Ilona Nurminen

Proxmox-virtuaalipalvelinympäristö Jyväskylän yliopiston kyberturvallisuuden kursseja varten

Tietotekniikan pro gradu-tutkielma

15. kesäkuuta 2021

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Ilona Nurminen

Yhteystiedot: ilo@satudisketti.net

Ohjaajat: Panu Moilanen ja Tapani Tarvainen

Työn nimi: Proxmox-virtuaalipalvelinympäristö Jyväskylän yliopiston kyberturvallisuuden kursseja varten

Title in English: Proxmox virtual server environment for cyber security courses at University of Jyväskylä

Työ: Pro gradu-tutkielma

Opintosuunta: tietotekniikka

Sivumäärä: 82+0

Tiivistelmä: Laajan teknisen kyberturvallisuusharjoituksen järjestäminen vaatii runsaasti asiantuntijoiden aikaa ja resursseja. On siis syytä selvittää, millä tavoin tällaisen harjoituksen toteuttaminen voitaisiin saada yksinkertaisemmaksi, edullisemmaksi ja joustavammaksi. Tämän tutkielman tavoitteena on selvittää, voidaanko Jyväskylän yliopistolla järjestettävä Kyberhyökkäys ja sen torjunta -kurssi (ITKST55) eli pienimuotoinen kyberturvallisuusharjoitus järjestää virtualisoidulla palvelinjärjestelmällä. Erityisen kiinnostuneita ollaan mahdollisuudesta virtualisoida suuri osa harjoituksessa käytettävästä infrastruktuurista yhden laitekokonaisuuden sisään, ns. hyperkonvergoidusta infrastruktuurista. Tutkielma käy läpi virtualisointiin ja kyberturvallisuusharjoituksiin liittyvän olennaisen käsitteistön ja tarkastelee kyberturvallisuuden opetuskäytön arkkitehtuuri- ja infrastruktuurivalintoja konstruktiivisen tutkimuksen keinoin. Tutkimuksen empiirisessä osuudessa määritellään kurssin tavoitteiden mukaiset vaatimukset virtuaalipalvelinjärjestelmälle ja toteutetaan konstruktiivisen tutkimusmenetelmän mukainen artefakti eli prototyyppi virtuaalipalvelinjärjestelmästä käytettäväksi vuoden 2019 ITKST55-kurssilla. Prototyyppi rakennetaan Proxmox Virtual Environment-käyttöjärjestelmäjakelun avulla, ja asetettujen vaatimuksien toteutuminen arvioidaan vuoden 2019 kurssitoteuman valossa. Tutkimuksessa havaitaan, että virtuaalipalvelinjärjestelmä soveltuu kyberturvallisuusharjoituksen järjestämiseen prototyypin perusteella.

Lisäksi saadaan tietoa Proxmox VE-käyttöjärjestelmäjakelun ja hyperkonvergoidun infrastruktuurin periaatteiden soveltuvuudesta kyberturvallisuusharjoituksen järjestämiseen.

Avainsanat: Proxmox, kyberharjoitus, virtualisointi, kyberturvallisuus

Abstract: Organizing a technical cyber security exercise requires a tremendous amount of subject-matter expert time and resources. Attempting to streamline this process into a less complex and expensive form is thus worth the effort. This thesis aims to determine whether the "Network attack and its countermeasures" (ITKST55) course, a small-scale cyber security exercise in itself, is feasible to implement using a virtualized platform. Of particular interest is virtualizing as much of the required infrastructure into a single managed device in the manner of hyper-converged infrastructure. Key concepts related to virtualization and cyber security exercises are explored by the means of design science. In the empirical chapter requirements for the virtualized platform are determined from the goals of the course. These requirements are then implemented into the design science artefact, a prototype of a full-scale cyber exercise virtualization platform. This platform is then evaluated in use during the 2019 iteration of ITKST55. The prototype is built with tools available in the Proxmox Virtual Environment virtualization software distribution. The research carried out in the thesis posits that a virtualized platform is sufficient for implementing a cyber security exercise as proven by the prototype. Additionally, the thesis provides data on the application of Proxmox VE and hyperconverged infrastructure concepts in cybersecurity exercises.

