee samat turvallisuusmääritykset kuin muihinkin verkkoihin, mutta langattomien sensoriverkkojen toteutus ei ole kuitenkaan yhtä helppoa, koska WSN-verkot ja niiden laitteistot eivät rajoitettujen resurssien tähden pysty tukemaan monimutkaisia suojamekanismeja. Tämä tuokin suuria haasteita vah-

vojen turvallisuusprotokollien suunnittelulle ja tietoturvan analysoinnille [68].

Luvussa 5 kerrotaan toteutetusta kehittämistutkimuksesta tarkemmin. Luvussa esitellään millainen kehittämistutkimus on tutkimusmenetelmänä, esitellään sekä teoreettinen että empiirinen ongelma-analyysi. Luvussa esitellään myös tutkimuskysymykset, joihin haettiin vastauksia kehittämistutkimuksen käytännön toteuksessa. Käytännön toteutus noudatteli kehittämistutkimukselle tyypillistä kahden kehittämisysklin menetelmää. Teoreettisen ja empiirisen ongelma-analyysien perusteella muodostettiin vaatimukset, jotka IoT-tietoturvaoppimisympäristön tulee täyttää. Teoreettisen osuuden toteutuksessa merkittäväksi tekijäksi nousi haastattelututkimus, jossa kartoitettiin alueen yritysten näkemys IoT-tietoturvan osaamisvaatimuksiksi.

Luvussa 6 esitellään ensimmäisen kehittämisyklin kehittämistuotos. Luvussa kerrotaan myös kehittämistuotoksen pilotoinnin tuloksista ja kehityskohteista. Koska tavoitteeksi asetettiin se, että oppimisympäristössä pystytään opettamaan sekä Tietoverkkojen että IoT-tietoturvakursseihin liittyviä syventäviä harjoituksia, todettiin, että oppimisympäristössä sekä oppimistehtävissä on merkittäviä puutteita. Havaittiin, että oppimisen kannalta on tärkeää, että oppimistehtävissä on oltava perinteisen tietoverkkojen tietoturvan ja IoT-laitteita koskevan tietoturvan lisäksi myös fyysisen tietoturvan elementtejä.

Luvussa 7 käsitellään oppimisympäristön testausta kahdella eri IoT-tietoturvakurssilla. Molempiin kursseihin sisältyi teorialuentojen lisäksi käytännön harjoituksia ja oppimistehtäviä. Toisessa kehittämissyklissä oppimisympäristöön lisättiin paljon erilaisia IoT-laitteita, joista osa on kuluttajamyynnissä olevia laitteita. Toisen kehittämistuotoksen testausvaiheessa käytiin myös läpi hakkeroinnin etiikkaa eettisen hakkeroinnin periaatteiden mukaisesti. Lähtökohtana toisen kehittämissyklin oppimistehtävissä oli opettaa opettajajohtoisen teorian, tiimityön ja vertaisoppimisen menetelmin erityyppisiä penetraatiotestausmentelmiä. Oppimistehtävissä suoritettiin erilaisia testejä haavoittuvuuksien löytämiseksi oppimisympäristön IoT-järjestelmässä. Luvussa analysoidaan myös oppimistehtävillä saavutettuja oppimistavoitteita suhteessa oppimistavoitteisiin.

Oppimisympäristön jatkokehittäminen yhdessä opetussuunnitelman ja IoT-tietoturvakurssin sisällön kanssa on ensiarvoisen tärkeää. Luvussa 8 kerrotaan laajasti erilaisia jatkokehitystarpeita, jotka nousivat korostetusti esille myös oppimisympäristön esittelytilaisuudessa. Yhteistyöyritysten edustajat kokivat tärkeänä jo saavutettuja oppimistuloksia, mutta korostivat IoT-tuotekehitysprosessin ymmärtämisen

tärkeyttä. Myös tietoverkko-osaaminen nousi korostetusti esille tärkeänä osaamisvaatimuksena.

2 IoT-tietoturvakurssin oppimistavoitteet ja pedagoginen malli

Tässä luvussa kuvataan kehittämistutkimuksen kohteena olevan IoT-tietoturvakurssin oppimistavoitteet ja opetussuunnitelmassa suunniteltu sisältö. Luvussa kuvataan myös kurssin opetuksessa käytettävä pedagoginen malli. Pedagoginen malli on yleensä kuvattu jokaisen oppilaitoksen ja korkeakoulun johtosäännössä tai laatukäsikirjassa. Pedagogiset mallit ovat ovat teoreettisiä malleja siitä, kuinka opetus- ja oppimistilanteen etenevät ja millainen rooli opettajalla ja oppijalla on kussakin mallissa. Pedagogisissa malleissa keskeisintä on oppimisprosessin kuvaus ja kuinka eri oppimisprosessin vaiheet toimivat opetuksen suunnittelussa. Eri pedagogisissa malleissa korostetaan opiskelijan aktiivisuutta, oppimistilanteen vuorovaikutusta ja yhteistoimintaa. Pedagogisten mallien valintaa ohjaa kuitenkin aina ensisijaisesti opintojakson osaamis- tai oppimistavoitteet, vaikkakin oppilaitoksessa olisi käytössä joku tietty pedagoginen malli [25]. Käytännössä opettajan pedagogisen osaamisen näkemyksen pohjalta valitaan pedagoginen malli tai pedagogiset mallit, joita käytetään opintojakson eri vaiheissa. Hyvin usein voidaan käyttää erilaisia opetusmenetelmiä opintojakson eri toteutusvaiheissa, jotta halutut oppimistavoitteet voidaan saavuttaa [25]. Aliluvussa 2.1 esitellään kehitystyön kohteena olevan IoT-tietoturvakurssin kurssikuvaus ja oppimistavoitteet. Aliluvussa 2.2 kuvataan Kajaanin ammattikorkeakoulussa sovellettava pedagoginen malli.

2.1 IoT-tietoturvakurssin kuvaus ja oppimistavoitteet

Kurssilla opiskelijat saavat IoT (Internet of Things) -ratkaisujen tietoturvaa ja tietosuojaa koskevan perusosaamisen. Kurssilla käsitellään IoT-verkkojen aiheuttamia organisatorisia riskejä ja haavoittuvuuksien periaatteita. Kurssilla perehdytään myös teollisuuden IoT-järjestelmien tietoturvaan. Opiskelijat saavat myös ymmärryksen yleisistä tietoturva-arkkitehtuurimalleista, joita voidaan soveltaa IoT-järjestelmiin, samoin opiskelijat perehtyvät IoT-verkkojen tietoturvaan liittyviin sääntöihin, standardeihin ja tietoliikenneprotokolliin. Opiskelijoiden tavoitteena on myös oppia tuottamaan tietoturvallista ohjelmakoodia. Kurssin sisältö on kuvattu opetussuunnitel-

massa seuraavasti [31]:

1. IoT-arkkitehtuurimalien ja IoT-verkon toimilaitteiden tietoturva,
2. IoT-protokolliin liittyvä toiminnallisuus ja haavoittuvuudet,
3. IoT-verkkojen haavoittuvuuksien analysointi ja
4. IoT-tietoturvastandardien soveltaminen IoT-järjestelmiin, esim. puettava teknologia, terveydenhuollon IoT-järjestelmät, kodin IoT-laitteiden tietoturva, selainpohjaisten sovellusten tietoturva ja mobiililaitteiden tietoturva.

Kurssin oppimistavoitteet on kuvattu opetussuunnitelmassa seuraavasti [31]:

- Opiskelija ymmärtää perustiedot tietoverkkojen uhkista ja suojautumismekanismeistä,
- Opiskelija saa valmiudet tietoturvatavoitteiden määrittelyyn, ratkaisujen suunnitteluun sekä suojausmenetelmien käyttöönottoon,
- Opiskelija tuntee yleisimmät tietoturva- ja kyberuhat, hyökkäykset ja haavoittuvuudet,
- Opiskelija ymmärtää hakkerointiin liittyvät eettiset periaatteet ja
- Opiskelija osaa soveltaa kurssilla oppimiaan perusteita tunnistaakseen tietoja kyberturvariskejä sekä ehkäistäkseen niitä omassa työssään.

Kurssin laajuus on viisi (5) opintopistettä. Opiskelijan laskennallinen työmäärä on siten 135 tuntia, josta lähiopetusta on 65 tuntia. Kurssin laajuus huomioiden oppimistehtävät on päätetty kohdentaa langattomien sensorisolmujen tietoturvaan, IoT-ekosysteemissä käytettyjen tietoliikenneteknologioiden tietoturvaan ja tiedon tallentamiseen pilvipalveluun. Opetussuunnitelman sisällön läpileikkaavana ajatuksena on ns. "Anturista pilveen-teema. Oppimistehtävät eivät sisällä Web-selaimiin liittyviä tietoturvavaatimuksia.

2.2 Opetuksen didaktinen viitekehys ja pedagoginen malli

Kajaanin ammattikorkeakoulun opetussuunnitelmien lähtökohtana on sosiokonstruktivistinen oppimiskäsitys. Sosiokonstruktivismin mukaan tieto rakentuu ja jäsentyy

osaamiseksi, kun opiskelijat osallistuvat yhdessä ongelmien ratkaisuun ja tehtävien tekemiseen. Opiskelijan oma aktiivinen toiminta toimii näin ollen pohjana oppimiselle. Opiskelija tulee myös pystyä ottamaan vastuuta oppimisestaan ja pyrkiä aktiivisesti saavuttamaan tavoitteena oleva osaamisen taso [35].

Suomalaisen yliopistokoulutuksen arvioinnin ja kehittämisen keskeisiä näkökulmia ovat koulutuksen opiskelijakeskeisyys sekä osaamis- ja työelämälähtöisyys [33]. Korkeakoulutuksen ja työelämän työtehtävien vastaavuutta on kaikessa korkeakouluopetuksessa pyritty lisäämään esimerkiksi hyödyntämällä pedagogisia malleja ja opetuskäytänteitä, joissa opetussuunnitelman ja opiskelun lähtökohtana on todenperäinen työ- ja yhteiskunnallinen elämä, ilmiöt, tilanteet ja ongelmat. Esimerkkejä malleista ovat ongelmaperustainen oppiminen (*engl.PBL, Problem-Based Learning*) [51] ja tapausperustaiset opetusmenetelmät [17]).

Tavoitteena Kajaanin ammattikorkeakoulun Tieto- ja viestintätekniikan insinööriopintojen opetussuunnitelmissa on edetä ns. top-down -menetelmällä, jossa opiskelija oppii sekä teoreettisesti oman alansa perustietoa että perehtyy myös itsenäisesti tutkittuun tietoon tai käsiteltävään ilmiöön. Oppimisen tulee myös syventyä opintojen edetessä. Kokemusperusteisen oppimisen, teoreettisen tiedon lisääntyessä ja erityisesti oppimisvalmiuksien parantuessa opiskelija pystyy ymmärtämään eri tyyppisiä ilmiöitä ja löytää ratkaisuvaihtoehtoja esillä olevaan ongelmaan.

Opettajalta sosiokonstruktivistisen oppimiskäsityksen mukainen opetus edellyttää innovatiivista otetta. On myös pystyttävä luomaan sellainen oppimisympäristö tai erilaisten oppimisympäristöjen kokonaisuus, jossa opiskelijat pystyvät toimimaan yhteistyössä, etsimään tiiminä ratkaisuja ja ennen kaikkea soveltamaan aikaisemmin opittuja tietoja ja taitoja.

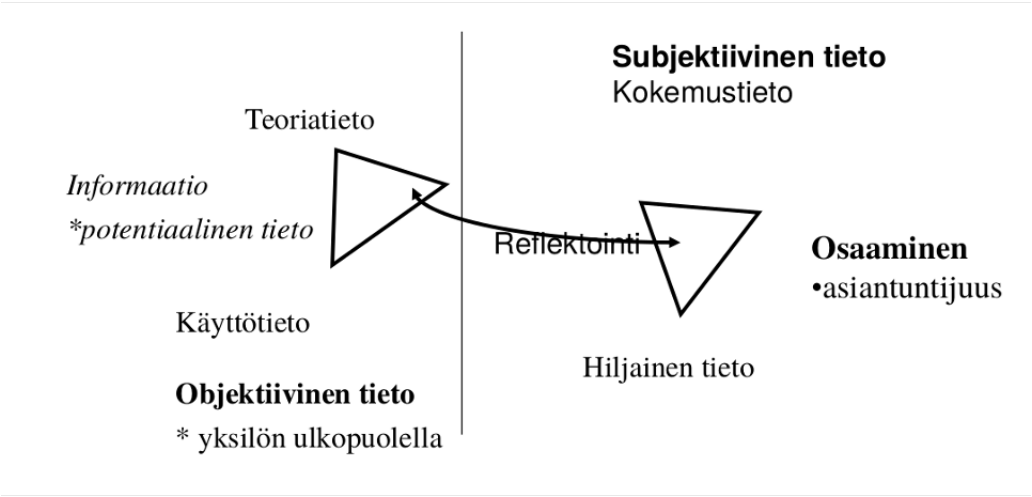
Lähtökohtana oppimistilanteessa on toteuttaa opetus mahdollisimman todentuntuisessa ympäristössä. Tämä tarkoittaa sitä, että käytettävät ohjelmointikielet, prosessityömenetelmät, laitteet, oppimisympäristöt ja tietoverkot on rakennettu vastaamaan työelämässä keskimääräisesti käytössä olevia. Tutkimusten mukaan tosielämää varten tapahtuva oppiminen sekä oppiminen aidoissa oppimisympäristöissä nostavat laaja-alaisen osaamisen oppimisen keskiöön [32].

Kajaanin ammattikorkeakoulussa opetus, TKI -toiminta ja tiivis työelämäyhteistyö muodostavat vuorovaikutteisen kokonaisuuden, jonka avulla pyritään vastaamaan ammattikorkeakouluopetukselle kohdistuviin uusiin odotuksiin. Opetuksessa pyritään luomaan ammatillinen ote opiskeluun ja yritysyhteistyöhön. Ammatillisuus edellyttää myös ns. hiljaisen tiedon muokkaamista vuorovaikutuksessa opis-

kelijan, opettajan ja työelämän välillä.

Kajaanin ammattikorkeakoulussa on panostettu monipuolisiin oppimisympäristöihin, joiden kautta oppijat kohtaavat uusia tilanteita, joissa on mahdollisuus myös uusille oivalluksille. Oppimis- ja opetusmenetelmiä käytetään luovasti työelämälähtöistä näkökulmaa painottaen. Opettajan tulee yhä vahvemmin olla oppimisen asiantuntija ja oppimisen ohjaaja, ei niinkään opetettavan tietosisällön suvereeni haltija ja välittäjä. Käytännössä opettaminen on nykyisin yhä enemmän tiimityötä. Opettajat tiimityön esimerkkinä tuovat parhaiten esiin myös yhdessä tekemisen tärkeyden ja välttämättömyyden. Tämä osaltaan vahvistaa niitä käytännön tiimityötaitoja, joita opiskelija tarvitsee menestyäkseen työelämässä.

Kajaanin ammattikorkeakoululla on opiskelijoiden ylläpitämä palvelinsali ja supertietokone. Opetusta pyritään toteuttamaan mahdollisimman laaja-alaisesti siten että työelämässä tarvittavaa substanssiosaamista tulisi myös muualta kuin opetussuunnitelman mukaisilta aiheilta. Kuvassa 2.1 on esitetty kokemusperäisen oppimisen malli. Kuva perustuu kirjassa ”Työ, identiteetti ja oppiminen” esitettyyn työssäoppimisen malliin. [50] Kuvasta nähdään kuinka saatu teoriatieto ja oppijan olemassa oleva käyttötieto jäsentyvät osaamiseksi. Todellinen hyöty kokemusperäisessä oppimisessa oppijalle on subjektiivisen tiedon jäsentyminen kokemustiedoksi.



Kuva 2.1: Informaation, tiedon ja osaamisen syventyminen työssäoppimisen prosessimallin mukaisesti.[50]

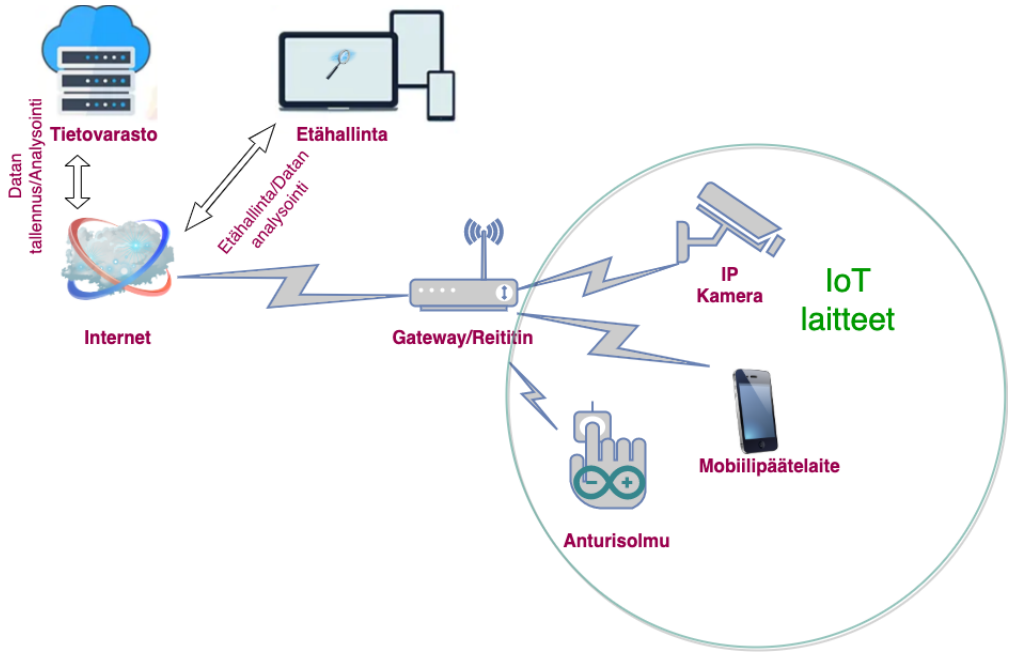
Kajaanin ammattikorkeakoulussa käytetyn didaktisen mallin mukaisessa oppimaan ohjaamisessa teknologia, oppimisympäristöt ja opeteltavat asiat, ongelmat ja

ilmiöt, pyritään liittämään samaan kokonaisuuteen. Mallin mukaisessa opetuksessa ei ratkaista ongelmaa, vaan opitaan se, millaista tietoteknistä osaamista ongelman ratkaisu vaatii ja opitaan myös se mitä kaikkea yhteistyötä esimerkiksi tietoturvallisten IoT-järjestelmien kehittäminen edellyttää.

3 IoT-järjestelmä

Internet of Things (IoT) voidaan määritellä eri tavoin riippuen käyttäjien erilaisista painotuksista. Suomen kielessä IoT:stä puhuttaessa tarkoitetaan yleisesti asioiden ja esineiden Internettiä tai pelkästään esineiden Internettiä. Joskus myös IoT-termiä käytetään, kun tarkoitetaan teollista Internettiä, joskin viime aikoina on yleistynyt termi Industrial IoT (IIoT).

Artikkelissaan Clark [29] kuvaa IoT:n, eli Esineiden Internetin järjestelmäksi, jossa mikä tahansa laite yhdistyy tietoliikenneverkkoihin ja niiden kautta toisiin laitteisiin. IoT on kuvauksen mukaisesti jättimäinen verkosto laitteita ja ihmisiä, jotka kaikki keräävät ja jakavat dataa toiminnastaan sekä ympäröivästä ympäristöstä. Kuvassa 3.1 on kuvattu esimerkinomaisesti IoT-järjestelmä. Iot-laitteet lähettävät tietoa omasta ympäristöstään käyttäjilleen Internet-verkkoa käyttäen. Eri tiedonsiirtoprokollilla toteutetut IoT-laitteiden verkot yhdistetään reitittimen tai välityspalvelimen (Gateway) kautta normaaliin TCP/IP protokollaa käyttävään Internet-verkkoon.



Kuva 3.1: Esimerkkikuva IoT-järjestelmästä

Aliluvussa 3.1 selostetaan IoT-järjestelmän arkkitehtuurimallia ja kerrosrakennetta. IoT-laitteita käsitellään aliluvussa 3.2 IoT-oppimisympäristön kehitystyötä ajatellen. Data- ja tietoliikenneprotokollia käsitellään aliluvussa 3.3. Pääpaino on suunnittelun kohteena olevan oppimisympäristön data- ja tietoliikenneprotokollissa.

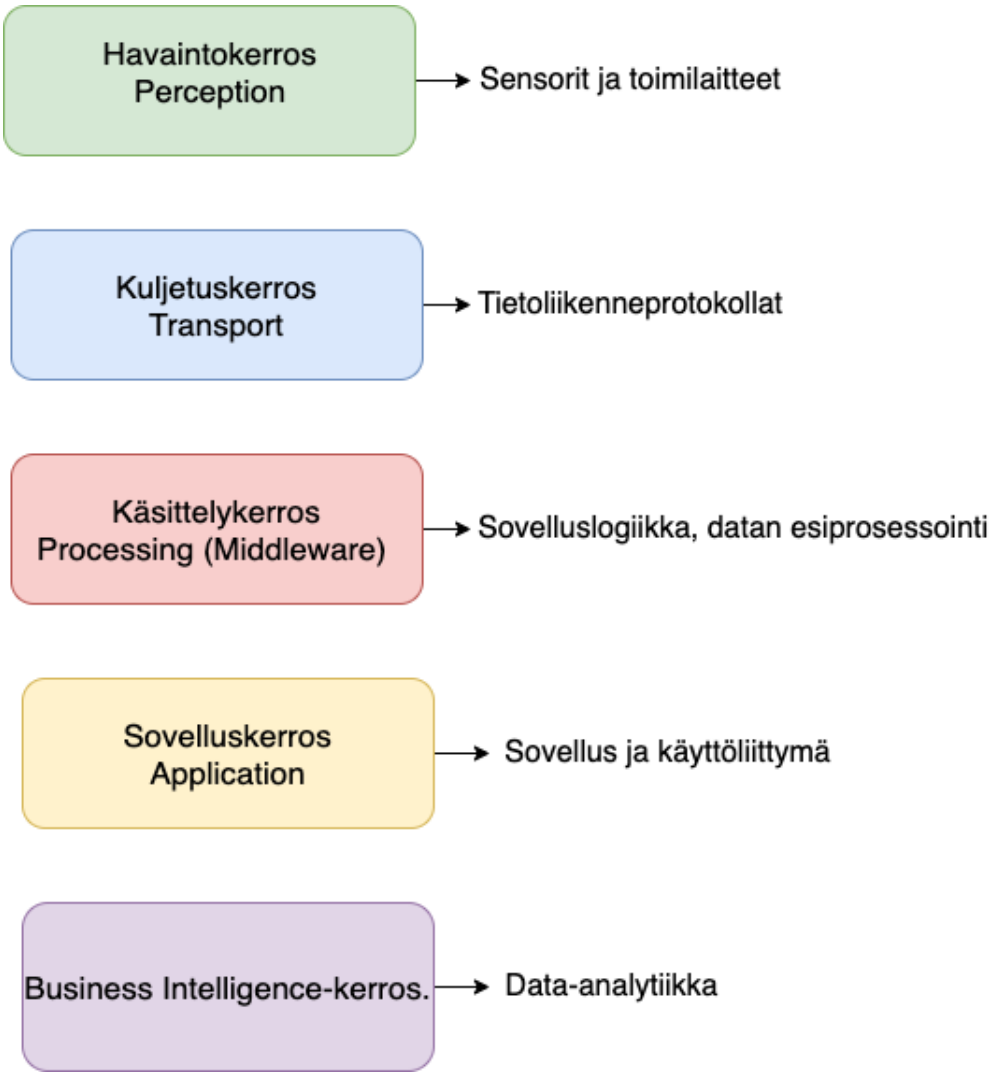
3.1 IoT-järjestelmän arkkitehtuuri

IoT-laitteet sisältävät yleensä antureita ja toimilaitteita. Nämä anturit keräävät tietoja ympäristöstään. IoT-järjestelmään liittyy olennaisena osana tiedon siirtäminen anturisolmusta pilvipalveluun. Usein IoT-järjestelmään liittyy myös data-analytiikka, jolloin IoT-laitteet toimivat integroidulla mallilla, joka sisältää tekoälyn ja koneoppimisen. Antureilla kerätään analogista tai digitaalista dataa fyysisestä maailmasta. Pääkäsittely-yksikkö, joka on yleensä mikroprosessori tai mikro-ohjain, suorittaa mittaustiedon käsittelyn. Viestintäjärjestelmä koostuu radiojärjestelmästä, yleensä lyhyen kantaman radiosta, jolla IoT-laite lähettää ja vastaanottaa tietoja. IoT-laite koostuu siten anturikomponentista, prosessointi-, viestintä- ja tallennusyksiköistä. IoT-arkkitehtuuri kuvataan useimmiten viisikerroksisella arkkitehtuurimallilla, joka sisältää kuljetuskerroksen, tiedon käsittely- ja Business Intelligence-kerroksen, havainto- ja sovelluskerrosten lisäksi. Tyypillisen IoT-ekosysteemin arkkitehtuurikuvaus on esitetty kuvassa 3.2.

Havaintokerros: Sensorit ovat tällä kerroksella ja tiedot voidaan kerätä mistä tahansa laitteeseen yhdistetyistä sensoreista. Toimilaitteet, jotka vaikuttavat ympäristöönsä, ovat myös arkkitehtuurin tässä kerroksessa. Tällä kerroksella ovat esimerkiksi langattomat sensoriyksiköt (*engl. Wireless Sensor Node*), kodin IoT-laitteet jne.

Verkkokerros:Verkkokerros yhdistää eri laitteet ja lähettää tiedot asianmukaisiin taustapalveluihin.

Sovelluskerros: Sovelluskerros on kerros, jonka käyttäjät näkevät ja jonka kautta IoT- ekosysteemin laitteiden ohjaaminen tapahtuu tai joka näyttää järjestelmään kuuluvien laitteiden sensoriarvot. Sovelluskerros on vastuussa muille IoT-kerroksille kerätyn ja siirretyn datan käsittelystä. Sovelluskerros suodattaa, prosessoi ja jakaa mittaustiedot käyttäjälle tai muille IoT-järjestelmässä oleville laitteille. Sovelluskerros sisältää myös IoT-laitteen älykkyyden, joka tyypillisesti toteutetaan erilaisilla algoritmeillä. Algoritmien perusteella IoT-laitteessa tehdään tarvittavaa laskentaa, mittausdatan esikäsittelyä jne. Näiden tekijöiden vuoksi sovelluskerroksen odote-



Kuva 3.2: IoT-järjestelmän kerrosarkkitehtuurimalli

taan täyttävän korkeat turvallisuusvaatimukset [40].

Kuljetus: Tämä kerros kuvaa tiedon siirtoa havaintokerroksen ja käsittelykerroksen antureiden välillä eri verkkojen kautta.

Käsittely: Joskus kutsutaan väliohjelmistokerrokseksi, tämä tallentaa, analysoi ja esiprosessoi kuljetuskerrokselta tulevat tiedot. Nykyaikaisissa ohjelmistosovelluksissa tämä sijaitsee usein pilven reunalla alhaisen latenssin viestintää varten.

Business: Tätä kerrosta kutsutaan usein Business Intelligence -tasoksi. Sovelluskerrosta korkeammalla tasolla sijaitseva Business-taso kuvaa kaikkea, mikä liittyy sidosryhmiin. Päätöksenteko tehdään täällä sovellustasolla löydettyjen ja kulutettujen tietojen perusteella.

3.2 IoT-laitteet

IoT-laitteet ovat tietokoneiden standardista poikkeavia tietokonelaitteita, jotka muodostavat langattoman yhteyden verkkoon ja joilla on kyky siirtää tietoa, samaan tapaan kuten monilla muillakin laitteilla Internetissä [9]. IoT-laitteet voivat kommunikoida ja olla vuorovaikutuksessa Internetin kautta. Niitä voidaan myös valvoa ja ohjata etänä. IoT-laitteilla on mahdollista kommunikoida laitteiden välillä (*engl. Machine-to-Machine, M2M*) ja laitteen ja ihmisen kesken. IoT-laitteet ovat siten osa IoT-järjestelmää, jossa jokainen laite keskustelee muiden vastaavien laitteiden kanssa. IoT-laitteita käytetään tyypillisesti ympäristöissä, joissa automatisoidaan esimerkiksi kodin ja teollisuuden tehtäviä. IoT-laittet voivat välittää sensoridataa käyttäjille, yrityksille ja muille osapuolille [27]. IoT-laitteet luokitellaan usein kolmeen pääryhmään, eli kuluttaja-, yritys- ja teollisuuskäyttöön kuuluviin IoT-laitteisiin [1].

IoT-järjestelmien valtavan kasvun myötä saatavilla on valtava valikoima IoT-laitteita. Nämä laitteet vaihtelevat ominaisuuksiltaan, mutta ne on yleensä suunniteltu jonkin tilatiedon ylläpitoon tai toiminnan tehokkuuden parantamiseen. Tällaisia ovat esimerkiksi älykkäät lukot, älykkäät termostaatit, älykäs valaistus ja älykäs tilojen suojaus ilkivallalta, tulipaloilta tai vesivahingoilta. Näistä teknologioista on olemassa paljon kuluttajaversioita.

Industrial IoT -laitteet (IIoT) on suunniteltu käytettäviksi tehtaissa tai muissa teollisuusympäristöissä. Useimmat IIoT-laitteet ovat sensorinoodeja, joita käytetään kokoonpanolinjan tai muun valmistusprosessin valvontaan. Mittaustiedot eri tyyppisistä sensoreista siirretään valvontasovelluksiin, jotka varmistavat, että keskeiset

prosessit toimivat optimaalisesti. Kunnonvalvontaan suunnitellut IoT-laitteet voivat myös estää seisokkeja ennustamalla milloin osat on vaihdettava.

Erilaisten kodin IoT-laitteiden kuten Amazon Echo tuoteperheen, robottipölynimurien, älyjääkaappien ja Google Home viihde- ja älylaitteiden, käyttöönoton myötä IoT-laitteiden lukumäärä on kasvanut viime vuosina räjähdysmäisesti. Viime aikoina markkinoille tulleiden laitteiden käyttöönotto on äärimmäisen helppoa. Myös käytettävyys on parantunut huomattavasti. Laitevalmistajan tarjoavat myös pilvipalvelun laitteiden käytön helpottamiseksi. Tämä kehitys on johtanut siihen, että turvallisuus ja yksityisyyden suojaaminen ovat jäänet taka-alalle saavutettavuuden kustannuksella. Bastos et al. [9] kirjoittavat, että pahantahtoisille toimijoille on tarjolla otollinen tilaisuus hyödyntää kodeissa yleistyviä haavoittuvia laitteita. Parin viime vuoden aikana tapahtuneet palvelunestohyökkäykset ovat osoittaneet, että hyökkäys IoT-laitteisiin voi olla sekä helppo että sillä voi olla myös tuhoisia seurauksia. Bastos et al.[9] kirjoittavat, että Internet-palvelut kärsivät valtavista hajautetuista palvelunestohyökkäyksistä *(engl. Distributed Denial of Service, DDoS)*, jotka on toteutettu kaapattujen IoT-laitteiden avulla. Lisäksi markkinoille tulee koko ajan paljon halpoja IoT-laitteita, joissa on vain vähän tai ei ollenkaan tietoturvaominaisuuksia.

IoT-laitteet sisältävät usein minimalistisen reaaliaikaisen käyttöjärjestelmän (*engl. Real-Time Operating System, (RTOS)*). Käyttöjärjestelmää käytetään IoT-laitteen prosessien ja muistin hallintaan sekä tietoliikenteen toteuttamiseen erilaisia tietoliikenneratkaisuja käyttäen. RTOS:n valintaan vaikuttaa IoT-laitteen vaatima suorituskyky, tietoturva ja toiminnalliset vaatimukset. Kaikkien IoT-laitteiden ei tarvitse suorittaa samoja tehtäviä. Kaikilla IoT-laitteilla on kuitenkin oltava yhteisiä toimintoja, jotta ne voidaan luokitella IoT-laitteiksi. Yhteiset piirteet tai toiminnot voidaan listata seuraavasti [27]:

- IoT-laitteen on kyettävä tunnistamaan fyysinen ympäristönsä. Sen on kyettävä hakemaan tiedot ulkoisesta ympäristöstään.
- IoT-laitteen keräämän tiedon ja informaation siirtäminen muille laitteille Internetverkon tai muun yhteyden kautta.
- Tiedon kerääminen on merkityksetöntä, jos tietoa ei voi hyödyntää. Siksi IoT-laitteen on kyettävä analysoimaan keräämänsä tiedot.
- IoT-laitteessa on oltava mekanismi, jolla varmistetaan, että IoT-laite on loppu-

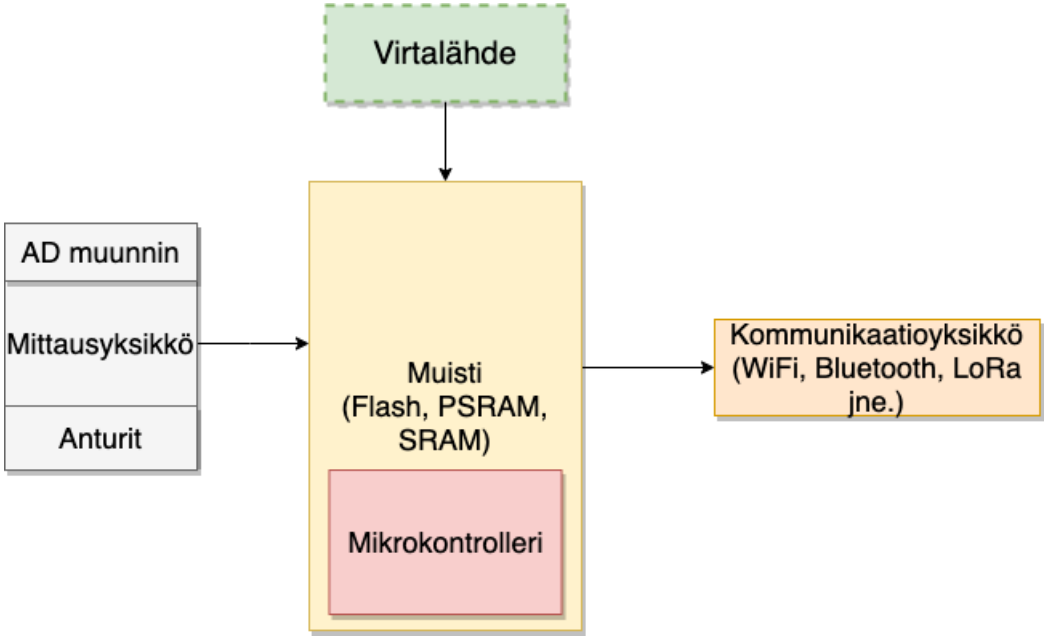
käyttäjän suorassa hallinnassa. Muuten se voi mahdollistaa jopa hakkeroinnin.

Vaikka IoT:n määritelmä vaihtelee hiukan eri näkökulmista katsottuna, on IoT-maailmassa joitakin keskeisiä käsitteitä, jotka erottavat sen muista tietoliikenneteknologioista. Nämä eroavuudet vaikuttavat myös kehitettävän oppimisympäristön suunnitteluun ja toteutukseen. Keskeisimmät erot on kuvattu IoT Security Foundation-organisaation julkaisemassa ja vertaisarvioidussa julkaisussa [27]:

- **Yhteydet**: IoT-laite on aina yhdistetty joko Internetiin tai vähintäänkin paikalliseen verkkoon.
- **Tunnistus**: IoT-laite tunnistetaan yksilöllisesti verkossa siten, että tiedoilla on kyseisen laitteen tunnistama konteksti. Lisäksi itse IoT-laitteen sisältämä sovellusohjelma (*engl. software*) tai laiteohjelmisto (*engl. firmware*) on mahdollista etäpäivittää (*engl. Over The Air, OTA*). Etäohjaus mahdollistaa myös IoT-laitteen diagnostiikan seuraamisen.
- **Autonominen toiminta**: IoT-järjestelmät on suunniteltu niin, että ihminen puuttuu asiaan mahdollisimman vähän tai ei lainkaan. Kukin laite kerää tietoja ympäristöstä, johon se on asennettu, ja voi sitten viestiä tiedot muiden laitteiden kanssa ja mahdollisesti vastatakseen laitteelle tuleviin ohjauskomentoihin.
- **Yhteentoimivuus**: IoT-ratkaisun laitteet vaihtavat tyypillisesti viestejä keskenään, mutta ne eivät välttämättä kuulu yhdelle toimittajalle.
- **Skaalautuvuus**: IoT-järjestelmät pystyvät horisontaalisesti skaalautumaan vastaamaan kasvavaan työmäärään. Uusi laite lisätään tarvittaessa kapasiteetin lisäämiseksi sen sijaan, että nykyinen korvattaisiin paremmalla laitteella.

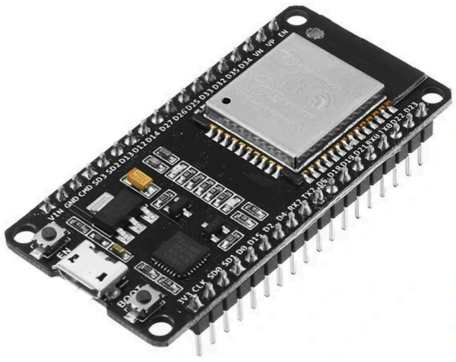
Seuraavaksi esitellään muutamia IoT-laitteita, joita on mahdollista käyttää IoT-oppimisympäristön suunnittelussa ja rakentamisessa. Keskeisenä valintakriteerinä ovat edullisuus, avoimet rajapinnat, luokkakirjastojen saatavuus ja mahdollisuus tutkia laitteessa olevaa ohjelmakoodia tai asentaa laitteeseen omaa ohjelmakoodia. Esiteltävien IoT-laitteiden tai IoT-järjestelmien rakentamiseen soveltuvien kehitysalustojen valintakriteerinä on myös langattoman sensorinoodin (*engl. Wireless Sensor Node*) ja langattoman sensoriverkon (*engl. Wireless Sensor Networks*) toiminnallisuuden toteuttaminen. Opetuskäyttöön soveltuvan IoT-laitteen ja langattoman sen-

sorinoodin arkkitehtuurin tulee noudattaa pääpiirteittäin kuvassa 3.3 olevaa arkkitehtuuria. Kuvasta nähdään, että tyypillinen sensorinoodi sisältää mikrokontrollerin, kommunikaatioyksikön (radion), analogia-digitaalimuuntimen, virtalähteen ja antureiden liittämistä varten kytkentänastoja (*engl. General Purpose Input Output, GPIO*. Yleisesti edellä mainituista käytetään nimitystä GPIO-pinnit.



Kuva 3.3: Langattoman IoT-laitteen arkkitehtuurikuvaus

ESP32: ESP32-kehitysalusta on edullinen WiFi- ja Bluetooth radiot sisältävä System on Chip (SoC) kehitysalusta. ESP32 pohjaiset kehitysalustat ja piirikortit valmistaa Espressif System. ESP32 integroi WiFi- ja Bluetooth-ratkaisut yhdelle sirulle. ESP32 tukee sekä vähävirtaista Bluetooth Low Energy -tekniikkaa (BLE) että vanhempia Bluetooth-yhteyksiä. SoC-piirissä on kaksi prosessoriydintä, joita voidaan ohjata erikseen. ESP3-kehitysalustan CPU:n kellotaajuus on säädettävissä 80 MHz - 240 MHz. SoC-piirissä on myös pienitehoinen prosessori, jota voidaan käyttää suorittimen sijasta suorittamaan tehtäviä, jotka eivät vaadi paljon laskentatehoa. ESP32:een on integroitu oheislaitteita ja erilaisia antureita, kuten kapasitiivinen kosketusanturi, Hall-anturi ja SD-korttilukija. ESP32:ssa on Ethernet, SPI-, UART- ja IIC-väylät sensoreiden ja oheislaitteiden liittämistä varten [18]. Kuvassa 3.4 on opetus- ja tuotekehitykseen sekä prototyyppien kehittämiseen hyvin soveltuva edullinen, mutta ominaisuuksiltaan monipuolinen kehitysalusta.



Kuva 3.4: ESP32 Wroom 32D kehityslusta [18]

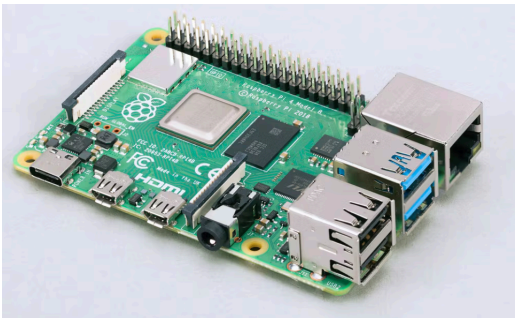
ESP32-kehitysalustan BLE- ja WiFi-integraatiot mahdollistavat hyvin monipuolisen käytön erilaisissa IoT-ratkaisuissa. WiFi mahdollistaa kohtuullisen pitkän fyysisen kantaman ja suoran yhteyden Internetiin WLAN-reitittimen kautta. Bluetooth-yhteyden avulla voidaan muodostaa yhteys puhelimeen, jolloin puhelimen voi toimia ns. välityspalvelimena (*engl. Gateway*) tiedon siirtämisessä Internettiin. ESP32-sirun lepovirta on alle 5 A, joten se sopii vähävirtaisiin IoT-ratkaisuihin tai puettavaan elektroniikkaan. ESP32 tiedonsiirtomoduuli tukee jopa 150 Mbps:n tiedonsiirtonopeutta, ja ESP32 radion 20 dBm lähtöteho antennissa mahdollistaa hyvän kuuluvuuden. ESP32:ssa on freeRTOS käyttöjärjestelmä ja ESP32 tukee myös salattua OTA-laiteohjelmiston päivitystä [18].

LILYGO TTGO T-Journal ESP32-kameramoduulin kehityskortti TTGO T-Journal on halpa ESP32-kamerakehityskortti 3.5, jossa on OV2640-kamera, antenni, 0,91 tuuman OLED-näyttö, muutamia avoimia GPIO-pinnejä ja mikroUSB-liitäntä [21]. Koodin lataaminen kehitysalustalle on helppoa ja nopeaa. ESP32-pohjainen kameramoduuli sopii hyvin oppimisympäristöön, koska ohjelmointi voidaan tehdä Arduino IDE:llä ja siihen on kohtuullisen helppo ohjelmoida haavoittuvuuksia.

Raspberry Pi Raspberry Pi 3.6 on Englantilaisen Raspberry Pi Foundation -säätiön valmistama yksilevyisten tietokoneiden sarja, jonka tavoitteena on helpottaa tietojenkäsittelykoulutuksen saatavuutta ja toteuttamista. Raspberry Pi 3.6 julkaistiin vuonna 2012, ja sen jälkeen on julkaistu useita iteraatioita ja muunnelmia. Alkuperäisessä Pi:ssä oli yksiytiminen 700 MHz:n prosessori ja vain 256 Mt RAM-muistia. Uusimmassa Raspberry Pi 4B-mallissa on 1,5 GHz:n neliydinprosessori ja versiosta riippuen 4 - 32 Gt RAM-muistia. Raspberry Pi:n hinta on aina ollut alle 100 dollaria

Kuva 3.5: LILYGO TTGO T-Journal ESP32-kameramoduulin kehityskortti [21]

[52].



Kuva 3.6: Raspberry Pi-tietokone [52]

Raspberry Pi on halpa tietokone, jossa on avoimen lähdekoodin Linux-pohjainen käyttöjärjestelmä [53]. Raspberry Pi Foundation osallistuu Linux-ytimen ja moniin muihin avoimen lähdekoodin projekteihin sekä julkaisee suuren osan omista ohjelmistostaan avoimena lähdekoodina. Opetuskäyttöön hyvin soveltuva käyttöjärjestelmä on Debian-pohjainen Raspberry Pi OS käyttöjärjestelmä. Raspberry Pi:ssä on myös 40 GPIO-pinniä, joihin voi liittää digitaalisia antureita tai oheislaitteita.

Raspberry Pi on noussut suureen suosioon harrastelija ja kaupallisessa IoT-laitekehityksen alustana. Opetuskäyttöön Raspberry Pi-pohjaiset IoT-laitteet ovat erittäin sopivia. Opiskelijat oppivat nopeasti käyttämään Raspberry Pi:tä ohjelmointitaitojen oppimiseen, laitteistoprojektien rakentamiseen, kotiautomaatioon ja jopa Edge-laskentaan.

IP-kamera: IP-kameralla tarkoitetaan kameraa joka kykenee kommunikoimaan tietoverkoissa. Suurin osa nykyään myytävistä IP-kameroista on langattomassa WLAN-verkossa toimivia IP-kameroita. IP-Kameroita voidaan seurata kaikilla laitteilla, joilla pystyy liittymään Internettiin [42]. Valmistajien tarjoama verkkotallennin on myös

usein käytetty vaihtoehto tallennuksen hoitamiseen. Koska IP-kamerat ovat yhteydessä Internettiin, voidaan niitä käytännössä seurata reaaliajassa mistä tahansa päin maailmaa tahansa [42].

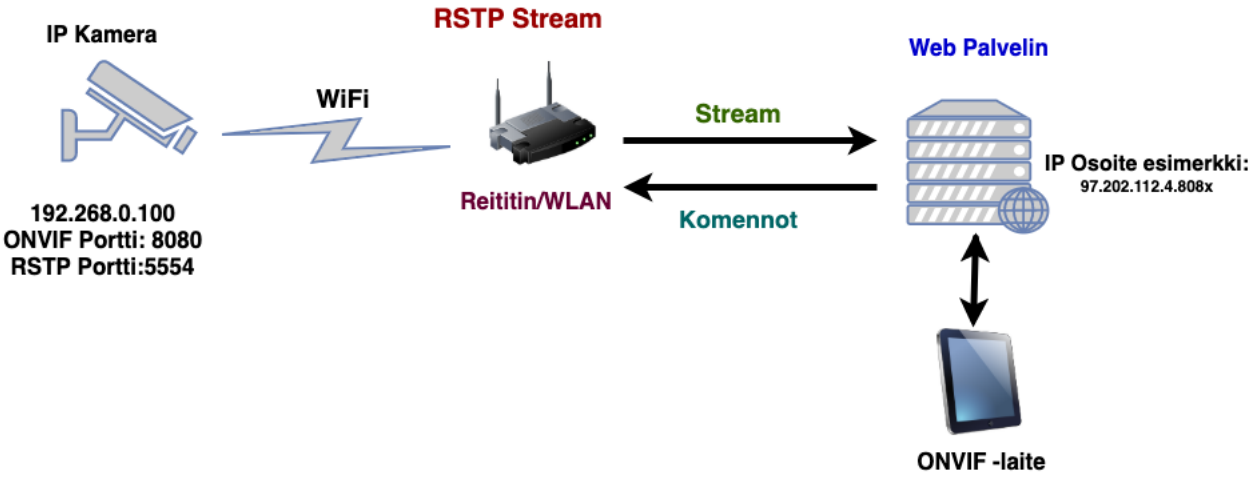
Kaikissa IP-Kameroissa on ns. Web-käyttöliittymä, joka mahdollistaa sen, että kameraa voidaan hallita normaalilla Internet-selaimella. Kameran käyttöä varten tarvitaan joko valmistajan tarjoama sovellus tai kameran IP-osoite. Kameran verkkotallenteelle tai sovellukseen kirjaudutaan käyttäjätunnuksella ja salasanalla. Hallinnasta voidaan katsoa kameran kuvaa, liikutella kameran linssiä sekä säätää kameran asetuksia [42].

IP-kameroista valtaosa tukee myös ONVIF eli Open Network Video Interface Forum:n määrittelemää standardia [42]. ONVIF on kansainvälinen ja avoin foorumi, jonka tarkoituksena on luoda avoin standardi IP-pohjaisille valvontatuotteille. ONVIF standardia noudattamalla IP-pohjaiset laitteet voivat välittää viestejä toisten kanssa. ONVIF-järjestö on perustettu vuonna 2008 [45].

IP-kameroissa käytetään nykyisin RTSP-sovellusprotokollaa, (*engl. Real Time Streaming Protocol, RSTP*), joka on reaaliaikainen suoratoistoprotokolla. RTSP:n avulla voidaan etäohjata tietovirtaa palvelimelta ja antaa palvelimelle komentoja. RTSP mahdollistaa multimediasisällön lähetyksen käynnistäminen, keskeyttäminen ja lopettaminen sekä palvelimella olevien tiedostojen ajastetun käyttämisen [44].

Kuvassa 3.7 on kuvattu RSTP prokotokollan toimintaperiaate. Jotkut RTSP-palvelimet on määritetty sallimaan pääsy mediavirtaan myös ilman salasanaa. Mediavirran URL-osoite ei ole vakio, koska laitteet lähettävät URL-osoitteen, kun laite on yhdistetty olemassa olevien todennuksen jälkeen. Yleensä RTSP toimii porteissa 554, 5554 ja 8554 [45]. Jotta voidaan toistaa RSTP-protokollaa käyttävän IP-kameran videota on tiedettävä URL-osoite sekä kirjautumistunnus ja salasana, mikäli sellaisia vaaditaan [44].
Useimmat IP-kamerat käyttävät samaa porttia sekä HTTP- että ONVIF-liikenteeseen, jolloin IP-kameroiden selainpohjaisen käyttöliittymän (*engl. User Interface, UI*) palvelut ohjautuvat oletusportteihin. Tämä aiheuttaa merkittävän tietoturvariskin, jota pahentaa entisestään puutteellinen käyttäjän todennus, todennuksen puuttuminen tai kovakoodatun käyttäjätunnuksen ja salasanan käyttäminen [1].

Robottipölynimuri: Robotti-imurit ovat yleistyneet kodin IoT-laitteista erityisen nopeaan tahtiin. Koska robotti-imurit sisältävät paljon sellaista sensori- ja tietoliikenneteknologiaa, joiden opettaminen ja ymmärtäminen on erityisen tärkeää tulevai-



Kuva 3.7: RSTP protokollan periaate [44]

suuden IoT-laitteiden suunnittelun kannalta. Robotti-imurit sisältävät sensoriteknologian lisäksi myös kameran, sisätilapaikannuksen ja tekoälyalgoritmejä. Erityisesti sisätilapaikannuksen ja kameroiden yleistyminen IoT-laitteissa aiheuttavat paljon ongelmia sekä tietoturvaan että yksityisyyden suojaan. Robotti-imureita voidaan ohjata painamalla laitteen painikkeita tai mobiilisovelluksen kautta. Koska robotti-imuri käyttää mobillisovelluksen toteuttamiseen pilvipalvelua, voi robotti-imuri paljastaa käyttöpaikan osoitteen ja pohjapiirustuksen. Kalliimmissa robotti-imureissa on runkoon asennettu LiDAR-, 3D-laser- ja kosketusanturit, joiden avulla imuri osaa kartoittaa ympäristönsä ja liikkua itsenäisesti huoneiden ja esteiden mukaan. Edistyneemmät robotti-imurit sisältävät myös tekoälyn, jolloin robotti-imuri pystyy väistelemään esteitä ilman kosketusta [34]. Lidar on kartoitustekniikka etäisyyksien mittaamiseen, jossa lähetetään ultravioletti-, näkyvä- tai lähellä infrapunavaloa oleva säde kohteeseen ja analysoidaan takaisin heijastuvaa valoa antureilla [63].

3.3 IoT-järjestelmän data- ja tietoliikenneprotokollat

IoT-data- ja tietoliikenneprotokollat ovat olennainen osa IoT-teknologiapinoa. Ilman IoT-protokollia ja -standardeja IoT-järjestelmät ja IoT-laitteet olisivat hyödyttömiä eikä nykyisen kaltaista kasvua IoT-laitteiden osalla olisi tapahtunut. IoT-protokollat mahdollistavat laitteiston ja IoT-järjestelmien tiedonvaihdon ja laitteiden keskinäisen viestinnän.

Valitettavasti IoT-protokollat ja -standardit jäävät usein huomiotta, IoT-järjestelmien ja tiedonvälityksen suunnittelussa. Useimmiten huomioidaan vain IoT-laitteiden ja -järjestelmien kyky viestiä erilaisiin pilvipalveluihin. IoT-laitteiden suunnittelussa huomio kiinnittyy laite- ja järjestelmäsuunnitteluun, vaikka laitteiden, IoT-sensoriverkkojen, yhdyskäytävien, palvelimien ja käyttäjäsovellusten välinen vuorovaikutus ovat IoT:n olennaisia osia [8]. Ilman yhteisesti sovittuja standardeja ja tietoliikenne protokollia IoT-järjestelmien tietoliikenne olisi hankalaa ja jopa mahdotonta. Standardit ja protokollat tekevät IoT-järjestelmäsuunnittelusta erityisen tärkeän osa-alueen, koska suurin osa tietoturvahaavoittuvuuksista kohdistuu joko suoraan verkkokerrokseen tai verkkokerroksen välityksellä IoT-laitteisiin.

Seuraavaksi tarkastellaan yleisimpiä IoT-järjestelmien standardeja, dataprotokollia ja viestintäprotokollia. Ensimmäisenä esitellään muutamia IoT-dataprotokollia, joita käytetään vähävirtaisten IoT-laitteiden yhdistämiseen. IoT-dataprotokollien ja -standardien yhteys tapahtuu langallista, langatonta tai matkapuhelinverkkoa käyttäen.

Address Resolution Protocol (ARP): on tiedonsiirtoprotokolla, joka sitoo verkon IP-osoitteet linkkikerroksen MAC-osoitteisiin. ARP-tauluja käyttämällä protokolla voi muuntaa sovellusten käyttämät IP-osoitteet yksittäisen solmun käyttämiksi MAC-osoitteiksi [49].

Secure Shell-protokolla (SSH): SSH-protokollaa käytetään suojattuun etäkirjautumiseen ja tiedostojen siirtoon. SSH-protokolla käyttää oletusarvoisesti porttia 22. SSH-protokolla toimii asiakas-palvelin-periaatteella. SSH-asiakas aloittaa yhteyden ja käyttää julkisen avaimen salausta SSH-palvelimen tarkistamiseen. Kun palvelin-asiakasyhteys on muodostettu, SSH-protokolla käyttää salauksia ja vahvoja tiivisteitä varmistaakseen, että asiakkaan ja palvelimen välillä lähetettävät tiedot ovat turvallisia [66].

Hypertext Transfer Protocol Secure-protokolla (HTTPS): HTTPS-protokollaa käy-

tetään tietoverkoissa viestien välityksessä. Security Sockets Layer (SSL) ja Transport Layer Security (TLS) tarjoavat todentamiseen ja tiedonsiirtoon liittyvää turvallisuutta [59]. Protokollat käyttävät sekä kättelyä että tallennusprotokollaa siirtääkseen kaiken tiedon turvallisesti. Kun asiakas lähettää pyynnön verkkosivustolle, SSL ja TLS antavat asiakkaalle varmenteen. Asiakkaan saamalla varmenteella varmennetaan, että verkkosivusto on oikea ja turvallinen. Varmenteen on verifioinut verkkoselaimien varmenneviranomainen (*engl. certification authority, CA*). [59].

Message Queuing Telemetry Transport (MQTT): MQTT on kevyt IoT-dataprotokolla. Siinä on julkaisija-tilaaja -viestintämalli, joka mahdollistaa helposti toteutettavan tiedonsiirron eri laitteiden välillä [41]. MQTT on suosittu vähävirtaisissa IoT-laitteissa, koska MQTT:n arkkitehtuurin rakenne on yksinkertainen ja kevyt. MQTT toimii TCP/IP-protokollan päällä. MQTT on yksi yleisimmin käytetyistä IoT-dataprotokollista [8].

Data Distribution Service (DDS): DDS on datan jakelupalvelu IoT-protokolla, joka mahdollistaa korkealaatuisen ja skaalautuvan viestinnän IoT:ssä. MQTT:n tapaan DDS toimii myös julkaisija-tilaaja mallilla [41]. DDS on käytössä useissa IoT-järjestelmien laitteissa aina pilvipalveluista hyvin pieniin IoT-laitteisiin. Tämä tekee siitä erityisen sopivan reaaliaikaisiin ja sulautettuihin IoT-järjestelmiin. DDS-protokolla mahdollistaa yhteentoimivan tiedonvaihdon, joka on riippumaton laitteistosta ja ohjelmistoalustasta. DDS on ensimmäinen avoin kansainvälisessä käytössä oleva IoT-väliohjelmistostandardi [8].

Hypertext Transfer Protocol (HTTP): HTTP protokollaa ei suositella IoT-standardina sen tietoliikennekustannusten, akun vähäisen keston ja suuren virrankulutuksen vuoksi. Tästä huolimatta HTTP-protokolla on edelleen käytössä laajasti Web-sivujen protokollana [8]. Web-sivuilla esitetään usein sensoridataa ns. 3-kerrosmallin (*engl. 3-Tier*) mukaisesti toteutettuna. Kolmikerrosmallin mukainen arkkitehtuuri on vakiintunut ohjelmistosovellusarkkitehtuuri, jossa sovellukset on eroteltu kolmeen loogiseen ja fyysiseen tasoon. Ylin kerros on käyttöliittymä, eli esitystasokerros, keskimmäinen kerros on sovellustaso, jossa tietoja käsitellään ennen julkaisua käyttöliittymässä, alin kerros on tietokantakerros, jossa IoT-järjestelmän sensoritietoja tallennetaan ja hallitaan [41].

Constrained Application Protocol (CoAP): CoAP on sovelluskerroksen protokolla ja on yleisesti käytössä IoT-järjestelmien tietoliikenteessä [41]. Vaikka Internetin nykyinen HTTP-protokolla on vapaasti saatavilla ja kaikkien IoT-laitteiden käytet-

tävissä, on se usein liian raskas ja kuluttaa virtaa liikaa. CoAP on virrankulutukseltaan alhainen. Lisäksi CoAP on helppokäyttöinen ja mahdollistaa monilähetystuen. CoAP on erityisen sopiva protokolla käytettäväksi laitteissa, joissa on resurssirajoituksia. Tästä syystä CoAP:a käytetää yleisesti IoTmikrokontrollereissa ja langattomissa sensoriverkojen noodeissa (*engl. Wireless Sensor Network Nodes*). CoAP:a käytetään myös älykkään energian ja rakennusautomaation sovelluksissa [8].

WebSocket kehitettiin alun perin vuonna 2011 osana HTML5-standardia [19]. Yhden TCP-yhteyden kautta voidaan lähettää viestejä asiakkaan ja palvelimen välillä. Kuten CoAP, WebSocketin yhteysprotokolla yksinkertaistaa tietoliikenneratkaisuja, jotka liittyvät yhteyksien hallintaan ja kaksisuuntaiseen viestintään Internetissä. WebSocket sopii IoT-verkkoon, jossa dataa siirretään jatkuvasti useiden laitteiden välillä. Laitteet toimivat yleisimmin asiakkaina tai palvelimina [41].

IoT-dataprotokollien lisäksi IoT-verkkoprotokollia käytetään IoT-laitteiden yhdistämiseen Internetissä. Seuraavana on esimerkkejä yleisimmistä IoT-verkkoprotokollista, joiden tuntemus ja toiminnan ymmärtäminen on erityisen tärkeää. Tästä syystä kehitettävässä oppimisympäristössä tulee olla mahdollisuus testata ja analysoida seuraaksi esiteltäviä IoT-verkkoprotokollien avulla tapahtuvaa liikennöintiä.

WiFi / WLAN on langaton lähiverkkoprotokolla, joka tarjoaa erittäin nopean tiedonsiirron, mutta vaatii enemmän tehoa jatkuvan edestakaisen viestinnän toteuttamiseen. Uudet standardit, esimerkiksi 802.11ah, on suunniteltu pienitehoisiin IoT-laitteisiin [41]. WiFi on jatkossakin IoT:n laajasti käytetty yhteysvaihtoehto. Jotta IoT-laite voi muodostaa yhteyden WiFi-lähiverkkoon (WLAN), tarvitaan laitteessa verkkoliitäntäohjain, joka voi olla erillinen verkkokortti tai integroitu osaksi laitteen sisällä olevaa piirisarjaa. WiFi käyttää datan lähettämiseen Ethernet-tyylisiä datapaketteja 2.4 GHz tai 5 GHz radioviestintätaajuuksia käyttäen [8].

Bluetooth verkkoyhteysprotokolla on toteutettu taajuushyppely periaatteella. Kantavuuden rajoituksista huolimatta Bluetooth on saavuttanut suuren käyttäjäkunnan, koska se on integroitu nykyaikaisiin älypuhelimiin, tabletteihin, puettavaan teknologiaan laitteisiin ja uusimpiin terveysteknologian IoT-laitteisiin. Uusimmassa Bluetooth 4.0 -standardissa on 40 kanavaa ja 2Mhz:n kaistanleveys. Tämä takaa jopa 3 Mb/s maksimitiedonsiirtokapasiteetin [11]. Bluetooth 4.0 tekniikka tunnetaan nimellä Bluetooth Low Energy (BLE), ja se sopii erityisen hyvin pientä virrankulutusta vaativiin IoT-järjestelmiin.

LoRaWAN on Media Access Control (MAC) IoT-protokolla. LoRaWAN mahdollis-

taa pienitehoisten laitteiden kommunikoinnin suoraan Internetiin kytkettyjen sovellusten kanssa pitkän kantaman langattoman yhteyden kautta. Lisäksi se voidaan yhdistää sekä OSI-mallin toiseen että kolmanteen kerrokseen. Se on toteutettu LoRa- tai FSK-modulaation päälle teollisille, tieteellisille ja lääketieteellisille (ISM) radiotaajuuksille [57].

IEEE 802.15.4 on tärkeä muiden IoT-protokollien fyysisenä ja datalinkkikerroksena, mukaan lukien ZigBee, 6LoWPAN, WirelessHART ja Thread. Pohjimmiltaan 802.15.4 on suunniteltu toimimaan joko point-to-point- tai tähtitopologioilla, ja se on ihanteellinen käytettäväksi vähävirtaisissa IoT-laitteissa tai hitaissa tiedonsiirtonopeuksissa [26]. Lisäksi 802.15.4-laitteet toimivat 915 MHz, 868 MHz ja 2,4 GHz taajuusalueilla ja tukevat tiedonsiirtonopeuksia 250 kb/s asti. Fyysinen kerros on vastuussa RF (Radio Frequency)-verkkoon pääsyn hallinnasta, kun taas MAC (Media Access)-kerros on vastuussa kehysten lähetyksen ja vastaanoton hallinnasta datalinkille [57].

Matkapuhelinverkot Long Term Evolution (LTE) on nykyisin paljon käytetty yhteysvaihtoehto langattomien sensoriverkkojen yhteyden muodostamisessa Internettiin. LTE on neljännen sukupolven, eli 4G langaton tiedonsiirtotekniikka. Tyypillisessä LTE-verkossa SIM-kortin muistiin tallennetut todennustiedot mahdollistavat luotettavan todennuksen operaattorin Authentication Centerin (AuC) avulla [57]. Tulevaisuuden 5G-mobiiliverkko voi tarjota lisää tietoliikennevaihtoehtoja IoT-järjestelmille, jotka perustuvat korkeampaan suorituskykyyn ja kykyyn tukea monia yhtäaikaisia yhteyksiä.

Tietoturvan kannalta kaikissa edellä mainituissa standardeissa ja tietoliikenneprotokollissa on haavoittuvuuksia, joiden ymmärrys ja tunnistaminen on olennaisen tärkeää. Perustana oppimiselle on perehtyminen dataprotokollien toimintaan teoreettisen tiedon lisäksi.

4 IoT-järjestelmän tietoturva

Kaikkiin internetissä oleviin laitteisiin liittyy erilaisia turvallisuusongelmia. Suomessakin viime aikoina paljastuneet hyökkäykset IoT-tietojärjestelmiä vastaan paljastivat monia tietoturva-aukkoja ja räikeitä ongelmia, joihin olisi puututtava. Tieto- ja kyberturvallisuuden näkökulmasta olisi valmistuvilla opiskelijoilla oltava ymmärrys siitä, kuinka sekä yksittäiset hyökkääjät että myöskin valtiolliset toimijat voivat nähdä IoT-laitteiden lisääntymisen mahdollisuutena hyödyntää niitä. Pohjimmiltaan hyökkääjä hakee taloudellista hyötyä IoT:n käyttäjien kustannuksella, mutta valtiolliset hyökkääjät pyrkivät myös vaikuttamaan kriittistä infrastruktuuria ja IoT-palveluita tarjoavien ja operoivien palvelun luotettavuuteen ja sitä kautta myöskin näiden yritysten liiketoimintaan. Julkaistujen tutkimusten tulokset osoittivat, että IoT-laitteet ovat haavoittuvia ja vaarantavat käyttäjätiedot. Lähes 40 prosentissa IoT-laitteista löydettiin merkittäviä haavoittuvuuksia [65]. Useiden tapaustutkimusten mukaan markkinoilla on kuluttajien IoT-laitteita, kuten IP-kamerat, älytelevisiot, kodin koneet ja terveydenhuoltojärjestelmät, joissa on havaittu runsaasti tietoturvapuutteita [65]. IoT on vahvasti riippuvainen langattomista verkoista, joiden tiedetään olevan alttiita kaiken tyyppisille tunkeutumisille, mukaan lukien luvaton reitittimen käyttö, liikenteen häirintä, Man-In-The-Middle-hyökkäykset, erilaiset huijaukset, palvelunestohyökkäykset, brute force -hyökkäykset ja tietoliikenteeseen kohdistuvat injektiot [39]. Tässä luvussa määritellään, mitä IoT, Kyberturvallisuus ja tietoturva tarkoittavat. Tietoturvaan myös liittyy osittain myöskin tietosuoja, jota käsitellään yleisellä tasolla tässä luvussa.

Aliluvussa 4.1 käsitellään IoT tietoturvan teoreettista taustaa IoT järjestelmissä ja langattomissa sensoriverkoissa. Aliluvussa 4.2 käsitellään kyberturvallisuuden teoreettista taustaa ja määritelmiä, joka samalla rajaavat sisältöä ja edesauttavat aihealueen ymmärtämistä. IoT-järjestelmien suunnitteluun ja toteuttamiseen liittyy useita haasteita, jotka tulee ratkaista ennen tuotteen tai järjestelmän julkaisua. Näitä haasteita käsitellään aliluvussa 4.3. Tietoturvan toteuttaminen on vaikeaa, koska IoT-laitteilla on tyypillisesti pieni muistikoko, rajoitettu laskentakyky ja IoT-laitteet käyttävät kaikille avoimia tiedonsiirtokanavia. Aliluvussa 4.4 luodaan katsaus alati muuttuvaan IoT-tieturvamaailmaan. Luvussa esitellään luettelon omaisesti suu-

rimpia havaittuja puutteita aina käyttäjien osaamispuutteista laitevalmistajien välinpitämättömyyteen. Kaikilla näillä on oleellista vaikutusta IoT-järjestelmien tietoturvaan. Seuraavassa aliluvussa 4.5 käydään läpi merkittävimmät tietoturvauhkat ja haavoittuvuudet, jotka kohdistuvat IoT-järjestelmiin. Tieto- ja kyberturvan analysoinnissa on oleellista erottaa haavoittuvuudet tai risit, jotka kohdistusvat omaisuuteen tai muodostavat uhkan tai uhkia organisaation toiminnalle. Aliluvussa 4.6 kerrotaan millainen penetraatiotestaus on IoT-järjestelmien tietoturvan testausmenetelmänä. Aliluvussa 4.7 esitellään yleisesti käytettäviä uhkamallinnusprosesseja, joiden tavoitteena on lisätä tietoturvallisen IoT-järjestelmän suunnittelun kyvykkyyttä. Uhkamallinnusprosessit toimivat myös tietoturvan testausmenetelmänä. IoT-järjestelmien tietoturvan testaukseen kuuluu olennaisena osana myös tietoliikenteen testaus ja verkkoliikenteen analysointi. Näitä asioita käsitellään aliluvussa 4.8. Aliluvussa 4.9 on lyhyt yhteenveto IoT-järjestelmän tietoturvasta.

4.1 IoT-tietoturva

Monissa tapauksissa kodeissa tai yrityksissä sijaitsevia IoT-järjestelmiä ei suojata lainkaan. IoT-laitteet voivat mahdollistaa myös julkisen pääsyn laitteen mittaamaan tietoon. IoT-laitteet voivat sijaita syrjäisillä alueilla, liikkuvissa ajoneuvoissa tai olla kiinni käyttäjässään. Nämä ominaisuudet tekevät IoT-laitteista ja IoT-järjestelmistä erityisen haavoittuvia, ja siksi IoT on suurin yksittäinen hyökkäysalusta kaikentyyppisille kyberhyökkäykselle. Uutisoinnissa on ollut lukemattomia hakkereita, jotka ovat toteuttaneet hyvin organisoituja kyberhyökkäyksiä ja toteuttaneet jopa kansallisvaltioiden tietoturvaloukkauksia IoT-laitteita hyödyntäen. IoT-laitteissa tulisi olla suojattu käynnistys, suojattu päivitys ja suojatut viestintäominaisuudet luottamuksellisuuden, eheyden ja käytettävyyden varmistamiseksi [27].