Keywords: Proxmox, cyber exercise, virtualization, cyber security

Esipuhe

Lämmin kiitos kaikille graduprosessiin ja Proomu-projektiin osallistuneille.

Suosittelen lukijalle lasillista portugalilaista vihreää viiniä tai sitruunasoodaa.

Levysuositukseksi Courtney Barnett: Sometimes I Sit and Think, and Sometimes I Just Sit.

Jyväskylässä 15. kesäkuuta 2021

Ilona Nurminen

Kuviot

Sisältö

1 Johdanto

Todellisia tuotantoympäristöjä jäljittelevien tietojärjestelmien rakentaminen opetuskäyttöä varten on vaikeaa, työlästä ja kallista. Useiden aihepiirien, kuten kyberturvallisuuden kohdalla kuitenkaan pelkkä teoreettinen tutustuminen aiheeseen ei riitä, vaan käytännön harjoitustilanteita on järjestettävä. Jyväskylän yliopistolla järjestettävä kurssi ITKST55 Kyberhyökkäys ja sen torjunta on pienimuotoinen kyberturvallisuusharjoitus, jossa kurssilaiset saavat käyttöönsä tyypillistä yritysverkkoa jäljittelevän tietojärjestelmän palvelimineen ja työasemineen, jota heidän tulee suojata kyberhyökkäykseltä.

Aikaisempina vuosina Kyberhyökkäys ja sen torjunta-kurssi on järjestetty *bare metal* -palvelimista, -työasemista ja -verkoista koostuvassa ympäristössä. Tällainen ympäristö on kuitenkin hankala rakentaa ja ylläpitää, ja se on myös joustamaton: arkkitehtuurimuutokset vaativat suuren työpanoksen, ja niiden teko kurssin aikana on joko haastavaa tai mahdotonta. Kurssilla saatetaan esimerkiksi havaita, että tietyn aiheen käsittelyyn toivottaisiin enemmän seikkaperäisiä käytännön esimerkkitapauksia, mutta niiden toteutus ei ole mahdollista kurssin laitteisiin asennetun käyttöjärjestelmäversion puitteissa. Lisäksi laitteistossa tai ohjelmistossa ilmenevät tekniset ongelmat saattavat vaikeuttaa tai hidastaa kurssin etenemistä, ja varalaitteiston ylläpitäminen ei välttämättä ole aika- ja taloudellisten resurssien puitteissa mielekästä.

Ratkaisuksi esitetään virtualisoitua kurssiympäristöä. Virtualisoidussa ympäristössä kurssin vaatimat palvelimet, verkkolaitteet ja työasemat on toteutettu ohjelmistopohjaisesti sikäli, kun se on mahdollista. Virtuaalisia palvelimia ja työasemia käytetään joko fyysisiltä työasemilta suoraan kuten perinteisiä ei-virtuaalisia laitteita, tai niihin luodaan konsoliyhteys virtuaalikoneiden hallinta-alustasta. Toteutusalustaksi valittu Proxmox Virtual Environment on avoimen lähdekoodin Linux-pohjainen virtualisointialusta, jonka avulla on mahdollista muodostaa toiminnallisesti aitoa laitteistoa jäljitteleviä palvelimien, työasemien ja verkkolaitteiden kokonaisuuksia (*Proxmox VE Administration Guide* 2019).