4.2 Tietoturva vs Kyberturvallisuus

Teknologiateollisuuden [60] julkaisun mukaan kyberuhkat ovat haitallisia, koska ne vaikuttavat organisaation toimintaan, talouteen, yrityksen hallussa olevaan tietoon ja jopa yrityksen liiketoiminnan jatkuvuuteen. Kyber- tai tietoturvauhka on yleensä määritelty niin, että se tarkoittaa sellaista uhkaa, joka toteutuessaan vaarantaisi yhteiskunnan tai kohdeyrityksen elintärkeän toiminnon.

Russell ja van Duren [57] kirjoittavat, että IoT-tietoturva ei ole käsitteellisesti ky-

berturvallisuutta, koska IoT-tietoturva koskee paljon muutakin kuin pelkkää dataa, palvelimia, verkkoinfrastruktuuria ja tietoturvaa. Pikemminkin se sisältää Internetin kautta kytkettyjen fyysisten IoT-järjestelmien tilan suoran tai hajautetun seurannan ja hallinnan. Kyberturvallisuus ei useinkaan sisällä IoT-laitteiston fyysisiä haavoittivuuksia ja turvallisuusnäkökohtia. IoT-laitteiden prosessien digitaalinen ohjaus verkoissa tekee IoT:stä ainutlaatuisen siinä mielessä, että tietoturva ei rajoitu pelkästään perustietojen varmistusperiaatteisiin, kuten luottamuksellisuuden, eheyden ja kiistämättömyyden varmistamiseen, vaan myös fyysisiin resursseihin. Fyysisiä resursseja ovat esimerkiksi virrankulutus. He jatkavat, että IoT-laitteiden tietoturva ja fyysisten ominaisuuksien suojaus ei siis ole pelkästään tietoverkkoliikenteen tietoturvaa, koska IoT-laitteen turvallisuus riippuu laitteen käytöstä, laitteen sijainnista ja siitä, millaista fyysistä prosessia tai toimintoja laite ohjaa. IoT-laitteen tietoturvaan vaikuttaa myös niiden järjestelmien luonne, joihin IoT-laite on yhteydessä.

Järvisen [30] mukaan termejä tietoturva ja kyberturvallisuus käytetään usein ristiin. Molemmissa on kyse kuitenkin datan suojaamisesta ja tietojärjestelmien toiminnan varmistamisesta ja pääsyn hallinnasta. Edellä mainittujen teknologioiden tavoitteilla on kuitenkin selkeitä eroavaisuuksia. Järvinen [30] kirjoittaa, että tietoturvalla pyritään tietojen, tiedostojen ja yksittäisten tietokoneiden ja verkossa olevien IoT-laitteiden suojaamiseen luvattomalta käytöltä ja luvattomilta käyttäjiltä.

Järvisen [30] mukaan kyberturvallisuustermi tuli Suomeen vuonna 2011. Valtioneuvoston ulko- ja turvallisuuspoliittinen ministerivaliokunta päätti saman vuoden maaliskuussa käynnistää kansallisen kyberturvallisuusstrategian laatimisen. Tavoitteena strategialla oli parantaa verkkohyökkäysten havainnointikykyä sekä kyberuhkien valvontaa ja ennaltaehkäisyä. Rouskun ja Järvisen [56] mukaan kyberturvallisuus keskittyy vahvasti ICT-järjestelmien turvaamiseen niiden toimintaa uhkaavia riskejä vastaan. Pääpainon tulee olla niissä ympäristöissä, jotka ovat yhteyksissä tietoverkkoihin ja etenkin palvelinyhteyksiin.

Useimmat tieto- ja kyberturvallisuuden liittyvät ongelmat ja havaitut haavoittuvuudet liittyvätkin internetin kautta tehtyihin kyber- ja tietoturvahyökkäyksiin. Viimeaikaisilla, uutiskynnyksenkin ylittäneillä, hyökkäyksillä on käytetty hyväksi huonoa tai puutteellista tietoturvallisuutta. Näillä hyökkäyksillä on rikottu yksityisyyden suojaa sekä haavoitettu toimintojen käytettävyyttä. Usein kyseessä on ollut myöskin palvelunestohyökkäys, jolloin ei ole ollutkaan tavoitteena päästä käsiksi sensitiiviseen dataan.

Järvisen [30] mukaan termejä tietoturva ja kyberturvallisuus käytetään usein ristiin. Molemmissa on kyse kuitenkin datan suojaamisesta ja tietojärjestelmien toiminnan varmistamisesta ja pääsyn hallinnasta. Edellä mainittujen teknologioiden tavoitteilla on kuitenkin selkeitä eroavaisuuksia. Järvinen [30] kirjoittaa, että tietoturvalla pyritään tietojen, tiedostojen ja yksittäisten tietokoneiden ja verkossa olevien IoT-laitteiden suojaamiseen luvattomalta käytöltä ja luvattomilta käyttäjiltä.

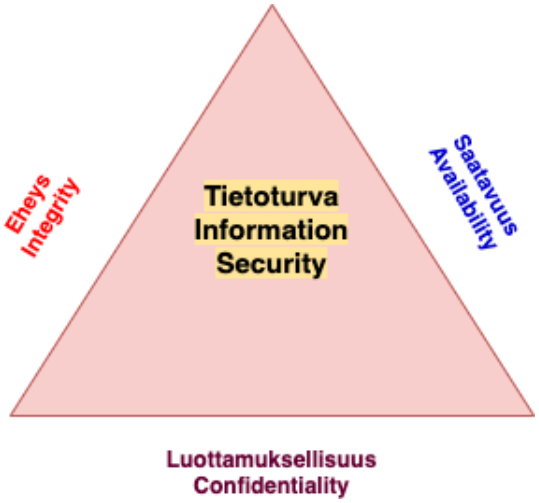
Järvinen jatkaa, että tietoturvallisuus tulee suhteuttaa suojattavan kohteen arkaluotoisuuden mukaisesti. Tiedon arvo ja tiedon luonne määrittelevät sen, kuinka paljon tietoturvaan kannattaa käyttää aikaa ja rahaa. Järvinen korostaa, että tietoturva ja tietosuoja voivat kuulostaa samalta ja siten aiheuttavat sekaannusta. Tietoturvan kohteena on itse tieto ja tietosuojan kohteena on ihminen. [30]

4.3 IoT-järjestelmän tietoturvahaasteet

Langattomat sensoriverkot (WSN-verkot) koostuvat useasta sensorinoodeista, jotka useimmiten ovat suurimman osan ajasta lepotilassa. Sensorinoodit ovat hyvin usein myös sijoitettu pysyvästi mittauspaikkaan. IoT-verkot taas ovat erittäin heterogeenisiä. IoT-verkkojen laitteiden on oltava koko ajan saatavilla ja osa sen laitteista saattaa olla hyvinkin liikkuvia [68]. Langattomien sensoriverkkojen laitteilla on tyypillisesti pieni muistikoko, rajoitettu laskentakyky ja ne ovat usein paristokäyttöisiä sekä matalalla tiedonsiirtonopeudella. IoT- ja langattomissa sensoriverkoissa voidaan myös kuljettaa hyvin arkaluontoista tietoa ja verkot ovat usein integroitu kriittisiin infrastruktuureihin. Nämä ominaisuudet yhdessä epäluotettavan, kaikille avoimen tiedonsiirtokanavan kanssa, tekevät tietoturvan toteuttamisen vaikeaksi [40]. Monet tämän hetkiset tietoturvamekanismit kuluttavat myös huomattavan määrän energiaa ja ne ollaan suunniteltu vain tiettyyn haavoittuvuuteen eivätkä näin ollen pysty kokonaan vastaamaan WSN- ja IoT-verkkojen vaatimuksiin [54]. Tietoturvamekanismien vaatimuksissa edellytetään, että ne pystyvät takamaan verkon turvallisen tiedonsiirron varmistaen, että lähetettävä tieto on aitoa, eheää ja että tieto tulee valtuutetulta semsorinoodilta. Tiedon on oltava myös ajantasaista ja sensorinoodien on oltava usein reaaliaikaisesti saatavissa. Tietoturvan kannalta on myös tärkeää, että koko WSN-verkko ei saa vaarantua, jos sensorinoodi on joutunut verkkohyökkäyksen kohteeksi [28].

CIA-kolmiolla 4.1 kuvataan tyypillisesti tietoturvalle asetettavia minimivaatimuksia. Kirjainyhdistelmä CIA tulee englannin kielen sanoista confidentiality (luot-

tamuksellisuus), integrity (eheys) sekä availability (saatavuus) [39]. Tietoturvan CIA-kolmiota voidaan hyödyntää myös IoT-laitteiden ja IoT-järjestelmän tietoturvan analysoinnissa [6].



Kuva 4.1: Tietoturvan CIA-kolmio

1. Luottamuksellisuus tarkoittaa, että tiedot ovat vain ja ainoastaan valtuutettujen käyttäjien luettavissa,
2. Eheys tarkoittaa, että tieto on muuttumatonta elinkaaren aikana,
3. Saatavuus tarkoittaa, että tiedot ovat milloin tahansa käyttäjien saatavilla,

4.4 IoT-järjestelmän tietoturvan nykytilan analysointi

Seuraavassa on luettelo IoT tieto- ja kyperturvallisuuteen liittyvistä ongelmista. Julkaisujen pohjalta tehdyn analyysin pohjalta suurin osa tutkijayhteisöstä on samaa mieltä puutteista, joita tulisi IoT-laitteiden valmistajien ja ohjelmisto-osaajien tiedostaa ja hallita.

Tuotannon vaatimustenmukaisuuden puute: Useimmat IoT-tietoturvaongelmat johtuvat valmistajista, jotka eivät käytä tarpeeksi aikaa ja resursseja tuotteidensa turvallisuuteen [12]. Tutkimusten mukaan TTM, eli Time to Market ajattelu vaikuttaa siihen, että IoT-laitteiden testauksessa ei huomioida läheskään kaikkia tietoturva-

puutteita. Myös ohjelmistokehittäjien asenteissa ja osaamisessa on havaittu olevan merkittäviä puutteita [23].

Heikot ohjelmistojärjestelmät ja haittaohjelmahyökkäysten haavoittuvuudet: Tutkijat ovat osoittaneet, että IoT-laitteet ovat erittäin suojaamattomia haittaohjelmahyökkäyksiä vastaan [48]. Perimmäiset syyt, jotka mahdollistivat tämän tilanteen, voidaan tiivistää seuraavasti:

1. Selkeiden ja laajalti hyväksyttyjen tietoturvastandardien puute IoT-laitteille ja samalla laitteet julkaistaan ilman valmiita suojausominaisuuksia,

2. Suurin osa IoT-alustoista otetaan käyttöön ja konfiguroidaan ja joskus jopa suunnitellaan ilman että tietoturvallisuutta ajatellaan [48].

IoT-järjestelmässä tulisi olla suojaustoimintoja, joiden on oltava seurausta huolellisesta suunnittelusta ja toteutuksesta [48]. Näillä toiminnoilla taataan riittävän hyvä tietoturvataso, mahdollistetaan ohjelmistopäivitykset ja pyritään käytön ja asennuksen yksinkertaisuuteen. Nämä suojaustoiminnot tulee olla myös pienissä, rajoitetuissa, akkukäyttöisissä IoT-laitteissa.

De Donno et al. [15] mukaan hajautetun palvelunestohyökkäyksen (*engl. Distributed Denial of Service, DDoS*) mahdollisen toteutumisen tunnistaminen on erityisen tärkeää oppia. On tärkeää tuntea ja ymmärtää verkon IoT-järjestelmien ja verkontoimilaitten skannausstrategiat, hyökkäyksen leviämismekanismit, vaikutukset uhriin, hyökkäysnopeus ja Botnet-tyyppisten haittaohjelmien asennusmekanismit.

Jos IoT-laitteet eivät saa tietoturvapäivityksiä, ne voivat muuttua tartunnan saaneiksi zombie-koneiksi [24]. Nykypäivänä useimmat näistä IoT-laitteista keräävät tietoja ulkoisesta ympäristöstään ja niissä käsitellään esimerkiksi terveydenhuolto-, teollisuus-, sosiaali- ja viranomaistietoja. Bottiverkkohyökkäys, salakuuntelu, henkilökohtaisiin ja yritysasetuksiin tunkeutuminen ja IoT-laitteiden kaappaaminen voi aiheuttaa suurta vahinkoa [15].

Käyttäjien tietämyksen ja tietoisuuden puute: IoT:n kehittynyt toiminnallisuus edellyttää hyvää tietoisuutta erityyppisistä tietoturvauhkista ja haavoittuvuuksista. Käyttäjien tietämättömyys voi tehdä heistä sosiaalisten manipulointihyökkäysten uhreja. Vastuu käyttäjien suojaamisesta hyökkäyksiltä on myös laite- ja järjestelmätoimittajilla.

Erilaisten IoT-laitteiden vuoksi IoT-oppimisympäristön suunnittelun ja kehittämisen kannalta on vaikea määrittää oppimistavoitteita tai oppimistehtäviä yksit-

täisen laitetyypin kannalta. IoT-laitteet ovat kuitenkin pohjimmiltaan mikrokontrolleripohjaisia ja sisältävät tunnistus- ja viestintäominaisuuksia. Ne voivat tukea myös käyttö-, tallennus- ja tiedon käsittelyominaisuuksia. Bastos et al. [9] kirjoittavat, että pahantahtoisille toimijoille on tarjolla otollinen tilaisuus hyödyntää kodeissa yleistyviä haavoittuvia laitteita. Parin viime vuoden aikana tapahtuneet palvelunestohyökkäykset ovat osoittaneet, että hyökkäys IoT-laitteisiin voi olla sekä helppo toteuttaa ja että hyökkäyksellä voi olla myös tuhoisia seurauksia. Bastos et al. [9] kirjoittavat, että Internet-palvelut kärsivät hajautetuista palvelunestohyökkäyksistä *(engl.Distributed Denial of Service, DDoS)*, jotka on toteutettu kaapattujen IoT-laitteiden avulla. Lisäksi markkinoille tulee koko ajan paljon halpoja IoT-laitteita, joissa on vain vähän tai ei ollenkaan tietoturvaominaisuuksia.

4.5 IoT-järjestelmän tietoturvauhkat ja haavoittuvuudet

Bastos et al. [9] kirjoittavat, että tietoturvallisuuden ymmärtämiseksi on sisäistettävä neljä keskeistä käsitettä: omaisuus, uhka, haavoittuvuus ja riski. Seuraavassa luettelossa on tarkennettu edellä mainittuja käsitteitä [9]:

1. Omaisuudeksi, joka tarvitsee suojaa katsotaan kaikki, mikä on arvokkasta kuten henkilöstö, materiaalit, operatiiviset toimet ja yrityksen tiedot, immateriaalioikeudet rahavarat jne,

2. Uhka on toimi, mahdollinen toiminta tai toimettomuus, joka todennäköisesti aiheuttaa vahinkoa, haittaa ja omaisuuden tai tiedon menetystä,

3. Haavoittuvuus on omaisuuden, tietoverkon tai tiedon suojauksen heikkous tai aukko, jota voidaan hyödyntää omaisuuteen tai tietoon käsiksi pääsemiseen.

4. Riski on kyseessä aina silloin, kun on pienikin mahdollisuus haitalliseen toimintaan, joka kohdistuu esimerkiksi kodin IoT-laitteisiin [39]. Riskejä sisältyy nykypäivänä esimerkiksi langattomiin sensorinoodeihin, jotka mittaavat esimerkiksi lämpötilaa, kosteutta, savukaasuja ja ilmanlaatua. Riskejä liittyy myös IP-valvontakameroihin, digitaalisiin lukkoihin ja puettaviin laitteisiin. Lisäksi kaikkiin niihin internettiin yhteydessä oleviin IoT-laitteisiin liittyy riskejä, joissa käytetään henkilökohtaisia tunnistetietoja, talletetaan valokuvia tai

videomateriaalia, kommunikoidaan IoT-laitteen kanssa tai laitteen pilvipalvelun kanssa sähköpostia käyttäen. Tällöin riski kohdistuu käyttäjätunnuksiin, salasanoihin tai jopa sormenjälkiin, jos niitä käytetään tunnistauduttaessa IoT-laitteelle tai tietojärjestelmään.

Iot-laitetta, laitteeseen tai sensoriverkkojen tietojärjestelmään kuuluvaa pilvipalvelua, josta puuttuu jokin edellä mainituista tietoturvaan liittyvien uhkien suojausominaisuuksista, voidaan pitää haavoittuvana [39]. Uhkien realisoituessa järjestelmässä oleviin tietoihin on siten mahdollisuus päästä helposti käsiksi. Markkinoilla on runsaasti toimijoita, joiden markkinoimat laitteet ja järjestelmät eivät täytä edellä mainittujen attribuuttien vaatimuksia [9].

Alladin et al. [6] kuvaavat testitapahtumien avulla vakavimpia hyökkäyksiä, ja niiden toteutustapoja puettaviin IoT-laitteisiin. He käsittelevät kahdeksaa yleistä hyökkäystä, jotka kohdistuvat pääasiassa IoT-laitteiden fyysiseen kerrokseen, mutta jonka kautta mahdollistuu hyökkääjän pääsy myös verkko- ja sovelluskerroksen palveluihin [6]. IoT-verkkojen ja IoT-laitteiden haavoittuvuudet voidaan luokitella kolmeen hyökkäystyyppiin, joiden merkitys tietoturvan kannalta on merkityksellisin:

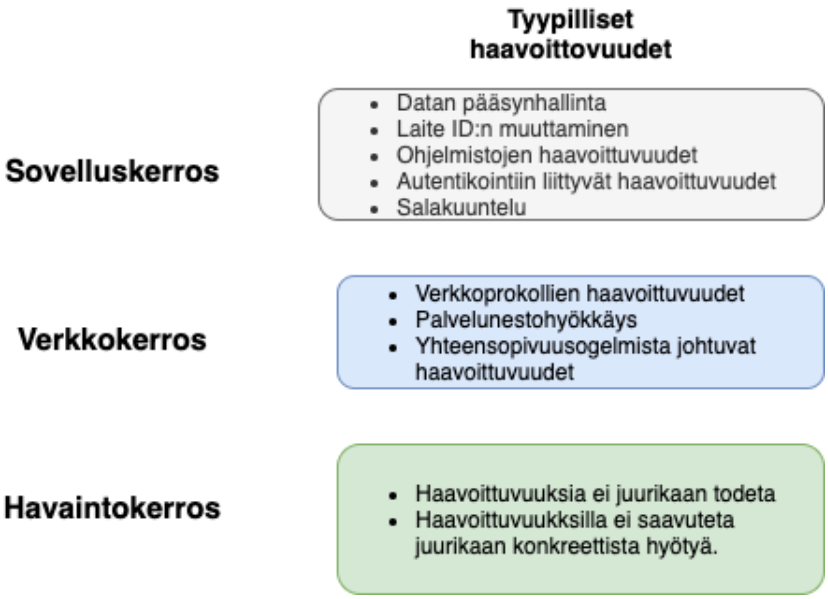
1. Hyökkäykset salassapitoon ja autentikointiin, jossa hyökkääjä suorittaa salakuuntelua,

2. Hyökkäykset palvelun eheyteen, missä verkko on pakotettu hyväksymään vääriä tietoja ja

3. IoT -verkon saatavuuteen kohdistuvat hyökkäykset, jossa hyökkääjä yrittää vähentää verkon kapasiteettia. Hyökkäys toteutetaan tyypillisesti palvelunestohyökkäyksenä (*engl. Denial of Service, DoS*).

Kuvassa 4.2 esitellään lyhyesti tärkeimmät hyökkäykset havainto-, verkko-, ja kuljetuskerroksella [39].

Havaintokerros koostuu IoT-järjestelmän fyysisistä elementeistä. Se sisältää anturit, radiolähettimet ja sensoriyksikön vuorovaikutuksen ulkomaailman kanssa. Havaintokerros liittyy ympäristöön, joka voi sisältää muita IoT-laitteita. Havaintokerros lähettää esimerkiksi mittaustiedot verkko- ja applikaatiokerroksille datan käsittelyä varten [39].



Kuva 4.2: IoT-järjestelmän tyypilliset haavoittuvuudet eri kerroksilla.

Sensorinoodin peukalointihyökkäys on fyysinen hyökkäys, jossa hyökkääjä onnistuneesti kaappaa laitteen ja peukaloi sen elektroniikkapiiriä manuaalisesti. Hyökkäyksen ensisijainen motiivi voi olla sensorinoodin laitetunnisteen, eli DeviceID:n muuttaminen. Laitetunniste tallennetaan hyvin yleisesti ohjelmoitavaan lukumuistiin (EEPROM). Hyökkäyksen onnistuessa hakkeroitu sensorinoodi pystyy esiintymään hyökkääjän määrityksen mukaisesti toisena sensorinoodina [6]. Laitesuunnittelijoiden tulisi varmistua, että sensorinoodissa käytettävä EEPROM-siru on suojattu ja siinä on käytetty jotain tietoturvallista digitaalisista sormenjälkeä.

Sovelluskerros Sovelluskerroksella korostuu erityisesti ohjelmistovirheet, jossa mitä tahansa laitteen laiteajurin tai ohjelmisto haavoittuvuutta voidaan hyödyntää jopa useiden eri tyyppisten hyökkäysten suorittamiseen [39]. Yksi esimerkki voisi olla käyttäjätilin hakkerointi, jossa salasanatodennusvaihe voidaan ohittaa siten, että lopputuloksena on pysyvä käyttäjätili, jolla on pääkäyttäjän oikeudet. Tämä haavoittuvuus käyttää hyväkseen puskurin ylivuotohaavoittuvuutta (*engl. Buffer overflow*). Onnistunut haavoittuvuuden hyödyntäminen mahdollistaa palvelunestohyökkäyksen. Usein palvelunestohyökkäyksiin liitty myös vihamielisiä toimenpiteitä. Hyökkääjä voi esimerkiksi muokata tai luoda mitä tahansa tiedostoa tiedostojärjestelmässä tai suorittaa minkä tahansa käyttöjärjestelmän varatun komennon laitteen ytimessä eli kernelissä [6]. Myös tietojen varastaminen on hyvin yleistä pal-

velunestohyökkäysten yhteydessä.

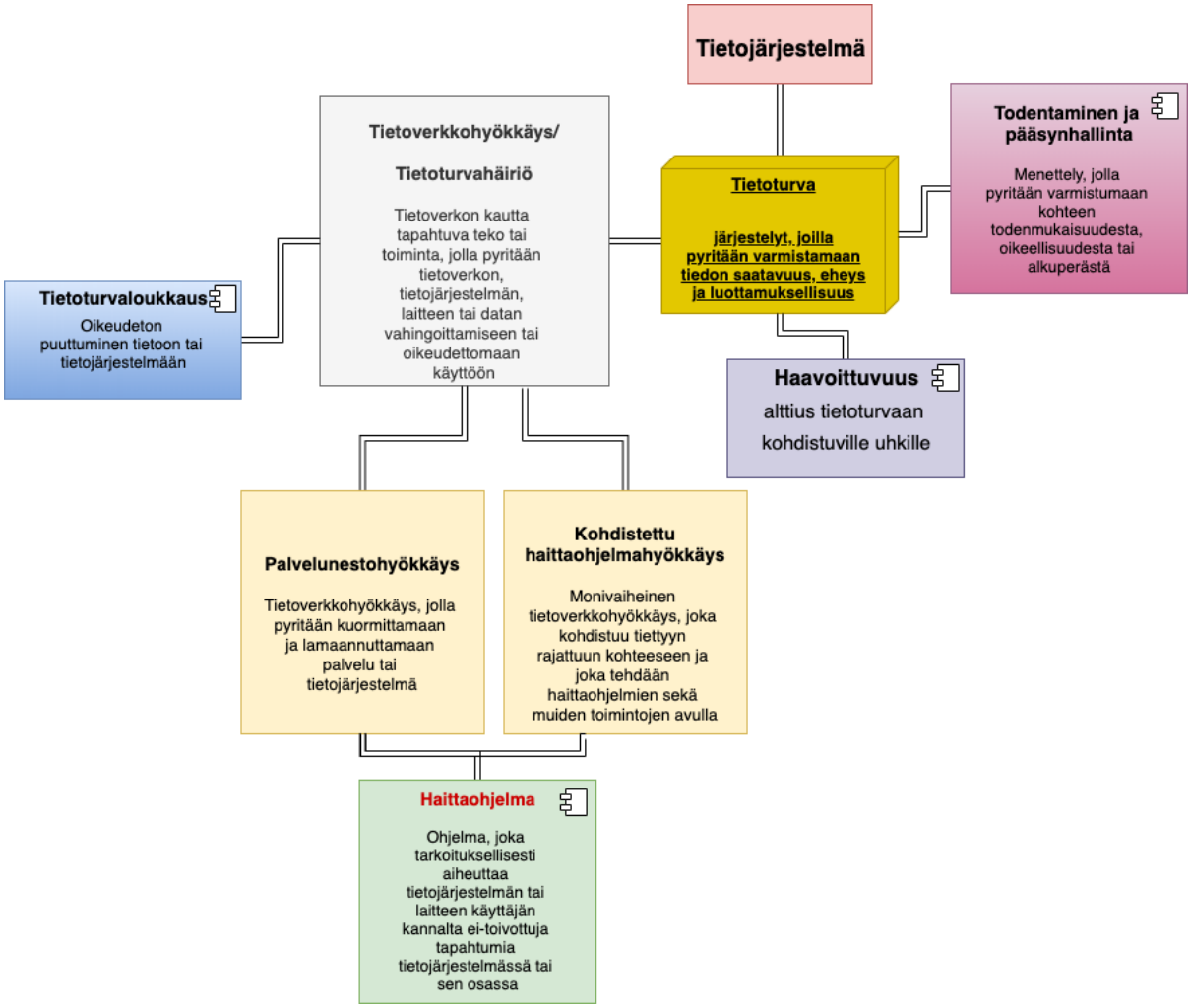
Puskuriylivuodon lisäksi ohjelmistohaavoittuvuus voi olla seurausta varsin harmittomalta vaikuttavalta ohjelmointivirheeltä, jossa pinomuistiin luettavaa salasanan pituutta ei tarkasteta. Helposti käy niin, että käytetään epähuomiossa haavoittuvuuden mahdollistavaa strcpy-funktiota, jossa ei verifioida salasanan pituutta. Ohjelmistokehittäjien olisi osattava tunnistaa ohjelmakoodissa toiminnot, joissa olisi käytettävä tietoturvan kannalta oikeaa srtncpy-funtiota, jossa salasanan tai syötteen pituus varmistetaan. Tämän tyyppinen haavoittuvuus on kuvattu artikkelissa [6], jossa salasanan pituuden tarkistamattomuus voi johtaa luvattomaan WiFi-yhteyden muodostamiseen tai pääsyyn autentikoinnin vaatimaan tiedostoon. Koska puskuriylivuotohaavoittuvuuksia voidaan käyttää tällaisen IoT-laitteisiin kohdistuvan hyökkäyksen toteuttamiseksi, olisi erityisen tärkeää katselmoida koodia ja käyttää vaikkapa staattisia varmennustekniikoita ohjelmistotestauksessa, jossa havaittaisiin haavoittuvuuksia aiheuttavien funktioiden vaarallinen käyttö.

Salakuunteluhyökkäyksessä kohdelaitteelta tulevaa dataliikennettä salakuunnellaan. Hyökkäys voidaan toteuttaa varsin perinteisellä MITM (Man-In-The-Middle) hyökkäyksellä. MITM-hyökkäyksessä pystytään poimimaan kriittistä verkkoinformaatiota, mutta onnistuakseen hyökkäys edellyttää sensorinoodin tai IoT-laitteen suojauksen kiertämisen, jotta haavoittuvuuden sisältämä ohjelman osa saadaan asennettua kohdelaitteelle. Alladin et al. [6] kuvaavat salakuunteluhyökkäyksen toteuttamisen Fitbit-palveluun.

Verkkokerroksen verkkoprokollista sekä IPv4 että IPv6 ovat laajasti käytettyjä useissa eri kohdissa IoT-järjestelmää. Esimerkiksi IPv6 Low Power Wireless Personal Area Networks (6LoWPAN) tukee IPv6:n käyttöä verkkorajoitteisissa ympäristöissä. Lisäksi 6LoWPAN on suunniteltu tukemaan langatonta Internet-yhteyttä pienemmillä tiedonsiirtonopeuksilla sellaisissa IoT-laitteissa, joissa IPv6-paketit kuljetetaan tehokkaasti pienissä linkkikerroksen kehyksissä [61].

Verkkokerroksen tietoturvaa simuloimalla on havainnollistettu kuinka huonosti konfiguroidut IoT-laitteet voivat paljastaa arkaluontoisia tietoja käyttäjistä. Verkkokerroksen suojaamattomuutta simuloitiin tutkimuksessa käyttämällä esimerkkinä Fitbit-palvelua, jossa käyttäjän laite lähettää terveysdataaa Fitbit-palvelimelle langattoman tukiaseman kautta [6]. Vaikka käyttäjätietojen varastaminen näyttää edellä olevassa esimerkissä olevan merkityksetön, hyökkääjä voi löytää verkkopalvelun tunnisteen (SSID) ja verkon todennusavaimen Wiresharkin lokitiedostosta. Artikkelin kirjoittajien ja haavoittuvuuden todentajien mukaan suurin ongelma on se, et-

tä Fitbit Ariasta puuttuu salattu yhteys taustapalvelimen ja IoT-laitteen välillä [6]. Tämä puute mahdollistaa edellä kuvatun kaltaiset salakuunteluhyökkäykset. Vihamielinen hyökkääjä voi hyödyntää haavoittuvuutta myös esiintymällä laillisena tahona ja päästä käsiksi uhrin henkilökohtaiseen dataan, koska hyökkääjä on ohjannut uhrin datan kulkemaan oman laitteensa ja verkkonsa kautta. Edellä kuvattu haavoittuvuus mahdollistaa esimerkiksi identiteettivarkauden, pääsyn esimerkiksi pankkitietoihin. Iot-verkkojen ja IoT-laitteiden tietoturvauhkat voidaan havainnollistaa kuvan 4.3 avulla [39].



Kuva 4.3: IoT-verkkojen ja IoT-laitteiden yleisimmät tietoturvauhkat.

Saatavuus- eli palvelunestohyökkäyksiä (*engl. Denial of Service, DoS*) IoT-laitteita vastaan voi tapahtua verkon eri kerroksilla. Hyökkäykset voivat kohdistus fyysiseen kerrokseen, jolloin pyritään aiheuttamaan häiriöitä tietoliikenteessä. Palvelunestohyökkäykset voivat kohdistua myös verkkokerrokseen, jolloin hyökkääjä pyrkii aiheuttamaan datapakettien törmäyksiä, joka taasen johtaa pakettien uudelleenlähetyspyyntöihin ja sitä kautta IoT laitteiden akun voimakkaaseen kulutukseen. Palvelunesto hyökkäykset voivat kohdistua myös verkkokerrokselle [39]. Tyypillisimmät menetelmät ovat pyrkiä hyödyntämään haavoittunutta solmua siten, että solmu kysyy naapurisilmuilta tilatietoja ns. hello-viestillä tai pyytämällä kuittausta lähetetystä paketista siten, että verkkokerroksen kyky siirtää dataa häiriintyy tai estyy kokonaan. Palvelunestohyökkäys, joka toteutetaan kohdennetulla haittaohjelmalla vaikuttaa usein kuljetuskerrokseen. Tällöin haittaohjelma aiheuttaa ruuhkaa dataliikenteessä ja purkaa jopa IoT-laitteiden ja langattomien sensoriverkkojen (*engl. Wireless Sensor Nodes, WSN*) synkronointeja [39].

Sensitiivistä dataa mittaavien IoT-laitteiden on käsiteltävä tietoja, jotka voidaan luokitella luottamuksellisiksi. Viestintäkanavan luottamuksellisuus voidaan näissä tapauksissa saada aikaan salausmenetelmien kautta. Nykyisin yleisesti käytössä olevat symmetriset ja epäsymmetriset salausalgoritmit tulee kuitenkin analysoida ennen niiden käyttöönottoa, jotta IoT-järjestelmän sovelluksen toiminta, IoT-laitteen suorituskyky tai kriittisen tiedon jakaminen ei vaarannu esimerkiksi liikaa laskentatehoa vaativan algoritmin vuoksi [39]. Tietojen luottamuksellisuuden ja eheyden puute voi myös vaarantaa käyttäjien yksityisyyden ja koskemattomuuden. IoT-laitteen sensoritietojen luvaton käyttö voi häiritä järjestelmän suunniteltua toimintaa ja pahimmassa tapauksessa mahdollistaa luvattoman pääsyn laitteelle ilman, että varsinainen käyttäjä edes huomaa IoT-laitteella olevaa tunkeutujaa.

Jopa maailmanlaajuinen IoT-laitteiden verkosto tuo haasteita turvallisuuteen ja yksityisyyteen. Luottamuksellisuus, eheys ja saatavuus ovat ensiarvoisen tärkeitä IoT-laitteiden välisessä tiedonvaihdossa. Näiden laitteiden älykkyys ja autonominen toiminta vaativat huomattavasti lisäpanostusta, jotta voidaan suojautua IoT-laitteiden korruptiolta, IoT-laitteiden lähettämän tiedon väärentämiseltä ja IoT-verkkoon kohdistuvien hyökkäysten vaikutukselta tietoliikenteeseen [39]. Bastos et al. [9] toteavat, että samalla kun IoT-laitteiden seurantaa ja käyttöä tulisi helpottaa, Shodan-hakukoneen [58] tyyppiset palvelut tekevät suojaamattomien IoT-laitteiden löytämisestä todella helppoa. Yhteenvetona he toteavat, että käyttäjien olisi oltava tietoisia tietoturvariskeistä ja heillä olisi oltava enemmän pääsyä työkaluihin, jot-

ka auttavat estämään, valvomaan ja vähentämään IoT-laitteisiin kohdistuvia hyökkäyksiä. Tietoturvan käyttökokemuksen tulee kuitenkin olla mahdollisimman yksinkertaista.