Tässä tutkielmassa perehdytään virtualisoitujen järjestelmien pääkomponentteihin ja selvitetään, miten niitä voidaan hyödyntää kyberturvallisuustaitojen harjoittelussa ja opetukses-

sa. Lisäksi perehdytään kyberturvallisuusharjoituksen erityispiirteisiin ja niiden yhteensovittamiseen virtuaalisen kurssiympäristön kanssa. Tämän taustatutkimuksen pohjalta luodaan suunnitelma virtualisoidun kurssiympäristön rakentamisesta ja arvioidaan samalla, onko Proxmox alustana soveltuva tähän tarkoitukseen. Suunnitelma toteutettiin käytännössä rakentamalla ITKST55-kurssin tarpeita vastaava ympäristö syksyn 2019 kurssia varten. Kurssitoteuman perusteella tutkielmassa arvioidaan, täyttikö rakennettu ympäristö sille asetetut tavoitteet ja millaisia jatkotutkimusmahdollisuuksia aihepiiri mahdollisesti tarjoaa.

Tutkielma lähestyy aihetta suunnittelutieteellisenä tutkimuksena. Suunnittelutieteellinen tutkimus pyrkii luomaan uusia, innovatiivisia artefakteja, joilla ratkaistaan tärkeitä ja liiketoiminnallisesti merkittäviä ongelmia (Von Alan ym. 2004). Tässä tutkimuksessa syntyy artefakti virtuaalisesta kurssiympäristöstä käsitteenä, ja tämän artefaktin toteumana syksyn 2019 ITKST55-kurssille rakennettu virtuaalinen kurssiympäristö. Kurssin tavoitteista muodostetaan vaatimukset, jotka artefaktin täytyy täyttää ollakseen onnistunut. Vaatimusten toteutumista evaluoidaan kurssin aikana tehdyn havainnoinnin perusteella, ja havainnoista muodostetaan mahdollisia jatkokehityspolkuja ja uusia tutkimusongelmia.

Tutkielmassa käsitellään ensin virtuaalipalvelinympäristön olennaisimmat osat käsitteellisellä tasolla ja esitellään lyhyesti Proxmox Virtual Environment:in ominaisuudet. Sen jälkeen tutustutaan lyhyesti kyberturvallisuusharjoituksiin ja niiden teknisiin erityisvaatimuksiin. Seuraavaksi esitellään kirjallisuuden pohjalta joitakin esimerkkejä virtualisoitujen järjestelmien opetuskäytöstä, painottuen erityisesti kyberturvallisuuden opetuskäyttöön. Tämän jälkeen esitellään tarkemmin tutkimuksessa käytettävä tutkimusmenetelmä ja sen soveltaminen tutkielman tarpeisiin. Tältä pohjalta siirrytään tutkimuksen varsinaiseen empiiriseen osuuteen, jossa muodostetaan ITKST55-kurssin virtuaaliselle kurssiympäristölle asetettavat vaatimukset. Sen jälkeen kuvataan virtuaalisen kurssiympäristön tekninen toteutus näiden vaatimusten pohjalta. Lopuksi kurssia varten luotu virtuaalipalvelinjärjestelmän tekninen toteutus evaluoidaan vaatimusten toteutumisen perusteella, ja toteutuksen onnistumista pohditaan niin kurssin aikana tehtyjen havaintojen kuin aiemman tutkimuskirjallisuudenkin kautta.

1.1 Tutkimuksen tavoitteet

Tässä alaluvussa käydään läpi, minkälaisiin kysymyksiin tutkimuksella pyritään vastaamaan ja määritellään tutkimuksen reunaehdot. Lisäksi käydään läpi tutkimuksen toteutusta sekä arvioidaan lyhyesti tutkimukselta odotettuja tuloksia ja pohditaan niiden sovellettavuutta.

1.1.1 Aihepiirin rajaus ja tutkimusongelma

Tutkimuksen päätavoitteena on selvittää, onko ITKST55-kurssin tarpeisiin soveltuva kurssiympäristö mahdollista toteuttaa virtualisointiteknologiaa hyödyntäen. Lisäksi virtualisoinnin käsite laajennetaan koskemaan laajemmalti hyperkonvergoituja laitteita, joissa niin tietojenkäsittely, verkkolaitteet ja tallennustila toteutetaan ohjelmallisesti samalla alustalla (*Proxmox VE Administration Guide* 2019). Tutkimuksessa pohditaan, ovatko nämä hyperkonvergoidun infrastruktuurin periaatteet sovellettavissa esimerkiksi ITKST55-kurssin vaatimusten mukaiseen kurssiympäristöön, ja miten ne näkyvät toteutetussa Proxmox VE-pohjaisessa kurssiympäristössä. Nämä tavoitteet voidaan purkaa seuraavanlaisiksi tutkimuskysymyksiksi:

- Onko ITKST55-kurssi mahdollista järjestää virtualisoidussa kurssiympäristössä?
- Soveltuuko Proxmox virtualisointialustana ITKST55-kurssin toteutukseen?
- Onko hyperkonvergoidun infrastruktuurin periaatteista hyötyä kyberturvallisuusharjoituksen järjestämisessä?

Tutkimuksen painopiste on suunnitellun virtuaalipalvelinympäristön teknisen toteutuksen kuvaamisessa painottuen erityisesti Proxmox VE:n verkkolaitekonfiguraation toteutukseen. Tämä lähestymistapa on perusteltu, sillä Proxmox VE:n käytöstä kyberturvallisuusharjoituksen vaatiman monimutkaisen verkkoinfrastruktuurin rakentamiseen ei ole aiempaa tutkimusta. Tutkimus ei kuitenkaan vertaile Proxmox-käyttöjärjestelmäjakelua muihin vastaaviin virtualisointiohjelmistoihin, eikä ota kvantitatiivisesti mitaten kantaa järjestelmän suorituskykyyn. Tutkimus ei myöskään ota kantaa kurssin oppimistuloksiin.

1.1.2 Tutkimuksen toteutus

Tutkimus toteutetaan suunnittelutieteellisenä tutkimuksena. Suunnittelutieteellinen tutkimus on luonteeltaan soveltavaa tutkimusta, joka pyrkii tuottamaan uutta suunnittelutietämystä eli tietämystä, jota alan ammattilaiset voivat käyttää ratkaistessaan erilaisia suunnittelu- ja konstruointiongelmia (Järvinen 2000; Aken 2004). Esimerkiksi lääkäri tarvitsee potilaan hoitoa suunnitellessaan tietoa erilaisista hoitomuodoista, jotta hän voi rakentaa juuri tietylle potilaalle soveltuvan hoitosuunnitelman. Tutkimuksessa syntyvä suunnittelutietämys ei siis ole suora resepti tai algoritmi yksittäisen esimerkkitapauksen toistamiseen, vaan yleistä tietämystä, jota voidaan soveltaa joukkoon samankaltaisia tapauksia (Järvinen 2000).

Von Alan ym. (2004) esittävät seuraavat seitsemän ohjetta sovellettaviksi suunnittelutieteellisen tutkimuksen toteutukseen:

- Tutkimuksessa on suunniteltava artefakti.
- Artefaktin on ratkaistava tärkeä ja liiketoiminnallisesti merkittävä ongelma.
- Artefaktin hyödyllisyys ja laatu on osoitettava evaluoimalla se.
- Tutkimuksen on tuotettava uutta tietoa, uusia menetelmiä tai huomattava artefakti.
- Tutkimuksen on oltava tieteellisesti tarkka artefaktin rakentamisen ja arvioinnin suhteen.
- Tutkimus on nähtävä iteratiivisena prosessina.
- Tutkimuksen tulosten raportoinnissa on huomioitava niin tutkija- kuin soveltajayleisöt.