Nykyisin kyberrikolliset voivat murtautua mihin tahansa internettiin yhdistettyyn järjestelmään, myös langattomiin IoT-laitteisiin tai mihin tahansa IoT-järjestelmän osa-alueeseen. Perinteisesti langalliset IT-järjestelmät ovat olleet hyökkääjien kohteena, kun taas sulautetut- ja IoT-järjestelmät on koettu liian vaikeiksi hakkeroitaviksi, eikä niiden tuottama lisäarvo hyökkääjälle ole ollut tarpeeksi merkityksellinen [16]. Hyökkäysten määrä on nykyisin lisääntynyt huomattavasti, koska IoT-järjestelmiin on nykyisin lisätty Ethernet-, WLAN-, USB-, Bluetooth-, GPS- ja muita tietoliikennerajapintoja. Hyökkääjien suosiossa on suorittaa hyökkäys tietoliikenne- tai havaintorajapintaan [16, 57]. Hyökkäysten kohteena voi olla myös avoinna oleva selainpohjainen käyttöliittymä, joka voi mahdollistaa hyökkääjälle pääsyn toimintoihin tai ainakin mahdollisuuden korruptoida tietoja ja estää suorituskyky palvelunestohyökkäyksellä [16].

4.6 IoT-järjestelmän tietoturvan validointi penetraatiotestauksella

Tietoturvan validointi aloitetaan yleensä yksikkötesteillä ja edetään edelleen penetraatiotestaustasolle (*engl. pentesting*). Penetraatiotestauksella tarkoitetaan yleisesti tietojärjestelmien tai palveluiden murtotestausta, jossa suoritetaan hyökkäyksen kohteeseen suunnattua tiedonhankintaa, valitaan sopivat työkalut hyökkäyksen suorittamiseksi, simuloidaan tietoturva- tai kyberhyökkäys, dokumentoidaan hyökkäyksen eteneminen ja kirjataan havaitut tietoturvapuutteet tai havaitut haavoittuvuudet. Toteuttamalla penetraatiotestaus ”sensorista-pilveen”- periaatteella, voidaan löytää riskejä ja haavoittuvuuksia kussakin IoT-järjestelmän komponentissa ja komponenttien rajapinnoissa. Penetraatiotestaus käsittää yleensä seuraavat vaiheet [64]:

1. Evaluoinnin kohteeseen tutustuminen ja kohteen laitteiden sekä verkon kartoittaminen (*engl. footprinting*),
2. Evaluoinnin käytössä olevien järjestelmien ja sovelluksien kartoittaminen ja niiden haavoittuvuuksien tunnistaminen (*engl. reconnaissance)*,
3. IoT järjestelmiin tukeutuminen havaittuja haavoittuvuuksia hyväksikäyttäen,
4. IoT- järjestelmän oikeuksien laajentaminen verkkoliikenteen analysoimiseksi,

5. Tiedon kerääminen IoT-järjestelmästä,
6. Mahdollinen takaoven asennus, joka mahdollistaa hyökkääjälle jatkossakin pääsyn järjestelmään ja
7. Hyökkäyksen piilottaminen ja jatkaminen syvemmälle hyödyntämällä järjestelmään asennettuja haittaohjelmia.

4.7 Uhkamallinnusprosessi ja STRIDE IoT-järjestelmän testausmenetelmänä

Uhkamalli on prosessi, joka tarkistaa minkä tahansa verkkopohjaisen järjestelmän turvallisuuden, tunnistaa ongelma-alueet ja määrittää kuhunkin alueeseen liittyvän riskin. Uhkamallinnus on käyttökelpoinen IoT-järjestelmän suunnittelu- ja mallinnusvaihessa. Mallinnuksessa tunnistetaan haavoittuvuuksia ja pyritään poistamaan niitä ennen kuin yhtään koodiriviä on kirjoitettu [22]. Prosessissa on tyypillisesti kuusi vaihetta:

1. Tunnistetaan kaikki laitteen omaisuudet,
2. Visualisoidaan laitteen arkkitehtuuri,
3. Puretaan IoT-laite,
4. Tunnistetaan uhat STRIDE-mallin avulla,
5. Dokumentoidaan löydetyt tietoturvauhkat ja
6. Arvioidaan löydetyt uhkat DREAD-mallin avulla.

STRIDE on malli uhkista, jota voidaan käyttää puitteena turvallisen sovellussuunnittelun varmistamisessa. STRIDE on lyhenne, joka tarkoittaa [37]:

- **Spoofing Identity - Huijaus henkilöllisyydestä** on uhka, jossa hyökkääjä käyttää uhrin henkilöllisyyttä. Hyökkääjä ottaa esimerkiksi järjestelmänvalvojan identiteetin.
- **Tampering With Data - Tietojen peukalointi** on uhka, jossa hyökkääjä muuttaa järjestelmän tietoja. Esimerkiksi hyökkääjä muuttaa tilin saldoa tai käyttäjän tunnistautumistietoja.

- **Repudiation Threats - Kieltäminen** on uhka, jossa hyökkääjä poistaa tai muuttaa tapahtuma- tai kirjautumistietoja yrittääkseen kumota niiden koskaan tapahtuneen.
- **Information Disclosure - Tietojen paljastaminen** on uhka, jossa arkaluontoisia tietoja varastetaan ja myydään voiton saamiseksi.
- **Denial of Service - Palvelunesto** on uhka, jossa järjestelmän resurssit ylikuormitetaan. Hyökkääjä on esimerkiksi voinut saada automaattiset palvelimet kirjautumaan jatkuvasti järjestelmään ja katkaisemaan kaikki yhteydet, jotta lailliset käyttäjät eivät pääse sisään.
- **Elevation of Privileges - Oikeuksien korottamisen** on uhka, jossa järjestelmän valtuutettu tai luvaton käyttäjä voi päästä käsiksi muihin tietoihin, joita heillä ei ole valtuuksia nähdä.

Uhkien arviointimalli DREAD (Damage potential, Reproducibility, Exploitability, Affected Users, Discoverability) on menetelmä, jonka Microsoft on kehittänyt tietoturvariskin laskemiseksi [22]. Kuvassa 4.4 on DREAD-menetelmän luokat, joista jokainen arvioidaan asteikolla 0-10. Menetelmä perustuu viiden luokan arvojen keskiarvoon, jota käytetään priorisoitaessa käsiteltävät tietoturvauhkat [37].

DREAD (Asteikolla 0-10)		
Damage Potential		Kuinka paljon vahinkoa uhka aiheittaa toteutuessaan?
Reproducibility		Kuinka helposti uhka voi toistua?
Exploitability		Kuinka vaativa uhka on toteuttaa?
Affected Users		Kuinka isoa määrää käyttäjiä uhka toteutuessaan koskettaa?
Discoverability		Kuinka helposti haavoittuvuus on löydettävissä?

Kuva 4.4: DREAD tietoturvauhkien arviointimenetelmä.

Keskiarvo koostuu tietoturvariskin tai uhkan todennäköisyyden arvioinnista, uhkan vakavuuden lukuarvosta, uhkan toteutuessa arvioidaan luku, jolla kuvataan vaikutusta ja uhkan aiheuttamaa vahinkoa, uhkan löydettävyydelle annetaan myös arvo ja viimeiseksi arvioidaan uhkan toistettavuus [37].

4.8 IoT-järjestelmän verkkoliikenteen analysointi testausmenetelmänä

Internetissä laitteiden välinen kommunikaatio tapahtuu tietoliikennepakettien välityksellä. Yleisellä tasolla määriteltynä tietoliikennepaketit ovat käytettävästä protokollasta riippuen tietyn mittaisia paketteja, jotka tyypillisesti sisältävät otsikon ja kuljetettavan tietosisällön. Verkkoliikenteen analysoinnissa tarkkaillaan tietoliikennepaketteja, jolloin niistä voidaan kerätä paljon erilaista tietoa. Kiinnostuksen kohteena ovat esimerkiksi vastaanottajan ja lähettäjän IP-osoitteet, käytetty protokolla ja lähetetyn paketin tietosisältö. Sisällön avaaminen on helppoa, jos salausta ei ole käytetty. Pakettien tarkkaileminen mahdollistaa näin ollen verkkoliikenteen tarkan analysoimisen ja myöskin valvomisen verkon luvattomalta käytöltä. Tietoliikennettä ja IoT -järjestelmässä käytettyjen protokollien toimintaa voi seurata useilla erilaisissa analysointiohjelmilla. Ohjelmien avulla voi kuunnella ja kaapata tietoliikenneverkossa kulkevia paketteja. Pakettien kaappaamisella (*engl. packet capture*) voidaan analysoida tarkemmin datapakettien sisältöä. Pakettien kaappaamisella voidaan varmistua myös yksityisyyden suojaan kuuluvien tietojen suojauksesta. Pakettien tarkkaileminen on yleistä tietoliikenneverkkoihin kohdistuvissa hyökkäyksissä, mutta paketteja tarkkailemalla voidaan myös analysoida IoT-laitteisiin kohdistuvien palvelunestohyökkäysten uhkaa. Verkkoliikenteen analysointi on IoT-kuluttajalaitteissa tärkeää, koska laitteiden on tarkoitus kytkeytyä Internetverkkoon ja laitteisiin pitää pystyä muodostamaan etäyhteys. Tästä syystä tällaiset laitteet ovat haavoittuvaisia myös ulkopuolelta tuleviin laittomiin yhdistämisyrityksiin. Haavoittumisen sisältävässä laitteessa voi olla esimerkiksi Telnet-etäyhteys tai kovakoodattu käyttäjätunnus ja salasana, jolloin yhteyden muodostaminen onnistuu ilman tunnistautumista.

Tietoliikenteen seuraamiseen on saatavilla useita valmiita ohjelmia, joista osa on maksullissia, mutta esimerkiksi suosittu Wireshark [20] on ilmainen. Wireshark on avoimen lähdekoodin ohjelma ja on helposti asennattavissa yleisesti käytössä oleville käyttöjärjestelmille.

4.9 Yhteenveto IoT-tietoturvasta

Verkkokaupoissa ja muissa vastaavissa kauppapaikoissa myytäviä IoT-laitteita ei yleensä ole suunniteltu käyttäjän tietoturvallisuutta ajatellen. Esimerkit kuvaavat

tietoturvaan tai pikemminkin sen puutteeseen liittyviä riskejä, joita käyttöön otetut, tietoturvattomat laitteet aiheuttavat käyttäjälleen ja laajemmin jopa yhteiskunnalle. Yhteiskunta ja kuluttajat tulevat yhä enemmän riippuvaiseksi IoT-laitteista ja niistä palveluista, joita IoT-järjestelmät tuottavat. Tämän tiedostaen tietoturvan ja yksityisyyden suojaamisen suunnittelun pitäisi olla hyvin standardoitu eikä tietoturvaa saisi jättää käyttäjän vastuulle. Laite ja ohjelmistosuunnittelu edellyttää jatkuvaa testausta ja korjauksia olemassa oleviin laitteisiin ja ohjelmistoihin. Tietoturvan kannalta tarvitaan myös monialaista yhteistyötä, jotta kuluttajien IoT-laitteista ja IoT-järjestelmistä saadaan turvallisempia. Valitettavasti markkinoilla on toimijoita, joita ohjaa enemmänkin Time-to-Market-ajattelu (TTM). Uusia tuotteita ja palveluja pitää saada markkinoille mahdollisimman nopeaan tahtiin, jotta saavutettu markkina-asema ei ainakaan horjuisi.

5 IoT-tietoturvaoppimisympäristön kehittämistutkimus

Tässä luvussa esitellään kehittämistutkimuksen tutkimusmenetelmänä sekä tutkimusta ohjaavat tutkimuskysymykset. Lisäksi käydään lyhyesti läpi, miten menetelmää IoT oppimisympäristön kehitystyössä toteutetaan. Aliluvussa 5.1 on esitelty kehittämistutkimusta tutkimusmenetelmänä. Aliluvussa 5.2 kerrototaan lyhyesti mitä tarkoittaa teoreettinen ongelma-analyysi. Kehittämistutkimuksen tutkimuskysymykset esitellään aliluvussa 5.3 ja aliluvussa 5.4 kuvataan kehittämistutkimuksen käytännön toteutusta. Aliluvun 5.5 sisältö kattaa kehittämistutkimukseen liittyvät empiirisen ongelma-analyysin käytännön toteutuksen.

5.1 Kehittämistutkimus tutkimusmenetelmänä

Kehittämistutkimuksissa on tavoitteena kehittää opetusta ja tutkia sen avulla oppimista aidoissa oppimistilanteissa. Menetelmän avulla pyritään kehittämään todelliseen tarpeeseen kohdistettuja pienessä mittakaavassa toimivia ratkaisuja, jotka saadaan lopulta yleistettyä osaksi suuremman käyttäjäkunnan toimintaa.

Pernaan [47] mukaan laadukkaaseen ja hyvään lopputulokseen pyrittäessä tärkein työvaihe on kokonaisvaltainen ongelma-analyysi. Tämä voidaan saavuttaa monipuolisella kehittäjätiimillä ja hyödyntämällä parhaalla mahdollisella tavalla saatavalla olevaa asiantuntijuutta tai toimimalla tiiviinä kehittäjätiiminä.

Kehittämistutkimus on usein monimuotoinen tutkimusmenetelmä eikä sillä ole yksiselitteistä määritelmää. Riippumatta siitä, että kehittämistutkimusta voidaan toteuttaa useilla eri tavoilla, kehittämistutkimukseen kuuluvat sekä kehittämissyklit että seuraavat kolme vaihetta [47]:

1. Ongelma-analyysi,
2. Kehittämisprosessi ja
3. Kehittämistuotos.

Ongelma-analyysissä tehdään kehittämistutkimuksen tarveanalyysi sekä määritellään tutkimuksen tavoitteet. Kehittämisprosessissa tehdään päätökset siitä, miten kehittämistutkimus suoritetaan ja millaisella kehittämistuotoksen materiaalilla voidaan vastata tutkimukselle asetettuihin tavoitteisiin. Kehittämistuotos on tutkimuksen tekijän vastaus ja ratkaisu ongelma-analyysissä määriteltyyn tarpeeseen ja asetettuihin tavoitteisiin [47].

5.2 IoT-järjestelmän tietoturvan teoreettinen ongelma-analyysi

Teoreettinen ongelma-analyysi on kehittämistutkimuksen lähtökohta. Opetusympäristön kehittämisen tulee perustua tieteellisesti todennettuihin ja valideihin havaintoihin, jotka liittyvät IoT-järjestelmien tietoturvaan. Konkreettisten kehittämispäätösten täytyy olla perusteltavissa teoreettisen viitekehyksen avulla. Kehittämispäätökset toimivat opetusympäristön suunnittelua ohjaavina tekijöinä. Oppimistehtävien tulee perustua tieteellisesti perusteltuihin kehittämispäätöksiin. Tavoitteena on toteuttaa sellaisia oppimistehtäviä, joiden avulla opiskelijat saavat ymmärryksen oppimishetkisestä IoT-tietoturvaosaamisesta. Teoreettisen ongelma-analyysin tulosten pohjalta voidaan myös hahmottaa oppimisympäristön jatkokehitystarpeita.

Teoreettisen ongelma-analyysin tarkoituksena on selvittää myös kehittämistutkimuksen haasteet. Tässä kehittämistutkimuksessa teoreettinen ongelma-analyysi tehdään tutustumalla aikaisempiin tutkimuksiin IoT-tietoturvaan liittyvistä vaatimuksista, katsaukseen olemasssa olevista oppimisympäristöistä ja katsaukseen yleisesti tieturvaan liittyviin kursseihin.

5.3 IoT-tietoturvan tutkimuskysymykset

Tämän kehittämistutkimuksen tarkoituksena on kehittää IoT-tietoturvan opetukseen soveltuva, skaalautuva fyysinen oppimisympäristö ja samalla kehitetään itsenäisen opiskelun mahdollistava virtuaalinen oppimisympäristö. Oppimisympäristön suunnittelussa on useita erilaisia haasteita, joista suurin lienee se, että IoT kattaa nykyisellään laajan skaalan erilaisissa ympäristöissä toimivia laitteita tai koneita, joilla on kyky välittää tietoa tai vastaanottaa tietoa esimerkiksi pilvipalveluista. Tutkimuskysymykset, joiden tarkoituksena on asettaa sekä tutkimuksen tavoitteet että toimia ohjenuorana toteutettavalle oppimisympäristölle, asetettiin seuraavanlaisiksi:

1. Millainen tarve Iot tietoturvaosaamisella on tutkimushetkellä, ja millaista osaamista valmistuvilta opiskelijoilta odotetaan,
2. Mitä laitteisto- ja järjestelmävaatimuksia oppimisympäristön suunnittelussa tulee ottaa huomioon ja
3. Minkälaisia oppimistehtäviä tulee pystyä toteuttamaan, että oppimistulokset olisivat tavoitteiden mukaisia?

5.4 Kehittämistutkimuksen käytännön toteutus

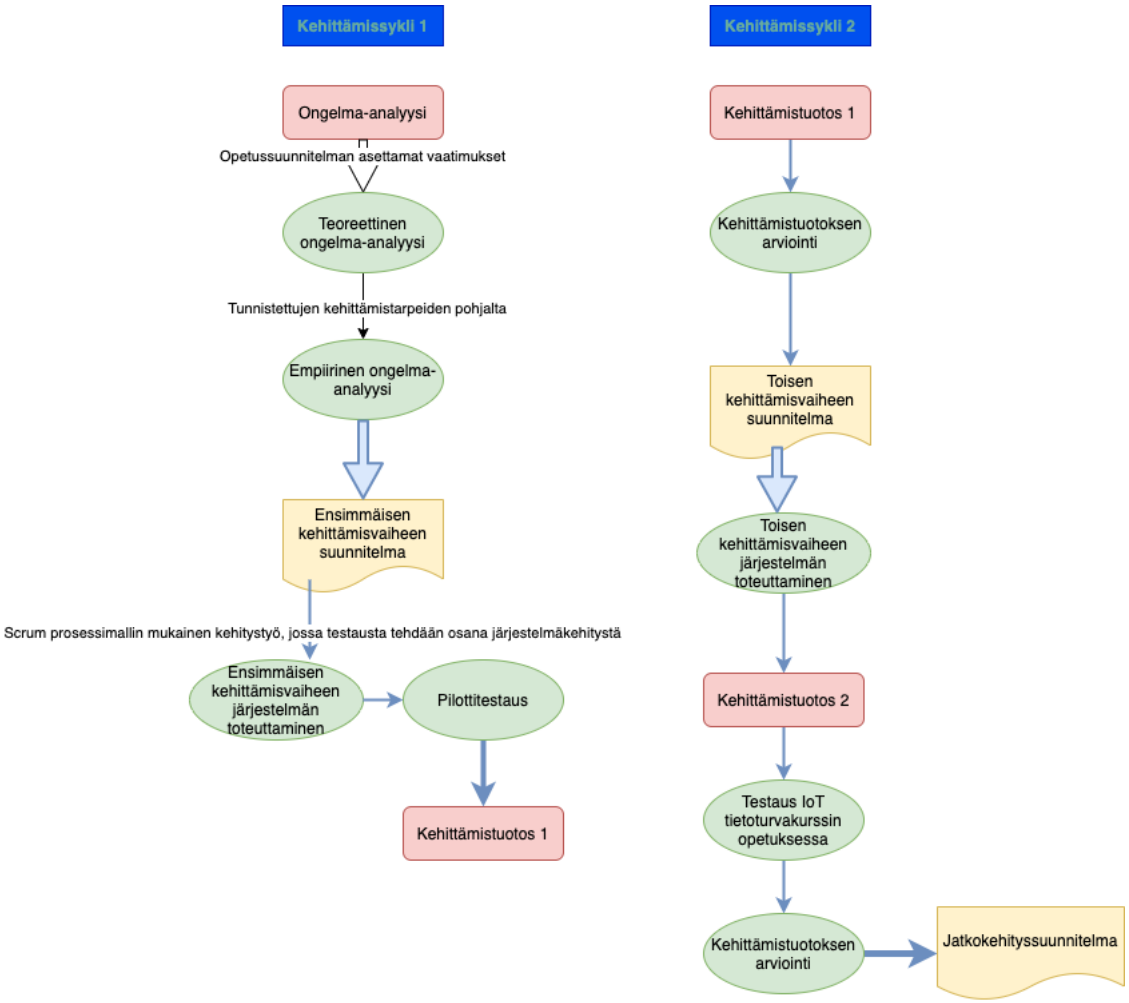
Aksela ja Pernaa [5] ovat esittäneet, että Pro gradu-työssä kehittämistutkimus voidaan toteuttaa kahdella syklillä. Tämän tutkielman kehityssyklit on esitetty kuvassa 5.1.

Kehittämissykleistä voidaan havaita, että ensimmäistä kehittämistuotosta tarkastellaan ongelma- ja teoreettisessa analyysissä esille tulleiden vaatimusten pohjalta. Lisäksi kehittämistutkimuksen ensimmäistä kehittämistuotosta testataan myös pienellä opiskelijaryhmällä ja kehittämistuotosta parannetaan esille tulleiden kehitysehdotusten pohjalta paremmin tutkimukselle asetettuja vaatimuksia vastaavaksi. Uutta kehittämistuotosta arvioidaan käytännön opetustilanteissa tehtävien huomioiden, opiskelijapalautteen ja oppimistulosten analysoinnin avulla. Seuraavan, eli kolmannen kehitysvaiheen mahdolliset parannus- ja kehitystehtävät jäävät jatkokehityshankkeessa tehtäviksi.

5.5 Empiirinen ongelma-analyysi

Oppimisympäristön ja oppimistehtävien lähtökohtana oli suunnitella mahdollisimman yrityslähtöinen oppimisympäristö, jossa pystyttäisiin toteuttamaan reaalimaailman skenaarioita ja oppimistehtäviä. Tavoitteeksi asetettiin haastattelujen perusteella se, että opiskelijat osaavat toteuttaa tieto- ja kyberturvahyökkäyksiä ja oppimansa perusteella myös opetella suojautumaan tieto- ja kyberturvahyökkäyksiltä.

Toteutetun haastattelututkimuksen mukaan opiskelijoiden tulisi ymmärtää IoT-laitteiden tietoturvavaatimusten eroavaisuudet. Silloin, kun IoT-laitteissa käsitellään sekä paikallisesti että pilvipalvelussa esimerkiksi terveysdataksi luokiteltavaa mittaustietoa, tulee suojaustason olla saman tyyppinen kuin perinteisten terveystie-



Kuva 5.1: Kehittämissuunnitelman eteneminen ja vaihetuotteet Pernaan [47] esittämän syklimallin mukaisesti

tojen suojaamisessa [38]. Tärkeintä tieto- ja kyberturvallisuuden kannalta on tieto ei yleisessä käytössä oleva teknologia [38]. Haastateltavat toivat esille sen, että suojattavia tietoja ovat esimerkiksi asiakastietojärjestelmät, tietokannat, teknologiset patentit, immateriaalioikeudet, IoT-järjestelmän kannalta kriittiset koodit ja tiedonsiirron kannalta tietoverkkojen reititystiedot ja IP-osoitteet [55].

Haastattelussa korostettiin useaan otteeseen, että opiskelijoiden tulisi ymmärtää digitaaliset varmenteiden merkitys ja mitä mahdollisuuksia ne tarjoavat tiedon suojaamiseen ja tietosuojan toteuttamiseksi. Seuraavat aihealueet tulisi olla toteutettavissa oppimisympäristössä:

1. Tietoliikennetekniikan perusteiden osaaminen.
2. Tietoverkkojen perusteiden hallinta.
3. Autentikointimenetelmien perusosaaminen.
4. Salaus ja salausalgoritmien toimintaperiaatteet.
5. Mitä tarkoittaa tiedon eheys ja miten tiedon eheys voidaan varmistaa.
6. Tietoturvahaavoittuvuuksien analysoinnin perusperiaatteiden osaaminen.
7. Ohjelmistovirheiden aiheuttamien haavoittuvuuksien tunnistaminen.
8. TCP/IP liikenteeseen liittyvät haavoittuvuudet.
9. Digitaaliset varmenteet.
10. Fyysisen tietoturvan ymmärtäminen ja sisäistäminen.

Lista on pitkä ja oppimisympäristölle asetettavat vaatimukset suorastaan haasteellisia eikä kaikkia aihealueita voida kattaa syvällisesti oppimistehtävillä. Varsinkin tietoliikennetekniikan ja tieverkkojen perusteiden osaamista varten tarvitaan erilliset kurssit, joissa edellä mainitut asiakokonaisuudet käsitellään. Haastateltavat korostivat sitä, että IoT-verkkojen ja langattomien sensoriverkkojen tietoturvan ymmärtämisen kannalta on tärkeää olla hyvä tietoverkko-osaaminen [36].

Haastattelujen ja oppimistavoitteiden analysoinnin perusteella kehitettiin yrityslähtöisiä oppimistehtäviä. Oppimistehtävillä pyrittiin opiskelijoille tarjoamaan

työkalut ja tekniikat, joita hakkerit ja tietoturva-ammattilaiset käyttävät murtautuessaan mihin tahansa tietokonejärjestelmään. Tavoitteena oli perehtyä "hakkeriajatteluun", jotta opiskelijat oppisivat ajattelemaan hakkerin tavoin ja puolustautumaan paremmin tulevia hyökkäyksiä vastaan. Empiirisen ongelma-analyysin perusteella oppimisympäristössä tehtävissä harjoituksissa päätettiin käyttää systemaattisesti eettistä hakkerointiprosessia.

Empiirisen ongelma-analyysin perusteilla opiskelijoiden tulisi oppia skannaamaan tietoverkkoja, testaamaan ja auditoimaan IoT-järjestelmän tietoturvaa, hakkeroimaan ja suojaamaan kohdejärjestelmiä. Oppimistehtävien tulisi kattaa viisi eettisen hakkeroinnin vaihetta, eli tiedustelun toteuttaminen, pääsyn saaminen, haavoittuvuuksien luetteloiminen, pääsyn ylläpitämiseen ja jäljitettävyyden piilottaminen. Myös IoT-laitteiden ohjelmakoodin tietoturvan evaluointi nousi empiirisessä tutkimuksessa merkittäväksi osaamistavoitteeksi. Empiirisen ongelma-analyysin perusteella IoT-tietoturvakurssin oppimistehtävien tulisi kattaa seuraavat aihealueet [55, 38, 36]:

1. Johdatus eettiseen hakkerointiin
2. Jalanjälki ja tiedustelu
3. Verkkojen skannaus
4. Haavoittuvuusanalyysi
5. Kohdejärjestelmän hakkerointi
6. Haittaohjelmien aiheuttamat uhkat
7. Sniffing, eli verkkoliikenteen haistelu
8. Palvelunestohyökkäys
9. Verkkopalvelimien hakkerointi
10. Verkkosovellusten hakkerointi
11. SQL-injektio
12. Langattomien verkkojen hakkerointi
13. Mobiilialustojen hakkerointi

14. IoT-hakkerointi
15. Pilvilaskennan tietoturva
16. Kryptografia

Oppimistehtäviä voidaan toteuttaa siten, että opiskelijat tekevät tiedustelua ja kohdennettuja hyökkäyksiä opetusverkossa oleviin virtuaalikoneisiin ja IoT-laitteisiin. Oppimistehtävien kuvauksessa tulee olla olennaista tietoa tehtävien suorittamiseksi. Empiirisen ongelma-analyysin perusteella oppimisympäristön tulisi koostua simuloidusta pienen yrityksen IoT-verkosta tai pienehköstä IoT-järjestelmästä, jossa on erilaisten IoT-laitteiden lisäksi sekä tiedonvälityksessä tarvittavia palvelimia että pilvipalvelun tarjoavia palvelimia. Perustason tehtävien lisäksi tulisi edistyneemmille opiskelijoille tarjota lisätehtäviä. Empiirisessä ongelma-analyysissä korostui ongelmalähtöisen oppimisen lisäksi opettajan rooli, koska lopuksi tehtävät tulisi käydä perusteellisesti läpi opettajan kanssa.

5.6 IoT-tietoturvaoppimisympäristölle asetettavat vaatimukset

Lähtökohtana on se, että oppimisympäristöllä tulee pystyä havainnollisesti ja käytännönläheisesti toteuttamaan tieto-, kyberturvallisuuteen ja tietoliikenneteknologiaan liittyviä oppimistehtäviä. Seuraavassa kootusti empiirisen ongelma-analyysin tulokset oppimisympäristölle ja oppimistehtäville asetettavista vaatimuksista.

Pedagoginen malli: Oppimistehtävien suunnittelussa on otettava huomioon pedagoginen lähestymistapa. Oppimistehtävien suunnittelussa tavoitteena on ns. Ongelmalähtöinen oppiminen (*engl. Problem-based-learning*), jossa opiskelijan reflektiolla, eli itsearvioinnilla ja vertaisarvioinnilla pyritään syventämään osaamista. Oppimistehtävien tulee sisältää käänteistä oppimista (*engl. Flipped Learning*), aivoriihiä ja tiimityötä, jossa oppilaat jakavat toisilleen osaamistaan. Oppimistehtävien rakenne tulee suunnitella siten, että tietoa ei tarjota valmiina ja oppimistehtävillä on toiminnallinen viitekehys, jossa otetaan huomioon oppimistavoitteet. Oppimistehtävien tulee olla myös arvioitavissa siten, että kurssiarvosana koostuu harjoitustehtävien ja itsenäisten oppimistehtävien numeroarvosanoista.

IoT-järjestelmän haavoittuvuuksien analysointi: Oppimistehtävissä tulee erityistä huomiota kiinnittää IoT-ekosysteemin haavoittuvuuksien analysointiin. Haavoittuvuus voi olla mikä tahansa heikkous, joka mahdollistaa hyökkäyksen toteutumi-

sen tai jota voidaan käyttää hyökkäyksen aiheuttamisessa. Haavoittuvuuksia voi olla tietojärjestelmissä, langattomissa sensoriyksiköissä, tiedon käsittelyprosesseissa ja myöskin käyttäjien toiminnassa. Oppimistehtävissä pitää olla mukana tietoturvatapahtumien havainnointia esimerkiksi tunnistamalla muutoksia tai poikkeamia (*(eng. anomalies)* sensoriverkon lähettämässä datassa tai muutoksia tietojärjestelmän toiminnassa. Muutosten ja poikkeamien havainnointiin tulee oppimistehtävissä käyttää pääasiassa ohjelmistotyökaluja.

Tietoturvaloukkaus: Tietoturvaloukkaus voidaan määritellä oikeudettomaksi puuttumiseksi tietoon tai tietojärjestelmään. Koska tietoturvaloukkaukset ovat hyvin yleisiä, niin oppimisympäristössä tulee olla käytännönläheisiä harjoituksia, joilla havainnollistetaan erilaisia tietoturvaloukkauksia. Harjoituksissa tulee olla tehtäviä, joilla havainnollistetaan käyttäjätunnusten ja salasanojen väärinkäyttöä, tietomurtoja, haittaohjelmatartuntoja, palvelunestohyökkäyksiä, tietojen varastamista ja kohdistettuja haittaohjelmahyökkäyksiä.

Fyysinen tietoturva: Haastattelututkimuksen tulosten analysoinnissa tuli myös esille, että organisaation työtekijöillä tulee olla osaaminen ja ymmärrys tietoturvan tilannekuvan seurantaan ja analysointiin. Ohjelmistokehittäjien tulee myös kyetä ehkäisemään, tunnistamaan ja analysoimaan tietoturvahäiriöitä, dokumentoidaan niitä sekä reagoimaan niihin yrityksessä olevan ohjeistuksen mukaisesti.

Eettinen hakkerointi: Oppimistilanteissa on myös korostettava sitä, että on olemassa sekä laitonta että laillista hakkerointia. Opiskelijoilla on hyvä olla ymmärrys milloin henkilö syyllistyy laittomaan hakkerointiin. Laittomaan hakkerointiin saattaa syyllistyä tietoturvatestausta tekevä henkilö, joka tunkeutuu ilman erillistä lupaa tietoverkkoon, tietojärjestelmään tai tietojärjestelmän sisältämään tietoon. Myös sellaisten ohjelmien, palvelujen tai muun resurssin käyttö ilman asianomaisen tahon lupaa täyttää laittoman hakkeroinnin tunnusmerkit. Tällöin on vaara tuhota tietojärjestelmästä tietoja tai käyttää järjestelmää omiin tarkoituksiin. Tunkeutuminen on silloin luvallista, kun yritys tai organisaatio palkkaa niin sanotun valkohattuhakkerin etsimään tietoverkostaan tai -järjestelmästään tietoturva-aukkoja tai haavoittuvuuksia.

Palvelunestohyökkäykset: Haastattelututkimuksessa ja myöskin teoreettisessa katsauksessa tuli esille erilaisten palvelunestohyökkäysten tunnistaminen ja niiden estäminen. Oppimistehtäviin tulee sisällyttää hajoituksia, joissa haavoittuneen IoT-laitteen avulla suoritetaan palvelunestohyökkäys. Palvelunestohyökkäyksellä voi-

daan esimerkiksi lamaannuttaa tietokantapalvelu suurella määrällä viestejä.

IoT-järjestelmän klusterointi: Oppimisympäristössä tulee olla palomuuri ja opiskelijoiden tulee ymmärtää palomuurin toimintaperiaate. Jos palvelunestohyökkäys tulee yhdestä IP-osoitteesta, se on helppo havaita ja torjua palomuuriin asetettavien sääntöjen ja eston avulla. Palvelunestohyökkäys on yleensä hajautettu palvelunestohyökkäys (*engl. Distributed Denial of Service attack, DDoS*), jolloin se toteutetaan yhtä aikaa useista eri lähteistä. Oppimisympäristössä tulee pystyä toteuttamaan palvelunestohyökkäys siten, että verkossa olevat koneet tai IoT-laitteet kaapataan ja hyökkäys toteutetaan niihin asennettujen bottiohjelmien avulla. Tällä tavalla saadaan konkreettisesti ymmärrys bottiverkon toiminnasta. Oppimisympäristöön toteutetaan verkkosivusto, johon ohjataan ohjelmallisesti suuri määrä palvelupyyntöjä. Palvelunestohyökkäyksessä, eli DDoS-hyökkäyksessä, hyökkääjä täyttää verkon suurella määrällä datapaketteja. Opiskelijoiden tulisi pystyä oppimisympäristössä mittamaan hyökkääjän haltuunsa ottamaa kaistanleveyttä.

IoT-järjestelmän haittaohjelmat: Kohdistetun haittaohjelmahyökkäyksen tunnistaminen on haastattelututkimuksen mukaan myöskin oleellinen osa tietoturvaosaamista. Kohdistettu haittaohjelmahyökkäys on aina monivaiheinen tietoverkkohyökkäys, joka kohdistuu tiettyyn rajattuun kohteeseen ja joka tehdään haittaohjelmien sekä muiden toimintojen avulla. Kajaanin ammattikorkeakoulusta valmistuvat Tieto- ja viestintätekniikan insinöörit työllistyvät usein yrityksiin, joiden toimialana on erilaiset valtionhallinnon, puolustusvoimien ja terveysteknologian toimialat. Tällöin opiskelijoiden on ymmärrettävä, että hyökkääjä pyrkii toimimaan niin, että hyökkäystä ei huomata ja sen jäljet poistetaan tietojärjestelmistä hyökkäyksen lähteen selvittämisen vaikeuttamiseksi. Kohdistettu haittaohjelmahyökkäys voi olla myöskin kyberoperaatio tai kyberoperaation osa. Hyökkäyksen lähteen piilottamista tulee voida havainnollistaa ottamalla opetusympäristöön myös Tor-verkko.

IoT-järjestelmän tietoturvan perusperiaatteet: IoT-järjestelmän tietoturvallisuus on tärkeä näkökohta kaikissa IoT-projekteissa, joten oppimisympäristössä tietoturvan parhaita käytäntöjä on aina hyvä soveltaa IoT-laitteiden suunnittelussa ja kehittämisessä.

Langattomien sensoriverkkojen tietoturva: Oppimisympäristön tulee sisältää myös esimerkiksi ESP32-sensoriyksiköitä. ESP32-kehitysalustalla voidaan opettaa suojatun käynnistyksen ja flash-salauksen ottamista käyttöön. Oppimistehtäväna tulee olla myös OTA (*engl. Over-The-Air*) -päivityksiä, joissa korjataan esimerkiksi tietoturva-

aukkoja olemassa olevissa IoT-laitteissa. ESP32-sensorinoodien tietoturvan toteuttamisessa voidaan myös käyttää TLS/DTLS-protokollia, joilla voidaan suojata sovellustasolla tietoliikennettä etäpalvelimien tai muiden tietokonelaitteiden kanssa. Langattomien sensoriverkkojen (*engl. Wireless Sensor Network, WSN*) tietoturvan oppimisen ja tietoturvan tärkeyden ymmärtäminen kannalta edellä kertotun tyyppiset oppimistehtävät ovat ensiarvoisen tärkeitä.

Anomalioiden tunnistaminen IoT-järjestelmän tietoliikenteestä: Anomalioiden eli poikkeamien havaitsemista käytetään laajalti eri toimialoilla. IoT-järjestelmien kyber- ja tietoturvan varmistamiseksi on kehitetty erilaisia ohjelmia ja menetelmiä [3]. Osa menetelmistä perustuu silmämääräiseen tietoliikenteen tutkimiseen. IoT-järjestelmien tuottamien aikasarjaisen datan määrä on kuitenkin tehnyt verkkoliikenteen silmämääräisestä tehtävästä haastavan. Kehittyneemmissä menetelmissä käytetään koneoppista tai jopa lohkoketjuteknologiaa hyödyntäen aikasemmin kerättyä normaalitilanteessa kerättyä dataliikennettä. Tieto- ja kyberturvan ennakoivan ylläpidon pitää pystyä suodattamaan jopa tuhansien sensorien tuottamaa dataa ja kyetä löytämään datavirrasta mahdollisia poikkeavuuksia [3]. Järjestelmän suojaamiseksi poikkeamien havaitsemista pidetään tärkeänä työkaluna, koska se auttaa tunnistamaan järjestelmän epänormaalit toiminnot [3].

6 Ensimmäinen kehittämissykli

Ensimmäisessä kehittämissyklissä teoreettisen ja empiirisen ongelma-analyysin pohjalta suunniteltiin ja rakennettiin sellainen IoT-tietoturvan opetukseen soveltuva oppimisympäristö, joka vastasi asetettuihin tutkimuskysymyksiin ja samalla mahdollisti Kajaanin ammattikorkealussa käytettävän oppimiskäsityksen mukaisen opetuksen. Aliluvussa 6.1 esitellään lyhyesti kehittämissyklin aikataulu ja tavoitteet. Seuraavassa aliluvussa 6.2 kuvataan ensimmäisen kehittämissyklin tuloksena syntynyt oppimisympäristö. Kehittämissyklissä suoritettiin myös pilottitestaus, jonka toteutustapa ja tulokset on esitelty aliluvussa 6.3. Aliluvussa esitellään myös saatujen tulosten perusteella päätetyt kehitystarpeet, joiden pohjalta lähdettiin jatkokehittämään toisessa kehittämissyklissä sekä oppimistehtäviä että oppimisympäristöä.

6.1 Ensimmäinen kehittämistuotos

Oppimisympäristön ja harjoitustehtävien suunnittelu aloitettiin keväällä 2021. Tavoitteeksi asetettiin se, että oppimisympäristössä pystytään opettamaan ja demonstroimaan sekä Tietoverkkoteknologian että IoT-tietoturvakursseihin liittyviä itsenäisesti tehtäviä harjoituksia ja opiskelijatiimissä tehtäviä syventäviä harjoituksia. Oppimisen kannalta on tärkeää, että oppimistehtävissä on oltava perinteisen tietoverkkojen tietoturvan ja IoT-laitteita koskevan tietoturvan lisäksi myös fyysisen tietoturvan elementtejä. Digitalisaatio, pilvipalvelut ja IoT ovat nykyään käytössä tavalla tai toisella lähes kaikissa yrityksissä ja organisaatioissa. Digitalisaation ja pilvipalveluiden lisääntyessä myös tietoverkkoyhteyksien ja käyttäjien määrä kasvaa. Lisääntyvä digitalisaatio, pilvipalveluiden käyttö ja Internettiin yhteydessä olevien laitteiden lukumäärän kasvu aiheuttavat myös lisääntyviä riskejä yrityksen tai organisaation tietoturvallisuudelle.

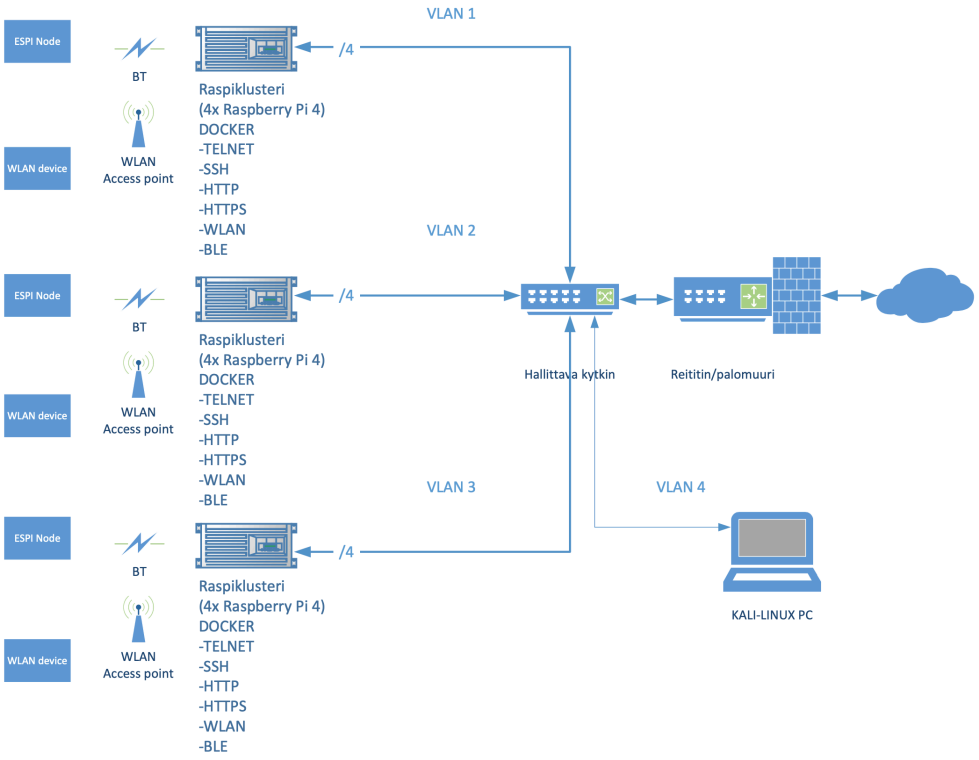
6.2 Fyysinen oppimisympäristö

IoT-tietoturvakurssin sisällön ja oppimistehtävien yhteydessä suunniteltiin myös fyysistä IoT-tietoturvaoppimisympäristöä. Kuvassa 6.1 on graafinen kuvaus oppimisympäristöstä. Suunnitelman pohjana oli teoreettisen ja empiirisen tutkimuksen pohjalta luodut skenaariot oppimistehtävistä. Päätimme rakentaa julkisesta Internetistä helposti erotettavan oppimisympäristön, jossa on sekä IoT-laitteita että verkon toimilaitteita. Hallittavan kytkimen kautta mahdollistetaan pääsy reitittimelle, joka samalla toimii palomuurina. Hallittavaan kytkimeen liitettiin Ethernet-yhteydellä kolme kappaletta Raspberry Pi -klustereita. Jokaisessa klusterissa on Docker-container (Docker-kontti) ja Docker-konteihin asennettiin Telnet-, SSH-, HTTP-, HTTPS-, WLAN ja BLE protokollaa käyttäviä palveluita. Raspberry Pi-klusterien Docker-konteissa ajetaan MariaDB-tietokantapalvelua, aikasarjaista InfluxDB-palvelua, MQTT Broker-palvelua ja sensoridatan esittämiseen käytettävää Grafana-palvelua. Langattomista sensorinoodeista tulee sensoridataa Bluetooth- tai WLAN-yhteyttä käyttäen Raspberry Pi-klusterissa oleville palveluille. Kuvan 6.1 kaltaisen ratkaisun toteutti Niilo Niskanen Kajaanin ammattikorkeakoululle tekemässään opinnäytetyössä [43]. Alkuperäisen suunnitelman mukaan tällaisella oppimisympäristöllä pystyttäisiin toteuttamaan IoT-järjestelmän mallintaminen ja toteuttamaan kurssikuvauksen edellyttämät oppimistavoitteet.

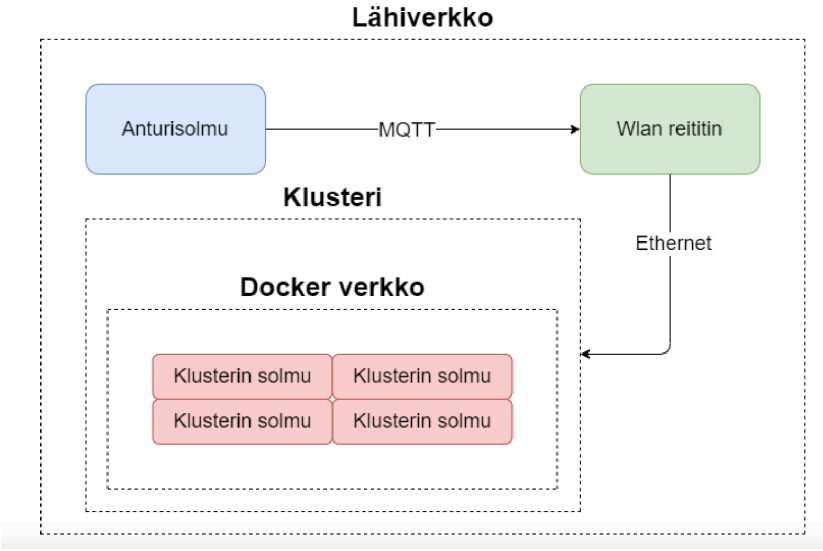
Kuvassa 6.2 on kuvattu Raspberry Pi -klusterin ja sensoriverkon toiminta-ajatus. Langattomien sensorinoodien mittausdata välitetään MQTT-protokollaa käyttäen InfluxDB-tietokantaan ja esitetään Grafana-näkymässä. Raspberry Pi -klusterin solmut tulee liittää samaan lähiverkkoon, jotta ne voivat kommunikoida keskenään. Raspberry Pi -klusterit luotiin ohjelmallisesti, koska laitteet eivät voineet muuten jakaa resursseja keskenään. Klusterien yhteisen massamuistin lisäksi kullakin klusterilla oli oma massamuisti [43].

6.3 Kehittämistuotoksen pilotointi

Alustavan suunnitelman mukaisen kehittämistuotoksen pilotoinnin suorittivat työharjoittelussa olevat opiskelijat. Opiskelijat suorittivat opintoihinsa kuuluvan työharjoittelun Kajaanin ammattikorkeakoulun Tekniikan yksikössä. Työharjoittelun kesto oli viisi (5) kuukautta. Pilotoinnin aikana opiskelijat testasivat sekä oppimisympäristöä että suunniteltuja oppimistehtäviä. Samalla opiskelijat kirjasivat ha-



Kuva 6.1: IoT-oppimisympäristön graafinen kuvaus opinnäytetyön jälkeen.



Kuva 6.2: IoT-oppimisympäristön Raspberry Pi -klusteri. [43]

vaintojaan oppimistehtävien suorittamisesta. Opiskelijoiden suorittaman testauksen ansiosta oppimisympäristöstä löytyi runsaasti virheitä ja puutteita, joiden korjaamiseen opiskelijat esittivät kehittämisehdotuksia. Löydetyt puutteet olivat niin oleellisia, että oppimisympäristö piti suunnitella osittain uudelleen vastaamaan paremmin asetettuja tavoitteita. Luettelo merkittävimmistä puutteita tai muutettavista teknisistä toteutuksista:

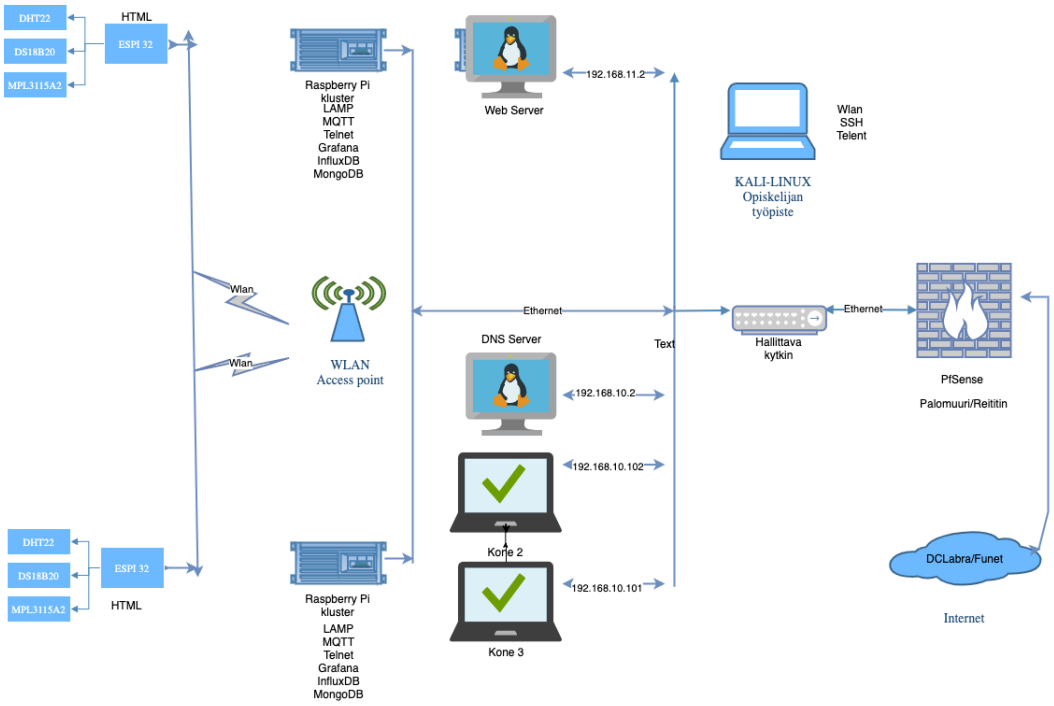
1. Docker-kontit vaikeuttivat IoT-ekosysteemin toiminnan analysointia,
2. Haavoittuvuuksien analysointi Docker-konttien sisällä ajettattaville palveluille vaatii erityisosaamista,
3. Oppimisympäristöstä puuttui fyysisiä palvelimia, esimerkiksi Web-serveri ja virtuaalinen palomuuri,
4. Oppimisympäristösta puuttuivat IoT-tietoverkkojen opettamiseen tarvittavat verkon toimilaitteet ja palvelimet,
5. Oppimisympäristön sensorisolmut eivät mahdollistaneet laaja-alaista haavoittuvuuksien evaluointia eikä ohjelmakoodin analysointia,
6. Oppimisympäristö oli liian suppea botti-verkkohyökkäysten mallintamiseen ja
7. Oppimistehtävien itsenäiseen suorittamiseen tarvitaan fyysisen oppimisympäristön lisäksi virtuaalinen oppimisympäristö.

Kuvassa 6.3 esitellään fyysinen oppimisympäristö pilottitestauksessa havaittujen puutteiden korjaamisen jälkeen. Alustavan suunnitelman mukaiseen oppimisympäristöön lisättiin fyysisiä palvelimia ja fyysisiä tietokoneita. Tietoverkkoratkaisun selkeyttämiseksi Pfsense-palomuurilla erotetaan oppimisympäristö tarvittaessa omaksi sisäverkoksi, joka voidaan tarvittaessa irrottaa julkisesta Internetistä. Sisäverkko jaettiin myös kahteen erilliseen aliverkkoon. Seuraavassa listassa on kuvattu oppimisympäristöön lisätyt fyysiset tietokoneet:

- PfSense palomuuri-reititin,
- Web-palvelin (LAMP), Python ja Node.js,
- Dns-palvelin, nimipalvelun toteuttamiseksi,

- Kone 2, ilman admin oikeutta ja
- Kone 3, admin oikeudella

Raspberry Pi -klustereista poistettiin Docker-kontainerit. Muutoksen jälkeen palveluita suoritetaan Raspberry Pi:ssä itsenäisinä prosesseina. Docker -kontainerien poiston yhteydessä Raspberry Pi -klustereihin asennettiin itsenäisinä palvelinprosesseina LAMP (Linux, Apache, MySQL ja PHP), MQTT Broker, Telnet, Grafana, InfluxDB ja MongoDB. Raspberry Pi:t kytkettiin Ethernet-kaapeleilla hallittavaan kytkimeen. ESP32-pohjaiset sensorinoodit kommunikoivat edelleen langattomasti Raspberry Pi -klusterien kanssa.



Kuva 6.3: IoT-oppimisympäristön graafinen kuvaus ensimmäisen kehittämissyklin jälkeen.

Seuraavaksi tässä luvussa kuvataan tarkemmin oppimistehtävien kokonaisuutta, joista yksittäiset pilottitestauksessa suoritetut oppimistehtävät koostuivat. Oppimisympäristön pilotoinnissa seurattiin myös oppimistehtävien onnistumista ja oppimistehtävillä saavutettavaa osaamista. Oppimistehtävien suunnittelussa hyödynnettiin sekä teoreettista että empiiristä analyysia. Luettelossa on kuvattu ensimmäi-

seksi oppimistehtävään liittyvä hyökkäystyyppi, tavoitteet ja toteutustapa oppimistehtävässä. Seuraavaksi kuvataan pilottitestauksessa saavutettu tulos.

Strukturoimattomat hyökkäykset: Nämä ovat niitä hyökkäyksiä, joissa hyökkääjällä ei ole aiempaa tietoa kohdeympäristöstä, johon hän hyökkää. Useimmiten tällaisissa hyökkäyksissä he luottavat kaikkiin vapaasti saatavilla oleviin työkaluihin. Strukturoimattomia hyökkäyksiä kohdistetaan usein ennalta haavoittuviksi tiedettyihin järjestelmiin, ohjelmiin tai IoT-laitteisiin, joiden IP-osoite on mahdollisesti ja haavoittuvien laitteiden listalla. Tämän hyökkäystyypin opettaminen on perusteltua siksi, että hyökkäyksen estämisen opetteluun voidaan liittää oppimista myös yleiseen haavoittuvuuteen ja käytettävissä olevaan ennakkotiedon hyväksikäyttöön.

Tulos: Oppimisympäristöllä pystyttiin suorittamaan kohdeympäristön tietojen keruu, verkkoliikenteen seuranta ja IP-osoitteiden kartoitus. Samoin oppimistehtävillä pystyttiin kartoittamaan haavoittuviksi oletetut IoT-laitteet.

Kehityskohteet: Langattomien sensoriyksiköiden osalta ohjelmakoodia pitää tehdä monipuolisemmaksi lisäämällä Web-selainnäkymän sisältäviä ESP32-sensoriyksikköjä ja Python-koodilla koodattuja langattomia sensoriyksikköjä.

Strukturoidut hyökkäykset: Strukturoidussa hyökkäyksessä hyökkäykseen on valmistauduttu ennakkoon ja hyökkääjällä on parempi suunnitelma hyökkäyksen suorittamiseen. Tätä hyökkäystyyppiä edeltää usein liikenteen tiedustelu ja mahdollisesti myös porttiskannaus. Näin opiskelijoille voidaan opettaa tietoturvakäytänteiden noudattaminen, jotta mahdollisesti hyökkääjät eivät saa haltuunsa varsinaisessa hyökkäyksessä hyödynnettävää tietoa hyökkäyksen kohteena olevista IoT-järjestelmistä ja sovelluksista. Hyökkäyksellä pyrittiin löytämään haavoittuvuuksia IP-kameroista.

Tulos: Oppimisympäristöllä pystyttiin suorittamaan kohdeympäristön IoT-laitteiden datapakettien seuranta Wiresharkilla, porttiskannaus ja Kali Linuxilla toteutettu hyökkäys. Myös IoT-laitteen salasanan ja Wlan-reitittimen salasanan murtaminen onnistuivat. Sen sijaan langattomien sensoriyksiköiden ja IP-kameran Fuzz-testaus ei onnistunut.

Kehityskohteet: Langattomien sensoriyksiköiden koodin Fuzz-testausta varten tarvitaan yksinkertaistettua ohjelmakoodia ja tarkennusta oppimistehtävän kuvaukseen.

Tietojen kalastelu: Tietojen kalastelu (*eng. phishing, spear phishing*) on kohdistettu

yleensä ihmiseen, joka on yksi heikoimmista lenkeistä tietoturvan kannalta. Tässä hyökkäyksessä tietojärjestelmän, IoT-laitteen tai ohjelman käyttäjää hyödynnetään eri tavoin. Usein nämä hyökkäykset onnistuvat tiedon puutteen tai tietämättömyyden vuoksi. Tietoja poimitaan käyttäjältä huijaamalla häntä tavalla tai toisella. Yleisin tapa on phishing eli kalastelu. Haavoittuviksi tiedetyt IoT-laitteet tarjoavat hyökkääjille pahimmillaan suoran pääsyn käyttäjän tietokoneelle tai mobiilipäätelaitteelle. Käyttäjät joutuvat hyökkääjien saaliiksi ja jakavat tietämättään arkaluontoisia tietoja. Tämän hyökkäystyypin opettaminen eri menetelmiä ja ohjelmia käyttäen on erittäin perustelua, koska opiskelijoilla on oltava ymmärrys esimerkiksi haavoittuvien IoT-laitteiden muodostamasta tietoturvauhkasta. Tämän hyökkäystyypin opettamiseen on suunniteltu käytettäväksi sosiaalista mediaa ja haavoittuvia IP-kameroita.

Tulos: Oppimisympäristöllä pystyttiin suorittamaan IP-kameran kaappaus onnistuneesti. Sen sijaan tietojenkalasteluhyökkäys ei onnistunut suunnitellusti. Haittaohjelman asennus onnistui samassa IP-osoiteavaruudessa oleville IoT-laitteille.

Kehityskohteet: Tietojenkalasteluhyökkäyksen toteuttamisen edellytys on erillisen IoT-laitteen lisäämisen oppimisympäristöön. IoT-laite voisi olla esimerkiksi Raspberry Pi, joka kommunikoi WLAN- tai BLE-yhteyden kautta sensorinoodien kanssa. Raspberry Pi:ssä tulisi myös suorittaa useampia prosesseja ja laskentaa. Raspberry Pi:ssä tulisi olla käyttäjän autentikointi.

Salakuuntelu: Tämä hyökkäys voidaan suorittaa hankkimalla luvaton pääsy verkkoon ja kuuntelemalla verkon viestintää. Yleensä kaikki salaamaton liikenne voi olla helposti hyökkääjän kohteena. Tämän hyökkäystyypin opettaminen on erittäin opettavainen kokemus ja lisää opiskelijoiden ymmärrystä sekä salasanakäytännöistä ja salausmenetelmistä ja niiden tarpeellisuudesta. Tätä hyökkäystyyppiä varten käytetään erilaisia salasanan murtoon suunniteltuja ohjelmia, verkon skannausta ja Wireshar-ohjelmaa, jolla voidaan seurata ja lukea verkossa tapahtuvaa liikennettä. Oppimisympäristöön IoT-järjestelmään suunniteltiin ja toteutettiin tietoliikennettä, josta osa on vahvasti salattua ja osa täysin salaamatonta. Täysin salaamaton liikenne on toteutettu Telnet-protokollaa käyttäen.

Tulos: Oppimistehtävien suoritus onnistui, ja pääsy IoT-verkkoon oli hyvinkin opettavainen kokemus. Oppimistehtävän konkreettisin opetus oli salasanojen murtamisen kautta saatu ymmärrys vahvan salauksen merkityksestä tietoturvan kannalta. Samoin salaamattoman liikenteen seurannan helppous oli erityisen hyvä tietotur-

van parantamisen kannalta.

Kehityskohteet: Wiresharkin käyttöä tulee opettaa Tietoverkot-kurssilla. Oppimisympäristöön voisi myös rakentaa useampia IoT-laitteita, joiden liikennettä voidaan analysoida ja opetella salausteknologioita.

Palvelunestohyökkäys ja Botnet-verkot: Palvelunestohyökkäys (*engl. Denial of Service, DoS*) tai Hajautettu palvelunestohyökkäys (*engl.Distributed Denial of Service, DDoS*) ovat yksi vanhimmista verkkopohjaisten hyökkäysten muodoista, jossa hyökkääjä yrittää tukkia järjestelmän, IoT-laitteen tai verkkopalvelun lähettämällä niin paljon dataa tai palvelupyyntöjä, että se ylittää rajan, jonka sovellus tai laite pystyy käsittelemään. Toisaalta hajautettu palvelunesto (DDoS) käynnistetään useista lähteistä yhteen uhrisovellukseen tai -järjestelmään erittäin suuressa mittakaavassa. Palvelunestohyökkäys on erittäin ajankohtainen nykyään, koska jopa valtiolliset toimijat kohdistavat kyberhyökkäyksiä tekemällä ne palvelunestohyökkäyksinä. Hyökkäyksillä voidaan lamaannuttaa esimerkiksi logistiikka, sähkönsiirto, pankkien tai eri virastojen palvelut. Oppimistehtävät suunniteltiin siten, että opiskelijat toteuttavat verkkohyökkäyksen käyttämällä IoT-laitteisiin asennettua Botnet-ohjelmaa. Verkkohyökkäys toteutetaan kohdistamalla useista eri IoT-laitteista tulevat palvelupyynnöt oppimisympäristössä olevalle Web-palvelimelle. Samalla analysoidaan Web-palvelimen vasteaikaa palvelupyyntöihin. Oppimistehtävän jatkuu siten, että opiskelijoiden tulee suunnitella IoT-laitteet siten, että vastaavaa Botnet-verkkoa ei ole ainakaan helposti mahdollista toteuttaa.

Tulos: Oppimistehtävien suoritus onnistui. Oppimisympäristössä pystyttiin suorittamaan palvelunestohyökkäys asentamalla Botnet-ohjelma, jonka palvelupyynnöillä estettiin yhden ESP32-sensorinoodin liikenne. Kaapattuun ESP32-sensorinoodiin asennetulla Botnet-scriptillä lamaannutettiin Web-palvelin.

Kehityskohteet: Wiresharkin käyttöä tulee opettaa Tietoverkot-kurssilla. Oppimisympäristöön voisi myös rakentaa useampia IoT-laitteita. IoT-laitteiden avulla voitaisiin jatkossa opetella suojautumiskäytänteitä Botnet-hyökkäyksiltä. Oppimisympäristöön tulee myös asentaa toinen IP-kamera, jossa on oletussalasana ja käyttäjätunnus. Oppimistehtäviä tulee laajentaa, jotta opiskelijat oppivat tunnistamaan mahdolliset riskit, joita hyödyntämällä hyökkääjä voi asentaa haittaohjelman IoT-laitteeseen. Oppimistehtävien tulisi liittyä suurimmaksi osaksi ohjelmistojen tietoturvan analysointiin. Varsinkin kuluttajatuotteiden ohjelmistojen suunnittelussa mahdolliset riskit tulisi tunnistaa, koska hyökkääjä voi käyttää Bottiverkkoja esimerkik-

si palvelunestohyökkäyksiin. Usein Internetissä olevien IoT-laitteiden tietoturvasta huolehtiminen jää kuluttaja-asiakkaan vastuulle.

Man-in-the-middle-hyökkäys (MITM): Man-in-the-Middle-hyökkäyksessä hyökkäyksen kohteen liikenne tai liikenne ylipäätään tietoverkossa kaapataan manipuloimalla palvelimen ja asiakkaan välistä viestintää ja toimimalla esimerkiksi välityspalvelimena. Tällaiset hyökkäykset tapahtuvat usein uhrin tietämättä. Tämä oppimistehtävä on suunniteltu toteutettavaksi ensin opiskelijan omalla koneella olevassa virtuaaliympäristössä, ja opittua soveltaen oppimisympäristössä kaappaamalla Web-serverin kautta kulkevat chat-viestit opiskelijatiimin Kali Linux-koneen kautta.

Tulos: Oppimistehtävien suoritus onnistui siltä osin, että IoT-verkon liikenne onnistuttiin ohjaaman hyökkääjän määrittämän koneen kautta. Virtuaaliympäristön käyttö tämän tyyppisissä oppimistehtävissä on erityisen onnistunut ratkaisu.

Kehityskohteet: Wiresharkin käyttöä tulee opettaa Tietoverkot-kurssilla. Oppimisympäristöön pitää rakentaa lisää IoT-laitteita, jotta oppimistehtäviin saadaan useampia mahdollisia toteutustapoja ja opiskelijat pystyvät suorittamaan myös haasteellisempia oppimistehtäviä.

Haittaohjelmat: Haittaohjelmat on tarkoituksella suunniteltu aiheuttamaan vahinkoa tai saavuttamaan mikä tahansa muu haitallinen tarkoitus. Oppimistehtävien avulla opiskelijat saavat käsityksen siitä, mihin toimintaperiaatteisiin Madot (*engl. Worms*) ja Troijalaiset (*engl. Trojan horses*) perustuvat ja miksi niillä on kyky levitä tietokoneelta tietokoneelle tai miksi niillä on kyky kopioida itseään. Haittaohjelmat ovat sikäli vaarallisia, että niillä voidaan aiheuttaa tietovarkauksia, tietokonejärjestelmien massatuhoa, häiriöitä verkkotoiminnassa ja käyttää myös yritysvakoilussa.

Tulos: Oppimistehtävien suoritus onnistui ja erilaisten haittaohjelmien asennus oppimisympäristöön voitiin toteuttaa.

Kehityskohteet: Haittaohjelmien tunnistamiseen tarvitaan erilaisia oppimistehtäviä.

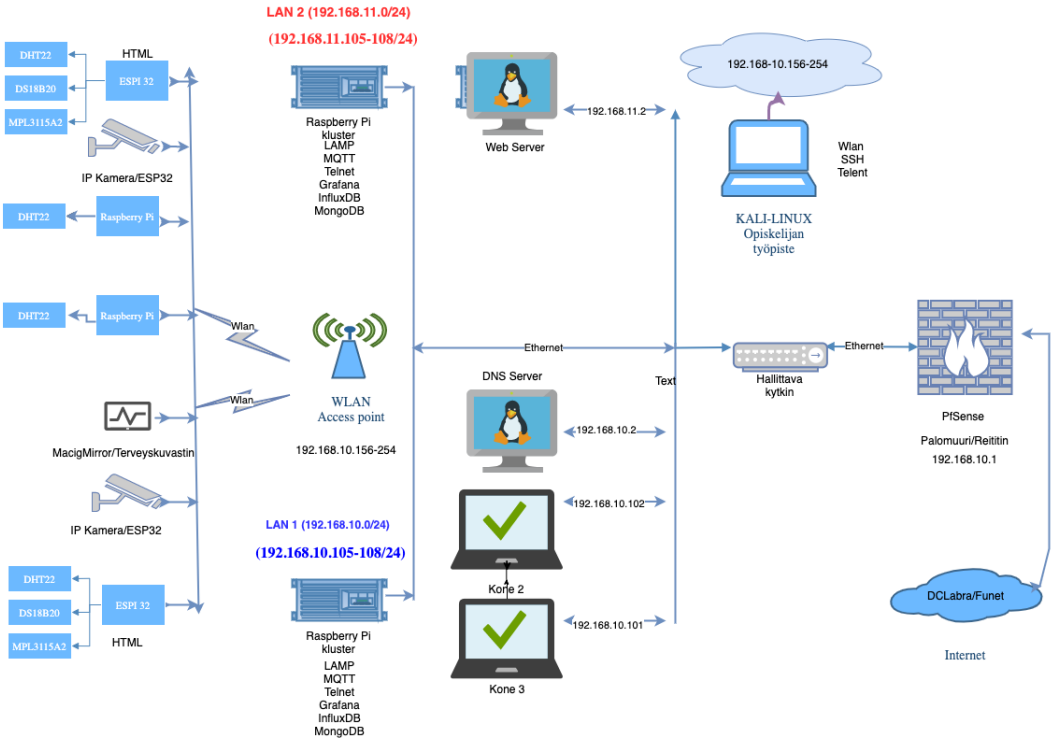
7 Toinen kehittämissykli

Kehittämistuotoksen arvioinnin ja pilottitestauksen pohjalta IoT-oppimisympäristöä jatkokehitettiin vastaamaan paremmin asetettuja tutkimustavoitteita ja samalla korjattiin oppimisympäristössä havaittuja puutteita. Pilottitestauksen perusteella saatiin myös selville se, että opiskelijoilla tulee olla merkittävästi enemmän tietoverkkoihin liittyvää osaamista. Pilottitestauksen perusteella päätettiin myös toteuttaa oppimistehtävät sekä fyysisessä että virtuaalisessa oppimisympäristössä. Näiden havaintojen perusteella käynnistettiin toinen kehittämissykli. Aliluvussa 7.1 kuvataan toisen kehittämissyklin kehittämistulos. Graafisen kuvauksen lisäksi aliluvussa perustellaan tehtyjä ratkaisuja. Seuraavassa aliluvussa 7.2 kerrotaan oppimistehtäviin liittyvästä eettisen hakkeroinnin etiikasta. Aliluvussa 7.3 kuvataan oppimisympäristön testausta, joka toteutettiin kahden eri opiskelijaryhmän IoT-tietoturvakurssin toteutuksen yhteydessä. Aliluvun 7.4 teemana on esitellä penetraatiotestaus testaus- ja oppimismenetelmänä. Oppimistehtäviin liittyvä kyber- tai tietoturvahyökkäysketjun eri vaiheet, oppimistehtävillä saadut oppimistulokset ja mahdolliset kehityskohteet on kuvattu aliluvussa 7.5.

7.1 Fyysinen oppimisympäristö

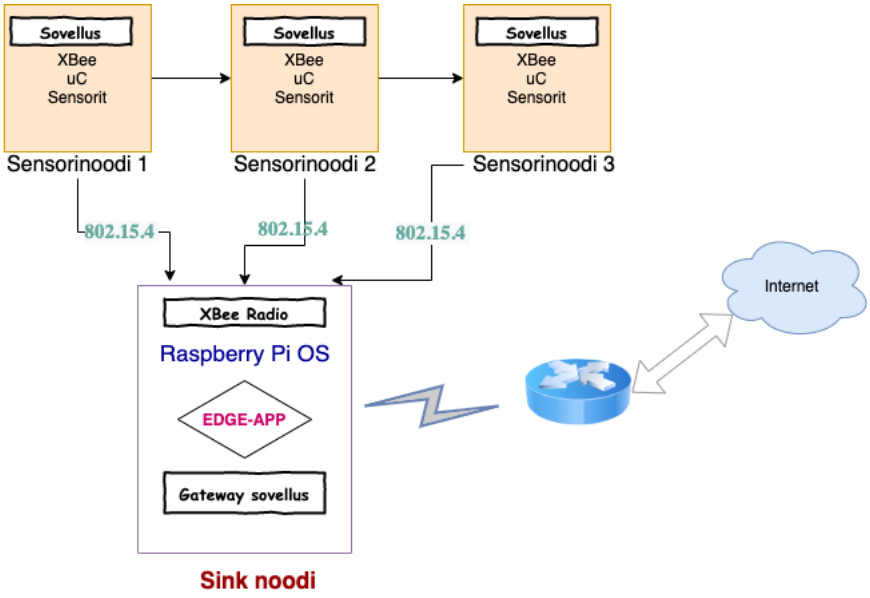
Toisen kehittämistuotoksen fyysinen oppimisympäristö on kuvattu kuvassa 7.1. Kuvasta nähdään, että oppimisympäristöön on lisätty IoT-laitteita ja selkeytetty IP-osoiteavaruuksia.

Opiskelijoiden tietoturvatestaukseen ja -auditointiin käyttämät Kali Linux-koneet on liitetty langattomasti oppimisympäristön sisäverkkoon. PfSense-reititin–palomuuriin luotiin säännöt, joilla liikenne kahden eri IP-osoiteavaruuden välillä on mahdollista. Raspberry Pi-klusterit jaettiin myös eri IP-osoiteavaruuksiin. Kaikki langattomat sensorinodet ja IoT-laitteet saavat IP-osoitteen WLAN-reitittimeltä. Oppimisympäristöön lisättiin kaksi ESP32-pohjaista IP-kameraa. Toinen IP-kamera jakaa kameran kuvaa Web-sivulla. Toinen IP-kamera lähettää videokuvaa Web-palvelimella olevaan pilvipalveluun. Oppimistehtäväksi suunniteltiin myös IP-kameran kaappaaminen, haittaohjelman ja takaportin asentaminen IP-kameraan.



Kuva 7.1: IoT-oppimisympäristön graafinen kuvaus toisen kehittämissyklin jälkeen.

Oppimisympäristöön lisättiin myös Arduino-kehitysalustalla toteutettuja XBee-sensorinoodeja ja Raspberry Pi -pohjainen Sink-noodi. Sink-noodi toimii Gateway-noodina ja välittää Arduino-noodien datan pilvipalveluun. Arduino on laajalti käytetty avoimen lähdekoodin kehitysalusta, joka perustuu Atmega328-mikrokontrolleriin ja sen kellotaajuus on 16 MHz [7]. Mesh-verkon langattomn tietoliikenteen toteuttamiseen käytettiin XBee-moduulia, joka toimii 2,4 GHz ISM-kaistalla. XBee-sensorinoodit käyttävät 802.15.4 radiota sensoridatan välitykseen Sink-noodille. Sensorinoodin XBee-moduuli kapseloi 802.15.4 protokolla- ja ZigBee-protokollapinot, ja se voidaan helposti integroida sekä Arduino- että Raspberry Pi-kehitysalustoille UART-sarjaliikenteen kautta. Kuvassa 7.2 on kuvattu Arduino-sensorinoodit ja Sink-noodi. XBee-moduuli voidaan konfiguroida kolmentyyppisiksi laitteiksi: koordinaattori, reititin ja päätelaite. Koordinaattorilla on kyky hallita koko verkkoa, joten Sink-noodi toimii myös Mesh-verkon koordinaattorina. XBee mesh-verkossa koordinaattorinoodi voi tiedustella sensorinoodien tilatietoja joko multicast- tai unicast-viestinnän avulla.



Kuva 7.2: Arduino/XBee-sensorinoodit ja Sink-noodi

Tässä kehittämissyklissä päätimme lisätä fyysiseen oppimisympäristöön Kajaanin ammattikorkeakoululla kehitetyn Terveyskuvastimen. Terveyskuvastin on eräänlainen Macig Mirror -tyyppinen järjestelmä, jota pystytään käyttämään terveysparametrien itsemittaukseen. Mitattavia parametrejä ovat verenpaine, paino, happisa-

turaatio ja verensokeri. Lisäksi laitteessa on sensoreita, jotka seuraavat huonetilan lämpötilaa, hiilidioksidipitoisuutta, valaistusta ja ilmankosteutta. Varsinkin terveydenhuollon asiakkaat voivat olla alttiimpia hyökkäyksille esimerkiksi heihin kytkettyihin lääkinnällisiin laitteisiin. Koska kohdelaite sisältää sensitiivistä terveysdataa suunnittelimme oppimistehtäksi SQL-injektion toteuttamisen ja SQL-injektion estämisen.

7.2 Hakkeroinnin etiikka ja oppimistehtävät

Toisen kehityssyklin oppimistehtävien suorituksen yhteydessä nousi esille myös hakkeroinnin etiikka. Hakkeroinnin opettelu ja opettaminen oppilaitosympäristössä voi herättää eettisiä kysymyksiä. Pashel [46] käsittelee aihetta artikkelissaan, jossa korostetaan, että yritykset käyttävät eettistä hakkerointia tietoturvansa toteuttamisessa. Tästä syystä on tullut tarve kouluttaa itseoppineiden hakkereiden lisäksi myös hakkereita esimerkiksi korkeakouluissa. Opittuja taitoja voidaan käyttää väärin, joten on ensiarvoisen tärkeää, että hakkerointia opetettaessa opiskelijoille kerrotaan selkeästi lailliset ja eettiset näkökulmat [46].

IoT-tietoturvakurssin verkko-oppimisympäristössä olevan opetusmateriaalissa on sekä vastuuvapauslauseke että ote Tieto- ja viestintärikoksista. [1] Lisäksi Cisco Akatemian opetusmateriaalissa käydään läpi eettisen hakkeroinnin perusteet ja käsitellään erilaiset hakkeroijat [13]. Valkohattuhakkerit ovat eettisiä hakkereita, jotka kartoittavat kohdeyrityksen luvalla, joko toimeksiantoina tai yritykseen palkattuna, kohteen haavoittuvuuksia ja raportoivat niistä yritykselle. Oppimistehtävissä on myös huomioitu se, että opiskelijat oppivat analysoimaan löytämiään haavoittuvuuksia ja oppivat löytämään niihin parannusehdotukset, joilla haavoittuvuudet voidaan korjata. Kaikki oppimisympäristössä tehtävät harjoitukset pohjautuvat valkohattuhakkerointiin, ja harjoitukset suunniteltu siten, että siirrytään yksinkertaisista harjoituksista kohti vaativimpia harjoituksia.

7.3 Toisen kehittämistuotoksen testaus IoT-tietoturvakurssilla

Toisessa kehittämissyklissä oppimisympäristöä testattiin kahdella eri IoT-tietoturvakurssilla. Kurssin toteutukseen sisältyi teorialuentojen lisäksi käytännön harjoituksia ja itsenäisiä oppimistehtäviä. Oppimistulosten analysoinnin lisäksi tässä ke-

[1] Rikoslaki (39/1889) 38 luku

hittämissyklissä pyrittiin analysoimaan oppimisympäristön sopivuutta erityyppisten oppimistehtävien suorittamiseen Kali Linuxilla. Kehittämissyklissä arvioitiin lisäksi sitä, kuinka oppimisympäristö vastaa didaktista ja pedagogista mallia.

IoT-tietoturvakurssilla päätettiin käyttää Kali Linuxia oppimistehtävien suorittamiseen, koska se sisältää runsaasti työkaluohjelmia, jotka soveltuvat erityisen hyvin tietoturva-auditointiin, murtautumistestaukseen ja takaisinmallinnukseen (*engl. reverse engineering*). Kali Linux on Debian-pohjainen avoimen lähdekoodin Linux-jakeluversio. Kali Linuxin on kehittänyt Offensive Security Ltd., jonka Web-sivulta jakeluversio on asennettavissa useimpiin käyttöjärjestelmiin. Offensive Security ylläpitää myös Exploit Databasea, josta voidaan hakea tunnettuja heikkouksia ja haavoittuvuuksia, ja hyödyntää niitä oppimistehtävissä. Haavoittuvuudet liittyvät esimerkiksi langattomiin sensorinoodeihin tai markkinoilla oleviin IoT-laitteisiin. Kali Linuxiin on myös mahdollista asentaa ohjelmistoja, joita ei ole valmiiksi sisällytetty jakelupakettiin.

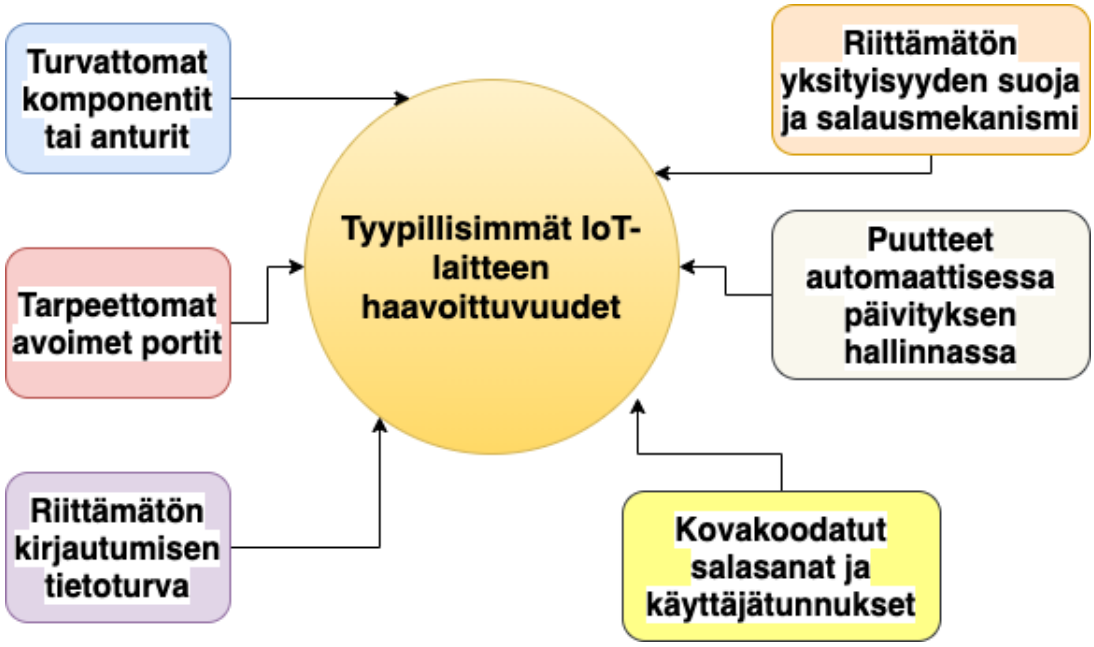
7.4 Penetraatiotestaus oppimismenetelmänä

Tällä hetkellä tieto- ja kyberturvallisuudessa on paljon avoinna olevia työpaikkoja eikä osaajien tarve tule ainakaan lähitulevasuudessa vähenemään. IoT-tietoturvakurssilla käytettiin samoja työkaluja ja tekniikoita, joita haitalliset hakkerit käyttävät. Eettiset seikat huomioiden penetraatiotestauksen oppitehtävissä opetettiin opiskelijalle hakkeri-ajattelutapa, jonka avulla opiskelija voi etsiä hyödynnettävissä olevia aukkoja tietojärjestelmistä ja tietoverkoissa.

Lähtökohtana toisen kehittämissyklin oppimistehtävissä oli opettaa opettajajohtoisen teorian, tiimityön ja vertaisoppimisen menetelmin erityyppisiä penetraatiotestausmentelmiä. Oppimistehtävissä suoritettiin erilaisia testejä haavoittuvuuksien löytämiseksi oppimisympäristön IoT-järjestelmästä. Oppimistehtävissä opeteltiin analysoimaan ja tunnistamaan kohdejärjestelmät, niiden haavoittuvuudet ja kunkin haavoittuvuuden hyödynnettävyys. Tavoitteena oppimistehtävissä oli löytää mahdollisimman monet oppimisympäristöön sijoitetut haavoittuvuudet ja raportoida niistä yleisesti hyväksyttävässä muodossa. Kuvassa 7.3 on havainnollistettu niitä haavoittuvuuksia, joihin oppimisympäristössä pyritään keskittymään. Oppimistehtävien tavoitteena on lisätä opiskeljoiden ymmärrystä IoT-standardeista, jotka eivät ole universaaleja. Lisäksi protokollat ja IoT-laitteiden hallinta vaihtelee suuresti IoT-järjestelmän eri sovelluksissa. Penetraatiotestauksen oppimistehtävillä pyri-

tään lisäämään osaamista IoT-laitteista, jotka tuottavat ja lähettävät suuria määriä dataa, jolloin käyttäjien tiedoista tulee haavoittuvia ja paljastuvat mahdollisesti laajemmalla tasolla. Kaapattujen IoT-laitteiden IP-osoitteita on esimerkiksi markkinoitu pimeässä verkossa (*engl. Dark Web*).

Asiakkaat eivät ehkä ole halukkaita investoimaan teknologiaan, joka ei takaa tietosuojaa ja yksityisyyttä. IoT-laitteet tuottavat valtavia määriä dataa ja eri puolilla maailmaa sijaitsevat etäpalvelut tallentavat saapuvan tiedon. Monilla IoT-laitevalmistajilla on käytössä omat pilvipalvelut, joten väistämättä tulee tilanteita, joissa saatetaan joutua tinkimään tietoturvallisuudesta. Tästä syystä IoT-tietoturvakurssiin sisältyi myös oppimistehtäviä, joilla pyrittiin opettamaan IoT-järjestelmissä olevien anomalioiden havaitsemista. Oppimistehtävissä pyrittiin opettamaan myös perusteet koneoppimisen hyödyntämisen perusteista anomalioiden tunnistamisessa.



Kuva 7.3: IoT-oppimisympäristössä penetraatiotestauksella suoritettavien tärkeimpien haavottuvuuksien analysointi

Oppimistehtävien suorittamisessa käytettiin Kali Linux-käyttöjärjestelmää, jossa on sisäänrakennettuna lukuisia ohjelmia, joilla voidaan suorittaa kattava penetraatiotestaus. Oppimisympäristössä voitiin harjoitella penetraatiotestaustyökalujen käyttöä ja saada ymmärrystä siitä, mitä työkaluja tulee käyttää ja missä tilanteessa. Penetraatiotestausketjua toistettiin eri variaatioilla lähes kaikissa oppimistehtävissä.

7.5 Oppimisympäristöllä saavutetut osaamisvaatimukset

Oppimisympäristölla oli mahdollista suorittaa kattavasti perinteinen Kyber- tai tietoturvahyökkäysketju (*engl. Cyber Kill Chain*). Cyber Kill Chain on sarja vaiheita, joita tarvitaan, jotta hyökkääjä pystyy onnistuneesti tunkeutumaan verkkoon tai IoT-laitteelle ja keräämään tietoja siitä. Opiskelijoiden osaamisen kannalta oli oleellista oppia suunnittelemaan eettisen hakkeroinnin seuranta- ja reagointisuunnitelma. Suunnitelman toteuttamiseen Cyber Kill Chain on tehokas menetelmä, koska siinä keskitytään siihen, kuinka todelliset hyökkäykset tapahtuvat esimerkiksi IoT-järjestelmää tai yksittäistä IoT-laitetta vastaan. Oppimistehtävissä noudatettiin sekä penetraatiotestaus- että STRIDE-menetelmiä. Oppimistehtäviin kuului myös soveltavana osana oppia tunnistamaan mahdolliset anomaliat voidaan tunnistaa hyökkäyksen valmisteluvaiheessa tai hyökkäyksen toteuduttua analysoimalla epänormaalia tietoliikennettä IoT-verkoissa.

Seuraavassa on kuvattu esimerkkejä oppimisympäristöllä onnistuneesti suoritetuista oppimistehtävistä. Oppimistehtävien tarkempi toteutuskuvaus yksityiskohtineen on jätetty tämän työn ulkopuolelle. Oppimistehtävät pyrittiin suunnittelemaan siten, että ne kattaisivat mahdollisimman laajasti IoT-järjestelmään ja IoT-laitteisiin kohdistuvia tietoturvauhkia.

Uhkamallinnus ja penetraatiotestaus: Penetraatiotestaus ja uhkamallinnus *engl. threat modeling* ovat tärkeitä osia missä tahansa sovelluskehityksessä, jossa laitteella tai järjestelmällä on kyky käyttää verkkoyhteyksiä. Hakkerit käyttävät paljon aikaa vain etsiessään tietoa, joka auttaa tai ohjaa hyökkäyksessä. Oppimisympäristössä pystyttiin opettelemaan erilaisia tiedonkeruutekniikoita, kuten IoT-laitteiden porttien skannausta, IoT-järjestelmän laitteiden ja käyttöjärjestelmien analysointia, suorituksessa olevien palvelujen skannaamista ja tietojen kalastelua. Tiedusteluhyökkäys toteutettiin sekä passiivisena että aktiivisena tiedusteluna. ***Passiivinen tiedustelu*** on menetelmä, jossa hakkeri hyödyntää uhrin verkkotunnusta kalastellakseen tai etsiäkseen tietoja. ***Aktiivinen tiedustelu*** on menetelmä, jossa käytetään järjestelmätietoja ja saadaan luvaton pääsy suojattuun digitaaliseen tai elektroniseen materiaaliin. Aktiivisessa tiedusteluhyökkäyksessä kierretään myös reititin ja palomuuri. Esimerkkiharjoitus on kuvattu liitteessä H. Oppimistehtävän soveltavaan osaan kuului verkkoliikenteen analysointi Wiresharkilla ja Python-ohjelmalla, joilla pyrittiin havaitsemnaan epänormaali tietoliikenne. Tässä harjoituksessa yhdistettiin sekä penetraatiotestaus että uhkakamallinnus. Harjoituksen kohteena oli TT-

GO T-Journal ESP32-pohjainen IP-kamera, johon oli koodattu haavoittuvuuksia. IP-kameraan oli myös kovakoodattu käyttäjätunnus ja salasana. Tietojenkalastelu (*engl. Phishing*) on identiteettivarkauden muoto, jossa yritetään kopioida toisen verkkosivun tai sovelluksen ulkoasu. Tietojenkalastelu aiheuttaa sen, että käyttäjät antavat vahingossa tilitietoja ja mahdollistavat täten hyökkääjän yhteyden IoT-laittelle. Aluksi tulee tehdä uhkamallinnusprosessi (*engl. threat modeling process*). Prosessi on kuvattu opetusmateriaalissa H. Prosessia sovelletaan IP-kameraan, jonka haavoittuvuudet ja tietoturvauhkat pyritään tunnistamaan ja arvioimaan. Aluksi kaikki löydetyt uhkat, jotka voivat olla suoritettavissa, arvioitiin korkealle tasolle, vaikka uhka ei johtaisi suoraan tiettyyn hyökkäykseen. Uhkien kriittisyyttä kannattaa arvioitiin lopuksi peilaten hyökkääjään siinä suhteessa, mitä hyökkääjät voisivat hyötyä siitä. Tämä vaihe tehtiin uhkien tarkemassa analysoinnissa. Uhkat määriteltiin sen mukaan millaisia ominaisuuksia IP-kamerassa on olemassa. Esimerkiksi onko siinä infrapunasignaalivastaanotin, kamera, liiketunnistin, mikrofoni ja kaiutin. Seuraavaksi tehtiin uhka-arvioita, jotka liittyvät Android- tai iOS- mobiilisovelluksiin, joita käytetään IP-kameran etäohjaukseen ja myös IP-kameran ottamien kuvien etäkatseluun. Uhka-arvioinnissa arvioitiin myös uhkia, jotka liittyvät käyttäjien sosiaalisessa mediassa jakamiin videoihin tai valokuviin. Uhkamallinnus tehtiin myös liiketunnisteminen antamiin varoituksiin ja IP-kameran pilvipalveluun lähettämiin lämpötila- ja ilmankosteusarvoihin. Lopuksi analysoitiin se, miten laiteohjelmiston päivitykset asennetaan ja miten laite liitetään WLAN-verkkoon ja millaisia tietoturvauhkia näihin sisäsltyy. Seuraavassa osassa tehtävää on tavoitteena toteuttaa penetraatiotestaus, jotka kohdistuvat löydettyihin uhkiin, joiden toteutuessa on riski esimerkiksi tietomurtoihin tai tietojen kalasteluun. Penetraatiotestaus on kuvattu Abdalla et all. [2] IP-kameran tietoturvaa koskevassa julkaisussa.

Tulos: Oppimistavoitteet saatiin toteutettua, ja opiskelijoille tuli osaamista siitä, kuinka tiedonkeruu johtaa tiedustelutietoihin. Saatujen tietojen perusteella pystyttiin suojaamnaa haavoittuviksi havaitut IoT-järjestelmät tai IoT-laitteet. Verkkoliikenteen analysointi ja verkkoliikenteen anomalioiden tunnistaminen ei sen sijaan onnistunut. Analyysin perusteella havaitsimme, että opiskelijoilla tulisi olla vahvempi Wireshark- ja tietoverkko-osaaminen. Uhkamallinnus onnistui annettujen ohjeiden mukaisesti tehtynä hyvin. tehtävän suoritettuan opiskelijat kokivat ymmärtävänsä sen, että tieto- ja kyberturvaa ei voi irrottaa omaksi tehtäväksen IoT-järjestelmäkehityksestä. Samoin IoT-laitteiden suunnitteluun liittyy uhkamallinnuksen toteuttaminen olennaisena osana. Penetraatiotestaus onnistui suurimmaksi osaksi, ainoastaan

verkkoliikenteen analysointi osoittautui haasteelliseksi. Opiskelijoille tuli hyvä ymmärräys, miten eri ohjelmia käyttäen saadaan kattavasti varmennettua IoT-laitteen tietoturva.

Fyysinen tietoturva Eräänlaiseen tiedusteluun kuuluu myös fyysisen tietoturvan haavoittuvuuksien hyödyntäminen. Fyysisen tietoturvan osalta tietoturva-ammattilainen tai eettinen hakkeri etsii arkaluonteisia tietoja, joita käyttäjät hävittävät sopimattomasti. Esimerkki fyysisestä tietoturvaharjoituksesta liittyi sähköpostista tulostettujen viestien etsimiseen esimerkiksi roskakoreista tai etsiä tarralapuihin kirjoitettuja salasanoja tai järjestelmän käyttäjätunnuksia. Tämä tehtävä toteutettiin kirjallisella oppimistehtävällä, jossa opiskelijat kirjoittyivat esseen aiheesta, jossa he käsittelivät järjestelmien fyysistä suojaamista ja sen tärkeyttä yritysmaailmassa ja miksei myöskin yksityisestikin. Esseessä tuli käsittellä sitä, miten materiaalin anastamisen aiheuttaumia vahinkoja voidaan estää ja miten arkaluontoisen materiaalin salaamisella voidaan edesauttaa tietoturvaa. Opiskelijat käsittelivät esseessä myös tietojen hajautuksella ja monistuksella tapahtuvaa tietoturvan parannusta.

Tulos: Tietokonejärjestelmät ovat haavoittuvimmillaan silloin, kun hyökkääjä pääsee fyysisesti käsiksi tietokonejärjestelmään. Opiskelijoiden fyysisen tietoturvan osaamista testattiin kirjallisella oppimistehtävällä ja paneelikeskustelulla aiheesta. Oppimistehtävään liittyi myös löydösten pohjalta tehty tunkeutuminen oppimisympäristön tietokantaan. Tulosten perusteella ei selvästikkään ollut riittävää ymmärrystä fyysisen tietoturvan tärkeydestä.

Tunkeutumistestauksen valmistelu Tunkeutumistestauksen valmisteluharjoituksessa keskityttiin sekä haavoittuneiden koneiden etsimiseen IoT-järjestelmästä että tunkeutumistestauksessa tarvittavaan aseistamiseeen. (*engl. weaponization*) hakkerit käyttävät satoja tuhansia Internetiin kytkettyjä laitteita, jotka olivat jo aiemmin saastuneet. Saastuneet laitteet tunnetaan yleensä Botnet-laitteina, joita tyypillisesti käytetään hajautetun palvelunestohyökkäyksen (DDoS) toteuttamiseen. Hajautetun palvelunestohyökkäyksen esimerkkiharjoitus on kuvattu liitteessä G

Tulos: Tämä oppimistehtävä oli haasteellinen. Vain osa opiskelijoista löysi edes yhden saastuneen IoT-laitteen oppimisympäristön IoT-järjestelmästä. Oppimistehtävän suorittaminen edellytti hyvää osaamista Kali Linux-työkaluista, Nmap-porttiskannauksen systemaattisesta toteuttamisesta ja Wiresharkilla tehdyn tietoliikenteen analysoinnista. Palvelunestohyökkäyksen toteuttaminen ei onnistunut eikä Botnet-ohjelmaa onnistuttu asentamaan saastuneille IoT-laitteille. Johtopäätöksenä to-

tesimme, että tällaisten oppimistehtävien suorittaminen edellyttää myös hyvää tietoverkko-osaamista.

Jamming ja wormhole attack IoT-järjestelmien laitteiden häirintähyökkäysten (*engl. jamming*) tarkoituksena on heikentää tai estää tietoliikenteen toimintaa IoT-järjestelmässä. Häirintä voidaan toteuttaa lähettämällä verkkoon tietoliikenteessä käytettävään protokollaan kuulumattomia ajuussignaaleja. Häirintä vaikuttaa verkon toimintaan vaikeuttamalla oikeiden IoT-noodien datan vastaanottamista ja datan lähettämistä. Häirinnällä voidaan myös vaikuttaa reititystietoihin. Harjoitustehtävä on kuvattu liiteessä C. Tehtävään kuului myös Wormhole-, eli madonreikähyökkäyksen suorittaminen. Wormhole-hyökkäyksessä tallennetaan verkossa lähetettyjä paketteja ja tunneloidaan kaapatut paketit toiselle hyökkääjälle langallisen yhdeyden kautta. Kun madonreikä on muodostettu, on hyökkääjällä mahdollista valikoida reitittämistä ja aiheuttaa verkon haittaavaa toimintaa. Madonreikähyökkäys on kuvattu liitteessä D.

Tulos: Tämä oppimistehtävä oli liian haasteellinen. Oppimistehtävä edellytti usean eri tekniikan ja usean eri ohjelman soveltavaa käyttämistä. Oppimistehtävän suorittaminen edellytti myös hyvää osaamista Kali Linux-työkaluista, Nmap-porttiskannauksen systemaattisesta toteuttamisesta ja Wiresharkilla tehdyn tietoliikenteen analysoinnista. Harjoitustehtävässä korostui langattomien sensoriverkkojen tietoliikenteen osaaminen, reititystietojen ymmärrys ja käytettävien protokollien tuntemus. Johtopäätöksenä totesimme lisäksi, että tällaisten oppimistehtävien suorittaminen edellyttää myös hyvää tietoverkko-osaamista. Samoin madonreikähyökkäys jäi kokonaan suorittamatta.

Toimitus Toimitus (*engl. Delivery*) tarkoittaa sitä, että hyökkääjä aktivoi hyökkäyksen IoT-järjestelmään tai IoT-laitteelle toimittamalla haittaohjelman. Haittaohjelmalla hyökkääjä voi aloittaa tietoverkon salakuuntelun, väärentää IoT-laitteen tunnistetietoja, aloittaa salakuuntelun tai aktivoida jonkin muun hyökkäyksen. Tätä voidaan kutsua myös fyysiseksi hyökkäykseksi IoT-järjestelmää tai IoT-laietta kohtaan. Tässä hyökkääjä käyttää fyysisesti laitetta siten, että tiedustelemalla avoimet dataportit ja lataamalla haittaohjelmien laitteeseen. Tämä voi antaa hyökkääjälle mahdollisuuden hallita laitetta. Haittaohjelma voi myös jakaa haitallista koodia muille IoT-verkon laitteille. Haittaohjelmaa käyttävä harjoitustehtävä on kuvattu liitteessä E.

Tulos: Tämä oppimistehtävä onnistui tehtävänannon mukaisesti. Halutulle IoT-lait-

teelle asennettiin haittaohjelma. Samoin palvelunestohyökkäys toteutettiin onnistuneesti. Tehtävä antoi hyvän käsityksen heikosti suojatun IoT-laitteen hyödyntämisessä esimerkiksi palvelunestohyökkäyksessä. Hyökkäyksen rajaus onnistui myös suhteellisen hyvin. Opiskelijat saivat hyvän perusosaamisen verkkosegmenttien merkityksestä palvelunestohyökkäyksen rajaamisessa ja leviämisen estämisessä.

Asennus (*engl. Installation*) Asennus vaiheessa todellinen haittaohjelma saastuttaa isännän. Hyökkääjä voi myös asentaa ns. takaoven (*engl. backdoor*) kohdelaitteelle, jolloin hyökkääjällä on helppo pääsy kohdejärjestelmään. Tällaisen hyökkäyksen onnistuminen on mahdollista silloin, kun hyökkääjä pystyy ohittamaan järjestelmän tai IoT-laitteen turvajärjestelmän. Harjoituksen esimerkkitoteutus on kuvattu liitteen G tehtävässä, jossa asennettiin haavoittuneille IoT-laitteille Bottiverkko, jonka avulla suoritettiin hajautettu palvelunestohyökkäys.

Tulos: Oppimistehtävän mukaisesti pystyttiin saastuttamaan kohteena ollut IP-kamera. Oppimistehtävään kuuluva takaoven asennus ESP32-sensorinodeen ei sen sijaan onnistunut. Syynä epäonnistumiseen oli ESP32-sensorinoden koodin rakenne, jonka murtaminen vaatisi erityisosaamista. Python koodin testaamista ei ehditty toteuttaa, mutta todennäköisesti takaoven asennus olisi ollut yksinkertaisempaa.

Fuzz-testaus: Fuzz-testaus on laajoissa ohjelmistoprojekteissa käytetty blackbox-, eli mustalaatikkotestausmenetelmä [67]. Fuzz-testauksen hyvä puoli on, ettei kohdejärjestelmästä tarvitse tietää ohjelman suoritusvuota eikä testien suorittamiseksi tarvitse olla lähdekoodia. Kun löydettyjä virheitä aletaan korjata, lähdekoodin tulee olla saatavilla [10]. Testausmenetelmällä pyritään löytämään ohjelmointivirheitä kohdejärjestelmästä syöttämällä virheellisesti muokattua dataa kohdelaitteen tai -järjestelmän protokollarajapintoihin. Syötteitä voidaan ikäänkuin *fuzzata* eli sekoittaa. Yleisimpiä kohteita testaukselle ovat verkkoprotokollat ja ohjelmistovirheet [62]. Fuzz-testaus on tehokas uusien haavoittuvuuksien löytämiseksi. Tämän tyyppisellä testauksella löydetään hyvin usein haavoittuvuudet, jotka saavat kohdejärjestelmän toimimaan odottamattomalla tavalla tai ohjelmointivirheistä johtuen tapahtuu satunnannaisesti ohjelman kaatuilua. Fuzz-testauksella löydetään myös ohjelmassa olevat satunnaisesti esiintyvät muistivuodot [10]. Tämän vuoksi Fuzz-testaus on erityisen hyödyllinen sulautettujen laitteiden Python, C- tai C++-sovelluksissa, joissa jokainen muistin käyttöön vaikuttava virhe on todennäköisesti vakava haavoittuvuus [10]. Fuzz-testaus ei välttämättä riitä yksinään paljastamaan kaikkia muistiriippuvaisia haavoittuvuuksia. Esimerkiksi huolellinen koodianalyysi on

yksi hyvä keino muistin korruptioon liittyvien haavoittuvuuksien löytämisessä. Sulautettujen laitteiden ollessa kyseessä myös GNU-muistinhallintatyökalut ovat varsin käyttökelpoisia Fuzz-testauksen ohessa. Fuzz-testaus ei välttämättä pysty tuottamaan tarvittavaa informaatiota, jonka avulla pystytään tunnistamaan ohjelmakoodissa olevia riippuvaisuuksia (*engl. triggered vulnerabilities*). Tämä johtuu siitä, että Fuzz-testauksessa ei varsinaisesti testata ohjelman suoritusta muisti- ja CPU-tasolla [10]. Fuzz-testauksen esimerkkiharjoitus on kuvattu liitteessä B. Myös liitteessä A kuvatulla harjoituksella harjoiteltiin Fuzz-testaukseen liittyviä taitoja.

Tulos: Tämä tehtävä osoittautui odotusten mukaisesti haasteelliseksi. Oppimistehtävään käytettiin aikaa kahdeksan (8) tuntia. Jatkossa tulee miettiä tarkemmin tämän tyyppisen oppimistehtävän toteutustapaa. Muistivuotojen testaaminen onnistui, mutta koodianalyysissä oli melkoisia haasteita. Vaikka kyseessä oli neljännen vuosikurssin opiskelijat, oli ohjelmointiosaamisessa vielä melkoisesti parannettavaa.

Salasananmurto Oppimistehtävän tavoitteena oli konkreettisesti osoittaa turvallisen salasanan merkitystä IoT-järjestelmien käyttämisessä. Harjoituksessa murrettiin salasanat Ncrack nimisellä työkalulla. Ncrack on sopiva yritysten tietoturvan todentamiseen. Ncrack-ohjelmalla voidaan murtaa mm. SSH, RDP, FTP, Telnet ja HTTPS-salasanoja raakaa voimaa käyttäen (*engl. brute force*). Salasananmurtotehtävä on kuvattu liitteessä F.

Tulos: Opiskelijat oppivat konkretian avulla vahvan salauksen ja vahvan salasanan merkityksen tietoturvan kannalta. Harjoitustehtävässä voitiin todentaa oikeaksi vaatimukset salasanasäännöt, joiden mukaan salasana ei saa olla luonnollinen sana ja salasanan tulee sisältää isoja ja pieniä kirjaimia, numeroita ja erikoismerkkejä. Myös salasanan pituuden merkitys korostui harjoitustehtävää tehdessä.

Tapausten raportointi Eettiseen tieto- ja kyberturvatestaukseen liittyy olennaisena osana myös raportointi asiakasyritykselle. Oppimistehtävissä oli raporttien kirjoittaminen kaikista suoritetuista hyökkäystapauksista, IoT-laitteiden tietoturvan analyysitehtävistä tai IoT-verkon skannauksista.

Tulos: Opiskelijat oppivat esittämään havainnot ja tietoturvalöydökset oikein ja selkeästi ei-tekniselle kohderyhmälle. Opiskelijat oppivat myös dokumentoimaan suunnitelmat tietoturvahaavoittuvuuksien korjaamiseksi.

8 Jatkokehittäminen

Tulevaisuuden IoT-opetusympäristön kehittäminen jatkuu, mutta samalla on myös tarkasteltava opetussuunnitelmaa kokonaisuutena. Oppimisen kannalta ei ole mielekästä uhrata aikaa perusasioiden opetteluun, jos ne voidaan opettaa tehokkaammin muilla kursseilla. IoT-tietoturva on nopeasti kasvava osaamisalue, jonka hallitsemisella pyritään poistamaan IoT-teknologian nykyisiä tietoturvapuutteita. Esittelimme oppimisympäristöä ja oppimistehtäviä yhteistyöyrityksen edustajille. Heidän näkemysten ja kommenttien perusteella muodostettiin uusia oppimistavoitteita, joita oppimistehtävillä pyritään jatkossa saavuttamaan. Seuraavana on luettelo osaamistavoitteista, jotka vastaavat yrityselämän asettamiin vaatimuksiin:

IoT-suunnitteluprosessin ymmärtäminen: IoT-laitteilla on erilaisia rooleja, kuten datan kerääminen tai joidenkin järjestelmän toimintojen toteuttaminen. Opiskelijoiden on opittava tunnistamaan suojattavat kohteet jo suunnitteluprosessissa. Jotkut IoT-järjestelmän ja IoT-laitteet ovat kriittisempiä kuin toiset. Hyvä esimerkki on terveysteknologian IoT-ratkaisut, joilla seurataan ja ohjataan kriittistä järjestelmää ja jossa käsitellään sensitiivistä dataa.

Tuotteen tietoturvan suunnittelun ymmärtäminen: IoT-laitevalmistajat ovat suunnitelleet laitteitaan eri tavoin, koskan standardit pääsääntöisesti puuttuvat. Tietoturvallisuuden kannalta on tärkeää ymmärtää, kuinka pääsynvalvontatoimenpiteet voidaan toteuttaa ja kuinka käyttäjien yksityisyys voidaan taata. On myös tärkeää tunnistaa takaovet (*engl. Backboors*) ja muut huonot tietoturvakäytännöt, jotka voivat johtaa IoT-laitteen haavoittumiseen.

Laitteen suojaaminen: Opiskelijan on ymmärrettävä hyvin IoT-laitteen suunnitteluperiaatteet, jotta hän osaa tehdä tarvittavat muutokset laitteen suojaamiseksi. Nämä muutokset sisältävät parhaiden tietoturvakäytäntöjen käyttöönoton laitteen suojauksen varmistamiseksi. Oppimistehtävissä tulee huomioida, kuinka laitteiden oletussalasanat voidaan vaihtaa, takaovet voidaan poistaa, kryptografia voidaan ottaa käyttöön, kuinka IoT-laite voidaan nollata ja poistaa siitä esimerkiksi Botnet-verkot.

Tietoverkko-osaaminen Osaamisvaatimuksiin tulee sisällyttää tietoverkkojen pe-

rusosaaminen. Opiskelijalla on oltava osaaminen tietoverkkojen segmentoinnista, ja kuinka kaikki IoT-laitteet voidaan yhdistää eristettyyn ja suojattuun verkkoon. Laiteen suojaamiseksi tehtyjen muutosten tulee olla suunniteltuja, jotta estetään mahdolliset vahingot tai palvelun häiriöt itse IoT-laitteelle.

Tietoturvakäytäntöjen toteutus: Jotta voidaan varmistaa standardoitu tietoturva kaikille IoT-laitteille, on opiskelijan tärkeää osata määrittää suojauskäytännöt, joita noudatetaan IoT-laitteiden osalta. IoT-laitteiden luonteesta johtuen koko organisaation vaarantamiseen tarvitaan vain yksi turvaton IoT-laite. Siksi opiskelijalla on oltava ymmärrys siitä, että turvallisuuskäytäntöjä tulee noudattaa tarkasti.

Opiskelijan tietojen ja taitojen kehittämisen näkökulmasta on tärkeää suunnitella kokemuksen ja osaamisen hankkimista alla mainituilta alueilta:

- Osaamista sovellusuhkien mallinnuksesta,
- Web-palvelimeen ja tietokantoihin liittyvät tietoturvataidot,
- Vahva sovellusprotokollien tuntemus,
- Osaamista turvallisista koodausmenetelmistä ja sovellusten turvallisesta muistinkäytöstä,
- Osaamista suorittaa suojatun koodin tarkistuksia (automaattisesti ja manuaalisesti),
- Tietämystä todennus-ja valtuutusmekanismeista ja niihin liittyvien teknologioiden standardeista ja ominaisuuksiata,
- Tietoa haavoittuvuuksista sekä siitä, miten niitä hyödynnetään,
- Sovellusten ja tietojärjestelmien penetraatiotestaustaidot,
- Tietämystä tietoturva-arkkitehtuureista ja niiden kehittämisestä,
- Ymmärrystä ja osaamista Blackbox (musta), Graybox (harmaa), ja Whitebox (valkoinen) laatikkotestauksiin liittyvistä eettistä ohjeista ja testausten suorittamismenetelmistä ja
- Laajempaa osaamista anomalioiden tunnistamiseen.

9 Johtopäätökset ja pohdinta

Pro gradu -tutkielma toteutettiin kehittämistutkimuksen menetelmin, jonka kahdessä eri kehittämisyksyklissä tuotettiin kohtuullisen kattava IoT-tietoturvan oppimisympäristö. Tutkimuskysymykset toimivat vaatimusmäärittelynä toteutettavalle oppimisympäristölle. Ensimmäisen tutkimuskysymyksen avulla haettiin myös vastauksia siihen millainen tarve IoT-tietoturva osaamisella on tutkimushetkellä, ja millaista osaamista valmistuvilta opiskelijoilta odotetaan. Toisen tutkimuskysymyksen tavoitteena oli ratkaista oppimisympäristölle asetettavat laite- ja järjestelmävaatimukst, jota suunnittelussa tulee ottaa huomioon. Kolmannessa tutkimuskysymyksessä keskityttiin oppimistehtävien suunnitteluun ja siihen millaiset oppimistehtävät olisivat oppimistavoiteiden mukaisia.

Teoreettisen tutkimuksen tuloksena saatiin hyvä käsitys IoT-järjestelmien tyypillisistä haavoittuvuuksista. Teoreettisessa osuudessa käsiteltiin myös IoT-järjestelmiä, IoT-laitteista ja IoT-järjestelmiin liittyviä data- ja tietoliikenneprotokollia. Teoreettisen osuuden yllättävin huomio oli se, että IoT-laitteiden suunnittelussa kiinnitetään päähuomio laite- ja järjestelmäsuunnitteluun, vaikka IoT-laitteiden, palvelimien ja käyttäjäsovellusten välinen vuorovaikutus ovat IoT:n olennaisia osia. Samassa yhteydessä huomioimme myös, että varsinkin kuluttajalaitteiden tietoturvaan ei juurikaan kiinnitetty huomiota. Oppimistehtävien osalta merkittävin huomio oli se, että standardit ja protokollat tekevät IoT- järjestelmäsuunnittelusta erityisen tärkeän osa-alueen. Tutkimusten mukaan suurin osa tieto- ja kyberturvahaavoittuvuuksista kohdistuu joko suoraan verkkokerrokseen tai verkkokerroksen välityksellä IoT-laitteisiin.

Empiirisessä ongelma-analyysissä kartoitimme Kajaanin ammattikorkeakoulun yhteistyöyritysten ansiokkaalla avustuksella valmistuvien tieto- ja viestintätekniikan insinöörien tieto- ja kyberturvan osaamistarvetta. Oppimistehtävien suunnittelussa korostimme empiirisen ongelma-analyysin tulosten perusteella sitä, että tärkeintä on kyberturvallisuuden kannalta on tieto ei yleisessä käytössä oleva teknologia. Erityisesti osaamista tulisi olla tietokannat, käyttäjätunnukset ja salasanat, teknologiset patentit, kriittiset koodit ja tiedonsiirron kannalta tietoverkkojen reititystiedot ja IP-osoitteet siten, että niihin kohdistuvat uhkat voidaan minimoida.

Empiirisen ongelma-analyysin tuloksena konkretisoitui tietoliikennetekniikan perusteiden osaaminen, tietoverkkojen perusteiden hallinta, autentikointimenetelmien perusosaaminen, tietoturvallisen ohjelmakoodin tekeminen, ja salaus ja salausalgoritmien toimintaperiaatteiden ymmärtäminen.

Kehitystyössä jouduimme tekemään runsaasti kompromissejä, mutta pääpaino pidettiin kuitenkin IoT-laitteissa ja IoT-järjestelmissä ja niihin liittyvissä tietoturvaongelmissa. Suunnitteluvaiheen ja toisen kehittämissyklin lopputulokset osoittavat sen, kuinka haasteellista on toteuttaa käytännönläheiseen opetukseen soveltuva oppimisympäristö, joka olisi samalla skaalautuva ja yrityslähtöinen.

Kehitystyön aikana huomasimme, että on myös tarvetta uudistaa muutaman muun kurssin opetussisältöä, jotta opiskelijoilla olisi paremmat valmiudet suoriutua IoT-tietoturvakurssin harjoitustehtävistä. Merkittävimmät muutokset kohdistuvat Tietoverkot-kurssin opetussisältöön.

Kehitetyssä oppimisympäristössä on mahdollista toteuttaa opetusta langattomien sensoriverkkojen ja IoT-järjestelmien kohdistuviin haavoittuvuuksiin ja haavoittuvuuksien vaikutuksiin luottamuksellisuuteen, eheyteen ja saatavuuteen. Kehitystyön tuloksena toteutettiin useita IoT-laitteita ja Internet-verkon toimilaitteita sisältävä oppimisympäristö, joka voidaan tarvittaessa irrottaa julkisesta Internetistä. Oppimisympäristön kehityssykleissä suunniteltiin ja pilotoitiin useita erilaisia oppimistehtäviä. Kehityssykleissä analysoitiin myös oppimistehtävien soveltuvuutta IoT-järjestelmän tieto- ja kyberturvallisuuden testausmenetelmien opettamiseen. Parhaaksi vaihtoehdoksi todettiin IoT-järjestelmän tietoturvan opettaminen siten, että opetus toteutetaan noudattamalla penetraatiotestauksen menetelmiä. Didaktinen malli noudattaa Kajaanin ammattikorkeakoulussa käytössä olevaa sosiokonstruktivististä oppimiskäsitystä, jossa korostetaan tekemällä oppimista teoreettisen tiedon pohjalta. Työn tuloksena saatiin myös ymmärrystä siitä, mitä osaamista opiskelijoilla tulee olla ennen IoT-tietotutvakurssin suorittamista. Merkittävin osaamisvaatimus liittyi tietoverkkoihin ja tietoliikenneteknologiaan osaamiseen. Tämän työn tuloksena päätettiin myös kehittää edeltävien opintojen sisältöä.

Oppimisprosessin analysoinnin ja opiskelijoiden osaamisen kehittymisen seurannan tuloksena voitiin todeta kuitenkin, että suurin osa kurssin osaamistavoitteista pystyttiin täyttämään. Opiskelijoilla on merkittävästi paremmat edellytykset tulevissa työtehtävissään toteuttaa tietoturvaan liittyviä työtehtäviä. Sen lisäksi ymmärrys tietoturvallisen ohjelmoinnin merkityksestä lisääntyi huomattavasti. Langattomiin sensoriverkkoihin kohdistuu paljon erilaisia uhkia, joilta suojautuminen

on tärkeä tutkimusaihe. Kurssitoteutusten analysoinnissa mietimme samalla, miten oppimisympäristöä ja oppimistehtäviä tulisi kehittää. Tärkeimpiä kehityskohteita ovat tietoturvallisen IoT-suunnitteluprosessin ja tietoturvan suunnittelun ymmärtäminen. Oppimistehtävien kehittämisessä merkittävin asia olisi soveltavan osaamisen korostaminen.

Lähteet

[1] ABDALLA, P. A., JA VAROL, C. Testing IoT Security: The Case Study of an IP Camera. Julkaisusarjassa *2020 8th International Symposium on Digital Forensics and Security (ISDFS)* (Beirut, Libanon, 2020), IEEE, 1–5.

[2] ABDALLA, P. A., JA VAROL, C. Testing IoT Security: The Case Study of an IP Camera. Julkaisusarjassa *2020 8th International Symposium on Digital Forensics and Security (ISDFS)* (Beirut, Libanon, Kesäkuu 2020), IEEE, 1–5.

[3] ABEBE, D., NAVEEN, C., VAN-DOAN, N., JA WILL, H. A Comprehensive Study of Anomaly Detection Schemes in IoT Networks Using Machine Learning Algorithms. *Sensors 21* (2021).

[4] AIOTI. AIOTI pilot project. URL: `https://aioti.eu/`, viitattu 13.3.2022.

[5] AKSELA, M., JA PERNAA, J. Kehittämistutkimus pro gradu - tutkielman tutkimusmenetelmänä. Julkaisusarjassa *Teoksessa J. Pernaa (toim.), Kehittämistutkimus opetusalalla.* (Suomi, Jyväskylä, 2013), PS-Kustannus, 181–200.

[6] ALLADI, T., CHAMOLA, V., JA SIKDAR, B. Consumer IoT: Security Vulnerability Case Studies and Solutions. *IEEE Consumer Electronics Magazine 9* (2020), 17–25.

[7] ARDUINO.CC. Arduino Uno R3. URL: `https://docs.arduino.cc/hardware/uno-rev3/`, viitattu 30.3.2022.

[8] BAGHAEI, N., JA HUNT, R. IEEE 802.11 wireless LAN security performance using multiple clients. Julkaisusarjassa *12th IEEE International Conference on Networks (ICON 2004)* (Singapore, Marraskuu 2004), 299 – 303.

[9] BASTOS, D., SHACKLETON, M., JA EL-MOUSSA, F. Internet of Things: A survey of technologies and security risks in smart home and city environments. Julkaisusarjassa *Living in the Internet of Things: Cybersecurity of the IoT* (Lontoo, Englanti, Kesäkuu 2018), IET, 1–7.

[10] BELLETTINI, C., JA RRUSHI, J. L. Vulnerability Analysis of SCADA Protocol Binaries through Detection of Memory Access Taintedness. Julkaisusarjassa *2007 IEEE SMC Information Assurance and Security Workshop* (West Point, NY, USA, Kesäkuu 2007), IEEE, 341–348.

[11] BLUETOOTH INC. Bluetooth Wireless Technology. URL: https://www.bluetooth.com/learn-about-bluetooth/tech-overview/, viitattu 6.12.2021.

[12] CAM-WINGET, N., SADEGHI, A.-R., JA JIN, Y. Can IoT be secured: Emerging challenges in connecting the unconnected. Julkaisusarjassa *2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC)* (Austin, TX, USA, Kesäkuu 2016), IEEE, 1–6.

[13] CISCO SYSTEMS, INC. Introduction to Cybersecurity. URL: https://www.netacad.com/courses/cybersecurity/introduction-cybersecurity/, viitattu 13.3.2022.

[14] DIGI- JA VÄESTÖTIETOVISRASTO. Turvallisen sovelluskehityksen käsikirja. URL: https://www.suomidigi.fi/sites/default/files/2020-05/Turvallisensovelluskehityksenkäsikirja.pdf/, viitattu 14.4.2022.

[15] DONNO, M. D., DRAGONI, N., GIARETTA, A., JA SPOGNARDI, A. DDoS-Capable IoT Malwares: Comparative Analysis and Mirai Investigation. *Security and Communication Networks 2018* (2018), 1–30.

[16] EBERT, C., REKIK, Y., JA KARADE, R. Security Test. *IEEE Software 37*, 2 (2020), 13–20.

[17] ENGELBERG, J. Oppimisesta ja opetusmalleista yliopistokoulutuksessa. Julkaisusarjassa *Opettajatiedon kipinöitä. Kirjoituksia pedagogiikasta* (Joensuu, 2000), Joensuun yliopisto. Savonlinnan opettajankoulutuslaitos, 7–33.

[18] ESPRESSIF SYSTEMS LTD. ESP32 WROVER datasheet. URL: https://www.espressif.com/sites/default/files/documentation/esp32-wrover_datasheet_en.pdf/, viitattu 22.02.2022.

[19] FETTE, I., JA MELNIKOV, A. *The WebSocket Protocol*, Joulukuu 2011.

[20] GERALD COMBS. Wireshark. URL: https://www.wireshark.org/, viitattu 13.3.2022.

[21] GITHUB, INC. LILYGO TTGO T-Camera Plus ESP32. URL: https://github.com/Xinyuan-LilyGo/esp32-camera-screen/, viitattu 22.02.2022.

[22] GUZMAN, A., JA GUPTA, A. *IoT Penetration Testing Cookbook: Identify vulnerabilities and secure your smart devices*. Packt Publishing Ltd., 2017.

[23] GYUSUN, H., JEONG-CHEOL, L., JINWOO, P., JA TAI-WOO, C. Developing performance measurement system for Internet of Things and smart factory environment. *International Journal of Production Research 55*, 9 (2017), 2590–2602.

[24] HASAN, M., ISLAM, M., ZARIF, I. I., JA HASHEM, M. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet of Things 7* (2019).

[25] HYPPÖNEN, O., JA LINDÉN, S. *Opettajan käsikirja: opintojaksojen rakenteet, opetusmenetelmät ja arviointi*. Teknillinen korkeakoulu, Helsinki, 2009.

[26] IEEE.ORG. *IEEE 802.15.4-2003*, 2003.

[27] IOTSF. IoT Security Compliance Framework. URL: https://www.iotsecurityfoundation.org/best-practice-guidelines/, viitattu 8.11.2021.

[28] IVANA, T., JA A., M. J. A Survey of Potential Security Issues in Existing Wireless Sensor Network Protocols. *IEEE Internet of Things Journal 4* (2017), 1910–1923.

[29] JEN CLARK. What is the Internet of Things? URL: https://www.ibm.com/blogs/internet-of-things/what-is-the-iot/, viitattu 22.10.2021.

[30] JÄRVINEN, P. *Kyberuhkia ja somesotaa*. Docento, Helsinki, 2018.

[31] KAJAANIN AMMATTIKORKEAKOULU. Opinto-opas. URL: https://opinto-opas.kamk.fi/index.php/fi/68146/fi/68097/, viitattu 14.2.2022.

[32] KANGAS, M., KOPISTO, K., JA KROKFORS, L. Tulevaisuuden koulussa opitaan kaikkialla, yhdessä ja luovasti - elämää varten". Julkaisusarjassa *Kansankynttilä keinulaudalla* (Suomi, 2016), PS-kustannus, 77–94.

[33] KANSALLINEN KOULUTUKSEN ARVIOINTIKESKUS. Korkeakoulujen arviointikäsikirja 20192024. Julkaisut 19:2019. URL https://karvi.fi/app/uploads/2019/09/KARVI_1919.pdf, viitattu 11.10.2021.

[34] KASPERSKY LAB. Xiaomi Mi Robot vacuum cleaner hacked. URL: https://www.kaspersky.com/blog/xiaomi-mi-robot-hacked/20632/, viitattu 24.02.2022.

[35] KAUPPILA, R. *Ihmisen tapa oppia. Johdatus sosiokonstruktivistiseen oppimiskäsitykseen.* PS-Kustannus, Jyväskylä, 2007.

[36] KEMPPAINEN, V. KajaPro Oy, Toimitusjohtaja. Haastattelu Kajaanissa 12.10.2021.

[37] KHAN, R., MCLAUGHLIN, K., LAVERTY, D., JA SEZER, S. STRIDE-based threat modeling for cyber-physical systems. Julkaisusarjassa *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)* (Turin, Italia, Tammikuu 2017), IEEE, 1–6.

[38] KINKKI, T. Bittium Oyj, Site Manager. Haastattelu Kajaanissa 4.5.2021.

[39] MENDEZ, D., PAPAPANAGIOTOU, I., JA YANG, B. Internet of things: Survey on security. *Information Security Journal: A Global Perspective 27*, 3 (2018), 162–182.

[40] MENDEZ, D., PAPAPANAGIOTOU, I., JA YANG, B. Internet of Things: Survey on Security and Privacy. *Information Security Journal 27* (2018), 1–16.

[41] MINTEER, A. *Analytics for the Internet of Things (IoT).* Packt Publishing Ltd, Birmingham, UK, 2017.

[42] MÄENPÄÄ IMPORT OY / SUOMEN TURVATUOTE. Mikä on IP valvontakamera? URL: https://suomenturvatuote.fi/page/11/mika-on-ip-valvontakamera/, viitattu 24.02.2022.

[43] NIILO NISKANEN. IoT-anturiverkon toteutus tietoverkkojen opetusympäristöön., 2021. URL: https://www.theseus.fi/bitstream/handle/10024/497222/Niilo_Niskanen_Opinnäytetyö.pdf?sequence=2&isAllowed=y/, viitattu 8.2.2022.

[44] ONVIF.ORG. ONVIF Application Programmer's Guide. URL: https://www.onvif.org/wp-content/uploads/2016/12/ONVIF_WG-APG-Application_Programmers_Guide-1.pdf/, viitattu 24.02.2022.

[45] ONVIF.ORG. ONVIF Profiles. URL: https://www.onvif.org/profiles/, viitattu 24.02.2022.

[46] PASHEL, B. A. Teaching students to hack: ethical implications in teaching students to hack at the university level. Julkaisusarjassa *Proceedings of the 3rd Annual Conference on Information Security Curriculum Development* (New York, NY, USA, Syyskuu 2006), ACM, 197–200.

[47] PERNAA, J. *Kehittämistutkimus opetusalalla*. PS-Kustannus, Jyväskylä, 2013.

[48] PERRONE, G., VECCHIO, M., PECORI, R., JA GIAFFREDA, R. Julkaisusarjassa *The Day After Mirai: A Survey on MQTT Security Solutions After the Largest Cyberattack Carried Out through an Army of IoT Devices* (Porto, Italia, Tammikuu 2017), ResearchGate, 246–253.

[49] PLUMMER, D. C. *An Ethernet Address Resolution Protocol*, 1981. URL: `https://www.rfc-editor.org/info/rfc826/`, viitattu 7.12.2021.

[50] POIKELA, E., JA JÄRVINEN, A. Työssä oppimisen prosessimalli opettamisen johtamisessa ja osaamisen arvioinnissa. Julkaisusarjassa *Työ, identiteetti ja oppiminen* (Suomi, 2007), WSOY, 178–197.

[51] POIKELA, E., JA POIKELA, S. Ongelmaperustainen pedagogiikka eilen, tänään ja huomenna. *Kasvatus ja Aika 4* (2010), 91–120.

[52] RASPBERRY PI LTD. Raspberry Pi 4. URL: `https://www.raspberrypi.com/products/raspberry-pi-4-model-b/`, viitattu 22.02.2022.

[53] RASPBERRY PI LTD. Raspberry Pi OS. URL: `https://www.raspberrypi.com/documentation/computers/os.html#introduction/`, viitattu 22.02.2022.

[54] RIAZ, M. N., BURIRO, A., JA MAHBOOB, A. Classification of Attacks on Wireless Sensor Networks: A Survey. *Wireless and Microwave Technologies 6* (2018), 15–39.

[55] RONKAINEN, J. Bittium Oyj, Project Manager. Haastattelu Kajaanissa 4.5.2021.

[56] ROUSKU, K., JA JÄRVINEN, P. *Työpaikan tietoturvaopas - Tunnista uhat, hallitse riskit*. Alma Talent, Helsinki, 2018.

[57] RUSSELL, B., JA DUREN, D. V. *Practical Internet of Things Security - Second Edition.* Packt Publishing Ltd, Birmingham, UK, 2018.

[58] SHODAN. Search Engine for the Internet of Everything. URL: https://www.shodan.io/, viitattu 13.3.2022.

[59] SIROHI, P., AGARWAL, A., JA TYAGI, S. A comprehensive study on security attacks on SSL/TLS protocol. Julkaisusarjassa *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)* (Dehradun, Intia, Lokakuu 2016), IEEE, 893–898.

[60] TEKNOLOGIATEOLLISUUS RY. Digitaalinen turvallisuus yhä tärkeämpää. URL: https://www.teknologiateollisuus.fi/fi/tyomarkkinat/yritysturvallisuus/digitaalinenturvallisuusyhätarkeampaa/, viitattu 20.5.2021.

[61] TEXAS INSTRUMENTS INCORPORATED. 6LoWPAN demystified. URL: https://www.ti.com/lit/wp/swry013/swry013.pdf?ts=1644581500885&ref_url=https%253A%252F%252Fwww.google.com%252F/, viitattu 7.2.2022.

[62] TULASI, A., SUKUMARA, ND KUMAR RAMAKANTH, S., ET AL. Robustness Evaluation of Cyber Physical Systems through Network Protocol Fuzzing. Julkaisusarjassa *2019 International Conference on Advances in Computing and Communication Engineering (ICACCE)* (Sathyamangalam, Tamil Nadu, Intia, Toukokuu 2019), IEEE, 1–6.

[63] VELODYNE LIDAR, INC. What is Lidar? URL: https://velodynelidar.com/what-is-lidar/, viitattu 13.3.2022.

[64] VELU, V. K. *Mastering Kali Linux for Advanced Penetration Testing - Fourth Edition*. Packt Publishing Ltd, Birmingham, UK, 2022.

[65] WILLIAMS, R., MCMAHON, E., SAMTANI, S., PATTON, M., JA CHEN, H. Identifying vulnerabilities of consumer Internet of Things (IoT) devices: A scalable approach. Julkaisusarjassa *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)* (Peking, Kiina, Heinäkuu 2017), IEEE, 179–181.

[66] YLONEN, T., JA LONVICK, C. *The Secure Shell (SSH) Transport Layer Protocol*, Tammikuu 2006.

[67] ZHANG, H., JA LU, K. SIoTFuzzer: Fuzzing Web Interface in IoT Firmware via Stateful Message Generation. *Applied Sciences 11* (2021), 1–18.

[68] ZHOU, Y., FANG, Y., JA ZHANG, Y. Securing wireless sensor networks: a survey, 2008.

A Puskuriylivuotoharjoituksen kuvaus

Sen jälkeen kun uhkamalli on luotu ja uudet uhat on lueteltu ja arvioitu seuraava askel on harjoituksissa on koodianalyysi ja puskuriylivuototestaus. On olemassa useita erilaisia lähestymistapoja testauksen toteuttamiseksi. Yleisimmät menetelmät ovat: Black box-, Gray box- ja White box-testaus.

Harjoitus 1. - Yleisiä haavoittuvuuksia
Kolme yleisimmistä ohjelmistohaavoittuvuuksista ovat:

- kokonaislukuvirheet,
- syötteiden vahvistusvirheet ja
- puskurin ylivuodot.

Tehtävä 1: Tutki seuraavaa koodia:

Listing A.1: Virhe muistiin kopioimisessa

```
char user_input[512];
char buffer[256];

gets(user_input);
strcpy(buffer, user_input);
```

Tee testiohjelmat sekä Black box- että White box-testauksena. Korjaa havaitsemasi haavoittuvuus. Kirjoita lyhyt kuvaus haavoittuvuudesta.

Tehtävä 2: Tutki seuraavaa koodia:

Listing A.2: Virhe muistiin kopioimisessa

```
#include <stdio.h>

int main(int argc, char **argv) {
  char buf[8]; // buffer for eight characters
```

```
gets(buf); // read from stdio (sensitive function!)
printf("%s\n", buf); // print out data stored in buf
return 0; // 0 as return value
}
```

Tee testiohjelmat sekä Black box- että White box-testauksena. Korjaa havaitsemasi haavoittuvuus. Kirjoita lyhyt kuvaus haavoittuvuudesta.

B Fuzz-testausharjoituksen kuvaus

Tehtävä 1: Fuzz-testausharjoitus: Fuzz-testaus on tietoturvaa painottava testausmenetelmä, jossa virheitä ja haavoittuvuuksia etsitään käyttämällä odottamattomia ja virheellisiä syötteitä.
Tutki seuraavaa koodia:

Listing B.1: Fuzz-testausharjoitus

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

//function prototype
void granted();
int checkPasswd();

int checkPasswd(){
    char passwd[16];
    printf("Enter␣your␣passwd:␣");
    gets(passwd);
    if(strcmp(passwd, "passwd1")){
     printf("\nYou␣fail!\n");
    }
    else{
        granted();
    }
    return 0;
}

void granted(){
    printf("\nAccess␣granted\n");
    printf("You␣have␣gotten␣the␣privileges,␣and␣can␣do␣anything␣you␣l
```

```
// Privileged stuff happens here.
    return ;
}

int main(void){
    checkPasswd();
    return 0;
}
```

Testaa koodia erilaisilla syötteillä. Korjaa havaitsemasi haavoittuvuus. Kirjoita lyhyt kuvaus haavoittuvuudesta ja kerro mitä Fuzz-testaus tarkoittaa.

C Fyysisen kerroksen haavoittuvuuteen kohdistuva hyökkäys - Jamming

Jamming eli häiriö johtuu verkkolaitteiden radiotaajuuksien estymisestä, jonka avulla hyökkääjä voi ohjata verkkoliikenteen toiminnan tietystä alueesta itselleen.

Tehtävänä on suorittaa Sinkhole-hyökkäys. Sinkhole-hyökkäys on tuhoisin IoT-järjestelmään kohdistuva reitityshyökkäys. Tehtävässä luodaan väärennetty verkkoliikenne ja katkaistaan verkkoviestintä IoT-nodejen välillä. Sinkhole-hyökkäyksessä luodaan ensin väärennetyt reititystiedot ja lähetetään seuraavaksi väärennetyt reitityspyynnöt haavoittuneesta IoT-solmusta naapurisolmuille. Tehtävässä tulee väärentää yhteyden laatua siten, että hyökkääjän SinkNode ei enää näy naapurisolmuille.

Tehtävä edellyttää verkkotiedustelun tekemisen esim. NMap-sovelluksella. Tietyn verkon porttiskannauksella voidaan tunnistaa avoimet portit ja verkkoon liitetyt laitteet. Jokainen portti tarjoaa tietyn palvelun, joten tehtävässä tulee ottaa kohteeksi portti 22 (SSH) ja portti 23 (Telnet). Oppimistehtävässä opit tunnistamaan avoimet portit, jonka jälkeen hyökkääjällä on tarkoituksena päästä laitteeseen. Tehtävässä tulee käyttää Kali Linux-konetta hyökkäyksen ja verkkotiedustelun suorittamiseen.

Tee lyhyt kuvaus kuinka suoritit tehtävän ja kuinka hyökkäys mielestäsi onnistui. Kerro myös minkä IoT noden haavoittuvuuden kautta pääsit sisälle nodeen. Ota kuvakaappaus graafisesta verkkokuvasta ennen hyökkäystä ja hyökkäyksen ollessa käynnissä.

D Verkkoliikenteen analysointihyökkäyksen kuvaus - Wormhole attack

Wormhole-hyökkäys: Wormhole-hyökkäys on vakava hyökkäys, jossa kaksi hyökkääjää sijoittuu strategisesti IoT-verkkoon. Hyökkäys kohdistuu verkkokerrokselle. Hyökkääjien tavoitteena on kuunnella verkkoliikennettä ja tallentaa langattomasti liikkuvaa tietoa. Hyökkääjät hyödyntävät sekä verkon ja IoT-laitteiden analysoinnin tuloksena syntynyttä sijaintitietoa. Hyökkääjien on asemoiduttava siten, että niillä on lyhin reitti IoT-solmujen joukossa. Kun hyökkääjäsolmut luovat suoran linkin toistensa välille verkossa, niin toisella puolella oleva Wormhole-hyökkääjä vastaanottaa paketteja ja siirtää ne verkon toiselle puolelle.

Tehtävänä on suunnitella hyökkäys siten, että työparisi kanssa luotte ensin graafisen kuvan IoT-laitteista ja verkossa olevista toimilaitteista. Kannattaa valita sellaiset IoT-noodit, joissa käytetään 802.15.4. radioita. Kohteeksi kannattaa ottaa siis Zigbee-sensorinoodit, joiden reititystiedot ja dataliikenne tulee ensin analysoida. Kun sopiva liikenne on kaapattu, asetetaan kaapatussa nodessa reittitietueotsikossa hyppyjen määrä nollaan. Lisäksi lähde- ja kohde-MAC-osoitteet muutetaan vastaamaan uhrisolmujen verkko-osoitteita. Seuraava tehtävä on suorittaa Man-in-The-Middle (MiTM)-hyökkäys, eli kerätä tietoja ja keskeyttää tiedonkulku, jolloin muut nodet ja reitittimet eivät ole käytettävissä. MiTM-hyökkäyksessä sijoitat kaapatun noden verkkoon kahden anturinsolmun tai anturin ja reititinsolmun tietovirran keskelle. Pyri kaappaamaan niin paljon liikennettä kuin mahdollista.

Analysoi vastauksessa tehtävän kulkua, kerro käyttämäsi ohjelmat ja kuinka niitä hyödynsit. Perehdy oppimateriaalien avulla MANET-verkossa tapahtuvaan Wormhole-hyökkäykseen, ja miten se vaikuttaa ns. Ad-hoc-verkkojen toimintaan.

E IoT-laitteen injektioharjoitus

Harjoituksen ensiimäinen osa toteutettiin käyttämällä Mirai-haittaohjelmaa. Haittaohjelma leviää haavoittuville laitteille etsimällä jatkuvasti verkosta IoT-laitteita, jotka on suojattu ainoastaan tehdasasetuksella tai kovakoodatulla käyttäjätunnuksella ja salasanalla. Haavoittuvan laitteen löytäessään Mirai-haittaohjelma ottaa laitteen haltuun ja alkaa monistamaan itseään haavoittuneen koneen kautta uusiin laitteisiin. Lopulta saastuneista IoT-laitteista muodostetaan bottiverkko. Hyökkääjä voi tämän jälkeen komentaa bottiverkoon liitettyjä IoT-laitteita lähettämään yhteyspyyntöjä yhteen tiettyyn kohteeseen. Tällöin yhtäaikainen verkkoliikenne muodostuu niin suureksi, että kohdejärjestelmä ei enää pysty vastaamaan palvelupyyntöihin. Harjoituksessa toteutettiin konkreettisesti palvelunestohyökkäys, jonka kohteena oli IoT-oppimisympäristössä oleva web-palvelin. Harjoituksessa käytettiin Telnet-protokollaa yksinkertaisuuden vuoksi. Harjoituksen kohde IoT-laitteina käytettiin haavoittuvaa IP-kameraa sekä kahta Raspberry Pi -klusteria, joissa kummassakin oli neljä (4) kpl Raspberry Pi -minitietokoneita. Raspberry Pi -klusterit oli jaettu eri IP-osoiteavaruuksiin. Harjoituksessa käytettävä lähdekoodi on löydettävissä internetistä. Tehtävän tarkempi kuvaus, ohjelman kääntäminen ja hyökkäyksen eteneminen toteutettiin opettajan ohjauksessa. Harjoitusta tehtäessä IoT-oppimisympäristö irrotettin julkisesta internetistä.

Opiskelijoiden osatehtävänä oli myös analysoida IoT-laitteiden merkitystä esimerkiksi palvelunestohyökkäyksissä. Opiskelijoiden tehtävänä oli etsiä vastaus kysymykseen: "Miksi IoT-laitteet ovat niin kiinnostavia verkkorikollisten näkökulmasta?"

Seuraavana osatehtävävnä oli selvittää haittaohjelman etenemistä haavoittuvissa IoT-laitteissa. Opiskeljoiden tavoitteena oli estää haittaohjelman etenemistä rajaamalla saastuneet tietoverkon osat pois IoT-oppimisympäristön verkkoratkaisusta.

Kolmantena osatehtävänä opiskelijat tutustuivat EU:n kyberturvallisuusdirektiiviin kyberturvallisuusstrategiaan. Direktiivissä määritellään minimiturvallisuustaso IoT-laitteiden ja kriittiselle infrastruktuurille. Internetiin yhdistetään kasvamassa määrrin Iot-kulutustavaroita ja teollisuuden IoT-laitteita. Lisääntyvän käytön myötä aiheutuu väistämättä tietoturvariskejä yksityisyydelle, tieto- ja kyberturvallisuu-

delle.

Lähde referoitavaan materiaaliin löytyy osoitteesta: Kyberturvallisuus: miten EU torjuu kyberuhkia?.

Opiskelijat referoivat myös EU:n suositusta, jossa määritellään minimiturvallistaso IoT-laitteille. Kyseessä on vielä suositus ja koskee vain EU:n sisämarkkinoita. Lähde referoitavaan materiaaliin löytyy osoitteesta: Baseline Security Recommendations for IoT.

F Salasanamurtotehtävien kuvaus

Salasanamurtotehtävän lähteenä on käytetty kirjaa "Nmap Network Exploration and Security Auditing Cookbook - Third Edition"Ncrack on verkon todennusmurtotyökalu, joka on suunniteltu tunnistamaan järjestelmät, joiden tunnistetiedot ovat heikkoja. Se on erittäin joustava ja tukee suosittuja verkkoprotokollia, kuten FTP, SSH, Telnet, HTTP(S), POP3(S), SMB, RDP, VNC, SIP, Redis, PostgreSQL ja MySQL. Ncatin avulla voidaan lukea, kirjoittaa, uudelleenohjata ja muokata verkkotietoja erittäin monipuolisilla tavoilla. Ncat tarjoaa mahdollisuuden suorittaa ulkoisia komentoja, kun yhteys on muodostettu onnistuneesti. Ncat:a voidaan käyttää monenlaisiin tehtäviin, mukaan lukien verkkoviestinnän diagnosointi. Ncat voidaan asettaa välityspalvelimeksi, kun on analysoitava asiakkaan lähettämää liikennettä. Ncatin avulla voidaan analysoida vaihdettuja tiedoja ja tunnistaa mahdolliset virheet. Nmap Scripting Engine (NSE) on lisännyt tukea useille tietokantapalveluille viime vuosina. Tietoturvan analysoijat voivat automatisoida useita tehtäviä käsitellessään lukuisia tietokantapalvelimia. Voidaan suorittaa kysely, joka ilmoittaa sovelluksen tilasta. Toisaalta tietokantapalvelimen suojaaminen on tehtävä huolellisesti. Nmap auttaa myös käytettävissä olevien komentosarjojen avulla automatisoimaan yleisiä suojaustarkistustehtäviä, kuten tyhjien pääsalasanojen tarkistamista tai suojaamattomien kokoonpanojen havaitsemista.

Harjoitustehtävässä noudatettiin soveltuvin osin Packt Publishing Ltd julkaisemaa kirjaa "Nmap Network Exploration and Security Auditing Cookbook - Third EditionChapter"Kirjan kappaleissa 1–3 on kuvattu IoT-verkkojen skannaukseen käytettävät ohjelmat ja niiden vaatimat komentosarjat.

Linkki kirjan lukuun 1: Nmap Fundamentals.

Linkki kirjan lukuun 2: NGetting Familiar with Nmap’s Family.

Linkki kirjan lukuun 3: Network Scanning.

Opiskelijat kirjoittivat harjoitustehtävän suorittamisesta lyhyen raportin.

G Tunkeutumistestauksen valmistelu ja palvelunestohyökkäysdemonstraatio

Tehtävänä on suorittaa ns. Flooding attack, joka tunnetaan myös palvelunestohyökkäyksenä (DoS). Palvelunestohyökkäys kohdistuu kuljetuskerrokselle. Hyökkäyksessä hyökkääjät lähettävät järjestelmään erittäin suuren määrän liikennettä, johon hyökkäyksen kohteena oleva IoT-laite ei voi tutkia eikä siten pysty sallimaan sallittua verkkoliikennettä. Esimerkiksi Flooding-hyökkäys tapahtuu, kun järjestelmä vastaanottaa liian monta ping-komentoa ja sen on käytettävä kaikkia resurssejaan vastauskomentojen lähettämiseen. Kohteeksi otettiin IP-kamera, jossa on Web-palvelin. Web-palvelimella näytetään videokuvaa luokkatilasta. Hyökkäykseen käytettiin Kali Linuxia ja hyökkäyksen onnistuminen todennettiin toiselta tietokoneelta. IP-kameran IP-osoite selvitettiin verkkoskannauksella. Tehtävää jatkettiin siten, että koottiin Bottiarmeija, jota hyödyntämällä pyrittiin kaatamaan oppimisympäristössä olevan Web-palvelin. WEb-palvelimen kaatamiseen käytettiin Ping-komennon lähetystä. Ping-komento koodattiin scriptiin, jota suoritettiin tauotta. Opiskelijat kirjoittivat lyhyen kuvauksen hyökkäyksen toteutuksesta, jossa he pyrkivät analysoimaan kuinka paljon liikennettä ja palvelupyyntöjä tulee kohdistaa Web-serverille, että serveri ei enää pysty vastaamaan palvelypyyntöihin.

Verkko- ja porttiskannaus - Tunkeutumistestauksen valmistelu

Tässä tehtävässä perehdytään verkon tiedusteluun. Harjoituksen alussa käytetään nmap, znmap ja Wireshark ohjelmia.
Kali linuxin käyttö saattaa tuntua hankalalta oletuksena olevan "lontoo" näppäinasettelun johdosta. Tästä pääsee eroon nätisti asentamalla "suomi" näppäinlayoutin. Aja komentorivillä:

```
setxkbmap fi
```

Oppimiseen tarvitaan reflektointia, joten kirjoita koko ajan dokumentaatiota tekemisestä. Dokumentista on jatkossa hyvä palauttaa mieliin asioita nopeasti.

Taustaa - Porttiskannaus ja haavoittuvuustarkistus

Aloitetaan kartoittamalla verkko ja pikkuhiljaa pidetään edetään kohti avoimia portteja. Tärkeää on edetä oikeassa järjestyksessä: porttiskannaus, IoT-laitteen tietoliikenteen seuraaminen, mahdolisten webpalveluiden arviointi ja haavoittuvuuksien analysointi. Usein tämä vaihe kestää kauan, koska hyökkääjän tavoitteena on mahdollistaa mahdollisimman laaja pääsy kohdelaitteelle.

Käytetään KaliLinuxilta löytyvää **nmap**-työkalua, joka on laaja työkalu, jolla voi erinomaisen tehokkaasti skannata haavoittuvuuksia ja portteja.

Nmapin graafinen ohjelmaversio on **Zenmap**.

Nmap lyhyesti

Tutustu Nmap-ohjelmaan Nmap-verkkosivulla (https://nmap.org/book/man.html)

Lue erityisen tarkaan porttiskannaustekniikka (https://nmap.org/book/man-port-scanning-techniques.html) ja kirjoita lyhyt kuvaus oppimispäiväkirjaan oleellisimmista komennoista.

Tutustumisen arvoinen lähde on myös tämä: "What are port scan attacks and how can they be prevented? (https://www.techtarget.com/searchsecurity/answer/What-is-a-port-scan-attack)

Portit

Palauta mieleen mitä tarkoittavat portit. Tässä linkissä (https://www.cloudflare.com/learning/network-layer/what-is-a-computer-port/) on hyvin kuvattu oleellisimmat asiat. Kirjoita tärkeimmät IoT-järjestelmien käyttämien porttien kuvaukset.

Protokollat

Verkkoprotokollat on kuvattu täällä (https://www.w3schools.in/types-of-network-protocols-and-their-uses). Kirjoita oppimispäiväkirjaan IoT-järjestelmiin liittyvien oleellsimpien porttien tiivistelmät.

Nmap

Nmapista on hyötyä verkon ylläpitäjille, mutta monet ominaisuuksista on erittäin käyttökelpoisia myös haavoittuvuus- ja tunkeutumistestauksen valmistelussa.
Nmapin pääasiallinen käyttötarkoitus on porttiskannaus. Skannaus on mahdollista tehdä useaan porttiin yhdellä kerralla. Skannattavat IP- osoitteet voidaan määritellä joko nimipalvelimen avulla koneiden niminä, IP-osoitteina, ip-osoite alueina tai käyttämällä jokerimerkkejä. Nmap skannaa monta osoitetta yhtä aikaa, joten suurenkin verkon skannaus onnistuu nopeasti.

Skannaustekniikat

Nmap-porttiskannauksessa hyödynnetään TCP-protokollan ominaisuuksia. Nmap-ohjelmalle annetaan parametreina erilaisia TCP-paketteja, joiden avulla porttiskannauksia voidaan soveltaa halutulla tavalla.

Connect()

Tutustu Nmap-dokumentaatioon (https://nmap.org/book/man-port-scanning-techniques.html) ja opettele, miten connect-porttiskannaus toteutetaan.

TCP SYN - -sS eli (TCP SYN scan)

Tekniikkaa sanotaan puoli-avoimeksi skannaukseksi. Skannattavan koneen porttiin lähetetään eräänlainen TCP SYN-paketti. Jos vastaanottava kone kuuntelee porttia, saadaan vastauksena SYN/ACK-paketti ja jos ei niin vastauksena tulee RST-paketti.

TCP FIN, Xmas, NULL - -sN; -sF; -sX eli (TCP NULL, FIN, and Xmas scans)

Nykyään skannaus täytyy suorittaa palomuurin läpi tai halutaan suorittaa skannaus sitene, että siitä ei jää jälkiä logitiedostoon.
Nämä kolme tarkistustyyppiä ovat toiminnaltaan täsmälleen samat, lukuun ottamatta koetinpaketteihin asetettuja TCP-lippuja. Jos RST-paketti vastaanotetaan, portti katsotaan suljetuksi. Jos RST-pakettia ei oteta vastaan tarkoittaa sen, että portti on auki tai suodatettu. Portti on merkitty suodatetuksi, jos vastaanotetaan ICMP:n tavoittamaton virhe (tyyppi 3, koodi 0, 1, 2, 3, 9, 10 tai 13).

UDP - -sU (UDP scans)

UDP-skannauksessa saadaan tulokseksi kaikki avoimet UDP-portit. UDP-skannauksessa lähetetään nolla-tavuinen paketti.

ICMP/TCP-ping-skannaus

Perinteinen ping-skannaus selvittää tietokoneet, jotka ovat verkossa. Nykyisin tänä on lähes hyödyton, koska verkot estävät ICMP echo request -pakettit pois DoS- tai DDoS-hyökkäysten estämiseksi. Tätä voidaan kylläkin kiertää käytämällä TCP:tä pingaukseen.

Nmap ja käyttöjärjestelmän tunnistaminen

Source: Nmap OS detection (https://nmap.org/book/man-os-detection.html)
Yksi Nmapin tunnetuimmista ominaisuuksista on käyttöjärjestelmän etätunnistus TCP/IP-pinon sormenjälkien avulla. Nmap lähettää sarjan TCP- ja UDP-paketteja etäisännälle ja tutkii

käytännössä jokaisen bitin vastauksista. Suoritettuaan useita kymmeniä testejä, Nmap vertaa tuloksia **nmap-os-db-tietokantaan**, joka sisältää yli 2 600 tunnetun käyttöjärjestelmän sormenjälkeä. Nmap tulostaa käyttöjärjestelmän tiedot, jos osuma löytyy.
Tutustu Nmap-komentoihin ja tunnista IoT-oppimisympäristössä olevia käyttöjärjstelmiä. Kirjoita oppimispäiväkirjaan kuvaus läydöksistä.

IP-osoitteen väärentäminen eli IP Spoofing - -S <IP_Address> (Spoof source address)

Tutustu Nmap-dokumetaatiossa komentoihin Firewall/IDS Evasion and Spoofing (https://nmap.org/book/man-bypass-firewalls-ids.html).
Kokeile väärentää Raspi-klusterin 1. Raspin Ip-osoite.

Haavoittuvuuksien etsiminen

1. Ensimmäinen askel haavottuvaisuuksien etsimisessä on verkkoyhteyksien tutkiminen. Nettiosotteiden rajaaminen on tärkeää, ettei eksy väärille sivuille.

- -p parametrillä ohjataan mitä portteja, esim -p- tarkoittaa kaikki portit väliltä 1-65535 skannataan.
- tärkeä oppia analysoimaan syötettä, jotta tietää mitkä ovat oikeasti haavoittuvuuksia.
- nmap-diff hyvä työkalu, voi seurata muutoksia avoimissa porteissa.
- nmapilla voi hyvin spesifioida mitä tietoja haluaa, esimerkiksi mitä portit ja mitkä ip:t
- nmap pystyy tunnistamaan käyttöjärjestelmät
- nmapilla voi suoraan tallentaa tulokset tiedostoiksi, mikä helpottaa huomattavasti seurantaa ja työskentelyä
- open: hyväksyy TCP yhteydet, UDP paketit ja SCTP valituissa porteissa
- closed: portti on auki, mutta vastapuolella ei ole sovellusta tai ohjelmaa ottamassa mitään vastaan
- filtered: Nmap ei tiedä onko portti auki, on saatettu blokata palomuurilla
- -sS (TCP SYN scan): on salainen koska ei suorita TCP-yhteyttä. Näyttää closed, filtered ja open tilat
- -sT (TCP connect scan): suorittaa TCP-yhteyden. Ei pitäisi käyttää jos -sS on mahdollista. Helpompi havaita logeista
- -sU (UDP Scan): lähettää UDP paketteja portteihin. Hidas, vaikea tehdä nopeasti. Portit eivät lähetä niin herkästi vastakaikua.

Harjoitukset:

a) Miten nmap toimii?

Tämä osa harjoituksesta on kätevintä aluksi tehdä virtuaaliympäristössä.

Tee nmapilla alla olevat testit, sieppaa liikennettä wiresharkilla ja analysoi tuloksia.
Porttiskannaa omaa konetta verkossa ja harjoituskohteena olevia Raspi-klusterin Raspeja (192.168.11.108 ja 192.168.11.106)

Toteutus:
Avaa terminaali ja anna seuraavat käskyt:

```
sudo apt update
hostname -I
man nmap
```

Ota oman koneen ip talteen ja aloita nmapilla sen skannaamisen.

Avaa toinen terminaali ja laita sinne **wireshark** pyörimään:

```
sudo wireshark
```

Seuraavaksi aja nmap ja skannaa oman koneen ip.

```
sudo nmap -sT 192.168.12.2
```

Katso wiresharkista mitä paketteje nmap lähettää. Nmap:n pitäisi lähettää kahta pakettia (DNS- ja ARP-paketteja).

Tällä DNS tarkisti pyöriikö webissä osoitteessa 192.168.12.1 web-palvelin. Koska kyseessä on oma virtuaalikone, ei ainakaan pitäisi pyöriä, joten Wireshark:n pitäis vastata ettei pyöri.

ARP

```
arp -a
```

192.168.12.2 haluaa lähettää dataa, mutta ei tiedä MAC-osoitetta. Selvitetään ARP-paketilla kenelle pitäisi tieoa lähettää.

Seuraava Nmap ajo:

```
sudo nmap -sS 192.168.11.2
```

Tulokset olivat samanlaiset, koska 192.168.12.2 on palomuurin takana, eikä ole liikennettä ulkomaailmaan.

Komento

```
sudo nmap -sn 192.168.11.2
```

-sn komento ei tee porttiskannausta, etsitään vain avoinna olevat hostit. Tästä syystä terminaalissa ei lue porteista mitään. Paketit pysyivät tässäkin samanlaisina.

Komento

```
sudo nmap -Pn 192.168.11.2
```

Komento

```
sudo nmap -sV 192.168.11.2
```

-Pn komento ei tutki onko host elossa, eli ei pingaa hostia. Muuten komento tutkii kaikki halutut portit. Paketit ovat vastaavanlaisia kuin aiemmissakin

Komento

```
sudo nmap -sV 192.168.11.2
```

-sV tarkistaa mitä versioita sovelluksista tai raudasta hostilta löytyy. Tällä kertaa skannaus lähettää samat ARP-paketit kaksi kertaa.

b)Nmap:n toimintoja.

Tämä osa harjoituksesta tehdään USB LiveKali:lla.

Aja aluksi edelliset komennot joko osoitteeseen 192.168.11.106 tai 192.168.11.108.

Kun edelliset komennot on suoritettu ja tulokset analysointu wiresharkilla mennään eteenpäin.....

Tehdään ensin koko paikallisverkon skannaus. Se selvittää kaikki IoT-tietoturvaopetuksen paikallisverkossa olevat koneet, niiden IP- ja MAC-osoitteet.

```
hostaname -I
```

Laita oman koneen ip-muistiin esim(192.168.10.180)

```
sudo nmap -sP 192.168.10.180/24
```

Mitä Nmap löysi verkosta?

Käyttöjärjestelmän scannaus.

```
sudo nmap -O 192.168.11.106
```

Raspin osoite ja parametrina on Iso O, ei siis nolla.
Listaa mitä löysit, saiko nmap selville käyttöjärjestelmän? Mitä selviää Wiresharkista?

Graafinen Zenmap

Source: [How to Install Zenmap in Kali Linux 2022.1 without any Error (https://techdhee.in/install-zenmap-in-kali-linux/)](https://techdhee.in/install-zenmap-in-kali-linux/)

Joskus on hyödyllistä kartoittaa verkkotopologia.

KaliLinuxissa ei ole zenmapia, joten asennetaan sellainen. Asennus suoraan "home/kali" juureen

```
wget https://nmap.org/dist/zenmap-7.91-1.noarch.rpm
```

Ja lisäosia asennukseen:

```
wget http://archive.ubuntu.com/ubuntu/pool/universe/p/pygtk/python-gtk2_2.24.0-5.1ubuntu2_amd64.deb

wget http://archive.ubuntu.com/ubuntu/pool/universe/p/pycairo/python-cairo_1.16.2-2ubuntu2_amd64.deb

wget http://archive.ubuntu.com/ubuntu/pool/universe/p/pygobject-2/python-gobject-2_2.28.6-14ubuntu1_amd64.deb
```

Ja sitten suoritusoikeudet

```
sudo chmod +777 <package-name>
```

Asennus

```
sudo apt install ./python-cairo_1.16.2-2ubuntu2_amd64.deb
sudo apt install ./python-gobject-2_2.28.6-14ubuntu1_amd64.deb
sudo apt install ./python-gtk2_2.24.0-5.1ubuntu2_amd64.deb
```

Alien asennus

Alien on tietokoneohjelma, joka muuntaa eri Linux-pakettien jakelutiedostomuodot Debianiksi. Se tukee muuntamista Linux Standard Base-, RPM-, deb-, Stampede- (.slp) ja Slackware (tgz) -pakettien välillä.

```
sudo apt install alien
```

Konveroidaan .rpm .deb paketiksi:

```
sudo alien --to-deb zenmap-7.91-1.noarch.rpm
```

Lisätään oikeuksia paketille

```
sudo chmod +777 zenmap_7.91-2_all.deb
```

ja lopulta asennetaan zenmap

```
sudo apt install ./zenmap_7.91-2_all.deb
```

Käynnistä zenmap-ohjelma komennolla:

```
sudo zenmap -n nmap -sS 192.168.10.180
```

Ohjelman lopetus:

```
 ctrl + c
```

Chapter 12. Zenmap GUI Users' Guide
Se löytyy [täältä (https://nmap.org/book/zenmap.html)](https://nmap.org/book/zenmap.html).

c) Information Gathering

Tämä harjoitus tehdään USB KaliLinuxilla.

Avaa KaliLinuxin ohjelmavalkosta 01 - Imformation Gathering.
Tutustu käytössä oleviin työkaluihin, ja kirjoita jokaisesta lausen tai parin kommentti, josta käy ilmi se, mitä työkalulla tehdään. Konsultoi vapaasti googlea ;)

d) TelnetChat

Lataa githubista **telnet.py** KaliLinuxille. Yksinkertaisuuden vuoksi suoraan "home/kali" juureen.

```
git clone https://github.com/eerohuusko/iot_tietoturva.git
```

Samalla tulee muutakin, jotka voit deletoida.

Käynnistä chat-client (telnet.py) komennolla:

python3 telnet.py hostname port

```
python3 telnet.py 192.168.11.106 5000
```

Raspeilla 192.168.11.106 ja 192.168.11.108 on käynnissä identtiset chat-server palvelut, jotka kuuntelevat palvelupyyntöjä portissa 5000.

Viestit, jotka kirjoitetaan telnet.py terminaaliin "kaiutetaan" kaikille chattailijoille.

Tehtävänä on seurata telnet broadcast-liikennettä Wiresharkilla. Tehtävässä opetellaan filteröimään liikennettä, jotta viestien seuranta olisi yksinkertaisempaa.

Katso telnet liikenteen snifferöintiin liittyvä video aluksi.

Linkki [videoon (https://www.youtube.com/watch?v=AN5m7IbErg8)]

Tee lyhyt yhteenveto liikenteen tuloksista harjoitusdokumentaatioon. Analysoi telnet-turvallisuutta verrattuna ssh-turvallisuuteen. Kuvat edesauttavat oppimisessa.

e)IP ping-skannaus

Host(Isäntien) löytäminen IP-protokollan ping-skannauksilla

Nmap tukee IP-protokollan ping-skannaus tekniikkaa. Tällä yritetään määrittää, onko isäntä online-tilassa. Komento lähettää IP-paketteja käyttämällä eri protokollia.

Tässä tehtävässä on aiheena IP-protokollan ping-skannaukset.

Avaa terminaali ja kirjoita seuraava komento:

```
nmap -sn -PO 192.168.11.106
```

Kuinka se toimii...

-sn -PO-valinnat komentavat Nmapia suorittamaan IP-protokollan ping-skannauksen 192.168.11.106 isännästä.

Oletuksena tämä tekniikka käyttää IGMP-, IP-in-IP- ja ICMP-protokollia yrittääkseen määrittää, onko isäntä online-tilassa.

--packet-tracen käyttö näyttää enemmän yksityiskohtia skannaus tapahtumasta:

```
nmap -sn -PO --packet-trace 192.168.11.106
```

Tarkastele viestiä...

f) Ping-skannaus

IP-protokollan ping-skannausta voidaan muunnella muutamalla Nmap-vaihtoehdolla esimerkiksi, kuinka voidaan muuttaa käytettyä protokollaa, lisätä satunnaisia tietoja ja selvittää mitä protokollia tuetaan.

- Vaihtoehtoisten IP-protokollien asettaminen

Voidaan määrittää käytettävät IP-protokollat luettelemalla ne -PO-valinnan jälkeen. Esimerkiksi ICMP (protokollan numero 1), IGMP (protokollan numero 2) ja UDP (protokollan numero 17) protokollien käyttämiseksi voidaan käyttää seuraavaa komentoa:

```
nmap -sn -PO --data-length 100 <valitse jokin host>
```

Tuetut IP-protokollat:

```
TCP: Protocol number 6
UDP: Protocol number 17
ICMP: Protocol number 1
IGMP: Protocol number 2
IP-in-IP: Protocol number 4
SCTP: Protocol number 132
```

Yleisimpiä ICMP-protokollaa (Internet Control Message Protocol) käyttäviä työkaluja ovat ping ja traceroute. Yhteyttä testaava tietokone lähettää Echo Requestin ja kohde vastaa Echo Replyllä. DoS-hyokkäyksessä käytetään ICMP-protokollaa lähettämällä ns. Ping of Death -paketti, joka on väärin muotoiltu tai yleensö liian suuri ping-paketti. Tällä saadaan kohdekone, yleensä palvelin, kaatumaan. Voidaan myös lähettä ns. Ping Flood, jossa tiuhaan ja suurella määrällä ping-pyyntöjä halvaannutetaan verkkoresurssit. Tähän hyökkäykseen valjastetaan usein ns. Botti-armeija, jossa haavoittuneet IoT-laitteet lähettävät Ping Floot -hyökkäystä samaan kohteeseen.

g) Mac-osoite huijaus

MAC-osoitteen huijaus (MAC address spoofing)

MAC-huijaus väärentää yhteyksien alkuperää ja voi olla hyödyllistä tunkeutumisen havaitsemisjärjestelmien (IDS) kiertämisessä.
Seuraavalla tavalla on mahdollista huijata oman koneen MAC-osoitetta suoritettaessa ARP-ping-skannausta.

Aseta uusi MAC-osoite käyttämällä --spoof-mac lippua.

HOX!!!! Käytä vain opettajan antamaan tarkettiin ;)

```
nmap -sn -PR --spoof-mac <mac address> <target>
```

h) Broadcast skannaus

Palataanpa takaisin chattailemaan....

Broadcast pingit lähettävät ICMP-echo pyynnöt paikalliseen lähetysosoitteeseen, ja vaikka ne eivät toimisikaan koko ajan, ne ovat toimiva tapa löytää isäntiä verkosta lähettämättä kokeiluja (probes) suoraan isännille.

Kokeillaan kuinka löytää uusia isäntiä Nmap NSE:n avulla.

```
nmap --script broadcast-ping
```

Laittakaapa telnet-chat laulamaan ja wireshark auki....mitä huomaatte?

Kuinka se toimii...

Broadcast pingit toimivat lähettämällä ICMP-echo pyynnön paikalliseen Broadcast-osoitteeseen 255.255.255.255 ja odottamalla sitten isäntien vastausta ICMP-kaikuvastauksella.

Analysoidaanpa liikennettä --packet-trace-vaihtoehdon avulla. Wireshark auki ja muutama chatti liikkeell....

```
nmap --script broadcast-ping --packet-trace
```

Kokeilkaa mitä seuraava tekee.

```
nmap --script broadcast-ping --script-args=newtargets
```

i) IPv6 skannaus nmapilla

Yksi Nmapin tärkeimmistä päivityksistä on sen IPv6-tuki. Kaikki porttien skannaus- ja isäntähakutekniikat voivat ottaa IPv6-osoitteita, mukaan lukien käyttöjärjestelmän tunnistus, ja on jopa uusia mielenkiintoisia etsintätekniikoita, jotka käsittelevät IPv6-osoiteavaruuden brute force- skannauksen ongelmaa.

Tämä ohje kuvaa IPv6-osoitteen skannaamisen Nmapilla.

Kuinka tehdä se...

Avaa pääte ja kirjoita haluamasi Nmap-komento lisätoiminnolla -6:

```
nmap -6 <target>

nmap -6 -sT <target>
nmap -6 -O <target>
nmap -6 -A <target>
```

j) Performing IP address geolocation

Identifying the location of an IP address may help system administrators or threat intelligence analysts identify the origin of a network connection. Nmap ships with several NSE scripts that help us perform geolocation of a remote IP address: ip-geolocation-maxmind, ip-geolocation-ipinfodb, ip-geolocation-geoplugin, ip-geolocation-map-bing, ip-geolocation-map-google, and ip-geolocation-map-kml.

Example, but don't do that!!!!

```
nmap -sn --script ip-geolocation-* www.kamk.fi
Starting Nmap 7.92 ( https://nmap.org ) at 2021-11-18 13:09 EST
NSE: [ip-geolocation-maxmind] You must specify a Maxmind database file with the
maxmind_db argument.
NSE: [ip-geolocation-maxmind] Download the database from
http://dev.maxmind.com/geoip/legacy/geolite/
Nmap scan report for www.kamk.fi (212.116.36.47)
Host is up (0.049s latency).

Host script results:
| ip-geolocation-geoplugin: coordinates: 64.2361, 27.7396
|_location: Kainuu, Finland

Post-scan script results:
Bug in ip-geolocation-map-kml: no string output.
Bug in ip-geolocation-map-bing: no string output.
Bug in ip-geolocation-map-google: no string output.
Nmap done: 1 IP address (1 host up) scanned in 0.62 seconds
```

```
Try this out - > Set target www.hs.fi
```

k) Obtaining traceroute geolocation information

Nmap can map network paths by tracing the hops between the origin and destination. Geographical information can be useful when tracing events, and we can include it with Nmap's traceroute functionality with some help from the traceroute- geolocation NSE script.

```
nmap --traceroute --script traceroute-geolocation <target>
```

Example:

```
sudo nmap --traceroute --script traceroute-geolocation 192.168.10.101
```

Output should be something like that:

```
Starting Nmap 7.92 ( https://nmap.org ) at 2021-11-18 13:20 EST
Nmap scan report for 192.168.10.101
Host is up (0.00096s latency).
All 1000 scanned ports on 192.168.10.101 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Host script results:
| traceroute-geolocation:
|   HOP  RTT   ADDRESS         GEOLOCATION
|   1    0.42  192.168.12.1    - ,-
|_  2    0.88  192.168.10.101  - ,-

TRACEROUTE (using port 53/tcp)
HOP RTT     ADDRESS
1   0.42 ms 192.168.12.1
2   0.88 ms 192.168.10.101

Nmap done: 1 IP address (1 host up) scanned in 0.69 seconds
```

l) Querying Shodan to obtain target information

Shodan is one of the search engines for internet-connected devices. It is a useful source of information that even includes port and banner information of remote targets, among other bits of interesting data. One of the advantages of passively port scanning with Shodan is that we don't need to communicate directly with the target to obtain the list of open ports, protocols, and service banners.

The shodan-api NSE script needs an API key before it can be used. Shodan offers free developer API plans that you can obtain by signing up at https://developer.shodan.io/.

Once registered, copy your Shodan API key before continuing.

To obtain host information of a remote target from Shodan, use the following command:

```
nmap -sn -Pn -n --script shodan-api --script-args shodan-api.apikey=<ShodanAPI
KEY> <target>
```

The results will contain all the host information available in Shodan, including port number, protocol, production, and version information.

How it works...

With the previous command, we obtained the same information as if we were performed a port scan with the version detection engine (-sV) enabled without directly communicating with the target at any point. ShodanHQ (https://www.shodan.io/) scans the internet regularly to gather port and service information and probes for important and common vulnerabilities. During security assessments, this is valuable information as the results are instantaneous. Hence, it is a great option with which to start our reconnaissance tasks while we perform more comprehensive scans.

Specifying a single target

Use the **shodan-api.target** script argument to set a single target to be queried from the database. Remember to use an IP address as the target since we are disabling DNS resolution (-n):

```
nmap -sn -Pn -n --script shodan-api --script-args shodan-api.apikey='<ShodanAPI
KEY>',shodan-api.target=<IP target>
```

That's all folks today!!!!! Happy Hacking weekend!

H IP-kameran uhkamallinnus ja penetraatiotestausharjoitus

Taustateoriaa ja ohjeistusta IP-kameraan kohdistuvan uhkamallinnusprosessin ja penetraatuotestauksen toteuttamiseen. Uhmallinnuksessa noudatetaan Digi- ja väestötietoviraston julkaisemaa Turvallisen sovelluskehityksen käsikirjan ohjeistusta [14]. Tehtävän kuvaus käsikirjan sivuilla 49-54. Tehtävässä piirrettiin tietovuokaavio (data flow diagram, DFD) ja viestisekvenssikaavio (*engl. message sequence chart, MSC*). Kaavioiden piirtämisen jälkeen jokainen piirretty tietovuo, sekvenssikaavio ja tietovarasto käsiteltiin STRIDE-menetelmän kohtien 1–6 mukaisesti.

STRIDE-menetelmä on kuvattu Digi- ja väestötietoviraston julkaisemassa Turvallisen sovelluskehityksen käsikirjassa sivulla 49. Linkki käsikirjaan löytyy osoitteesta: Turvallisen sovelluskehityksen käsikirja.

Seuraavassa osassa tehtävää on tavoitteena toteuttaa penetraatiotestaus, jotka kohdistuvat löydettyihin uhkiin, joiden toteutuessa on riski esimerkiksi tietomurtoihin tai tietojen kalasteluun. Penetraatiotestaus suoritetaan noudattamalla TestingIoT-Security pdf–dokumentissa kuvattuja ohjeita [2]. Harjoitustehtävän pohjana oleva konferenssijulkaisu löytyy osoitteesta: Testing IoT Security: The Case Study of an IP Camera.

Lopuksi opiskelijatiimin tulee kirjoitaa lyhyt essee, jossa analysoidaan tehtävää ja omaa oppimista. Esseessä tulee olla maininnat merkittävistä löydöksistä, joita penetraatiotestauksessa havaittiin.