Tässä tutkimuksessa suunniteltava artefakti on virtuaalinen kurssiympäristö, jota evaluoidaan toteuttamalla se käytännössä. Konstruktio pyrkii vastaamaan selkeisiin reaalimaailman tarpeisiin: on rakennettava edullisempi, joustavampi ja vähätöisempi vaihtoehto aidoista laitteista rakennetulle harjoitteluympäristölle. On tärkeää huomata, että konstruktion käytännön toteutus eli vuoden 2019 ITKST55-kurssia varten rakennettu virtuaalinen kurssiympäristö, on vain yksi mahdollinen esimerkkitulkinta konstruktiosta. Artefaktin laatu ja hyödyllisyys osoitetaan vertaamalla virtuaalisen kurssiympäristön toteutusta ITKST55-kurssin asettamiin vaatimuksiin. Tutkimuksessa pyritään kehittämään virtuaalisten kurssiympäristöjen rakentamiseen liittyvää suunnittelutietämystä eteenpäin ja muotoilemaan se niin, että siitä on hyötyä mahdollisemman laajalle yleisölle: niin aihepiiriä tieteellisestä näkökulmasta lähestyvil-

le kuin niillekin, jotka mahdollisesti rakentavat vastaavia konstruktioiden toteutuksia käytännössä.

1.1.3 Tutkimukselta odotetut tulokset ja niiden merkitys

Tutkimuksen tavoitteena on ensisijaisesti saada tietoa siitä, soveltuuko Proxmox virtualisointialustana kyberturvallisuuden opetuskäyttöön Jyväskylän yliopistossa, ja onko se mahdollisesti sopiva myös muiden IT-tiedekunnan kurssien tarpeisiin. Konstruktion toteumalla halutaan osoittaa, että virtuaalisten ympäristöjen käyttö osana kyberturvallisuuden opetusta on mahdollista ja mielekästä. Lisäksi saadaan luotua määritelmä hyperkonvergoidulle infrastruktuurille ja tietoa sen mahdollisesta soveltamisesta kyberturvallisuuden opetuskäyttöön.

2 Teknologia

Tässä luvussa esitellään virtualisoidun kurssiympäristön suunnittelun ja toteutuksen kannalta olennaisin käsitteistö. Ensiksi selvitetään, mitä virtualisoinnilla ylipäätään tarkoitetaan, jonka jälkeen virtualisoinnin käsite laajennetaan osaksi hyperkonvergoidun infrastruktuurin periaatetta. Sen jälkeen tutustutaan Proxmox Virtual Environment-käyttöjärjestelmäjakeluun ja sen ominaisuuksiin.

2.1 Virtualisointi

Virtualisoinnilla tarkoitetaan tietotekniikassa useimmiten jonkin fyysisen komponentin abstrahoimista loogiseksi objektiksi, jotta sen tarjoamia resursseja voidaan käyttää tehokkaammin hyödyksi (Portnoy 2016). Tällä voidaan tarkoittaa esimerkiksi kokonaisten tietokonejärjestelmien tai niiden yksittäisten komponenttien, kuten verkko- tai tallennuslaitteiden toteuttamista ohjelmallisesti fyysisen laitteiston päälle. Tässä tutkielmassa virtualisointia käsitellään erityisesti palvelin- ja työpöytäkoneiden ja niiden tarvitseman virtuaalisen verkkoinfrastruktuurin näkökulmasta, eli ns. alustavirtualisoinnin ja sen edellytysten kannalta.

2.1.1 Hypervisor eli virtuaalikonemonitori (VMM)

Hypervisor eli virtuaalikonemonitori (VMM) on ohjelmisto, joka toteuttaa ajoympäristön virtuaalikoneita varten ja hallinnoi niiden tarvitsemia resursseja. Nykyisiä yleisessä käytössä olevia virtuaalikonemonitoreja ovat esimerkiksi VMWare ESXi, Xen, Microsoft Hyper-V ja KVM (Manik ja Arora 2016.) Hypervisor-ohjelmistot voidaan jakaa kahteen pääkategoriaan niiden laitteistosuhteen mukaan. Nk. **tyyppi 1/bare metal** -hypervisor asennetaan suoraan laitteistolle, kun taas **tyyppi 2/hosted** -hypervisor asennetaan laitteistolle asennetun käyttöjärjestelmän päälle. Xen, Microsoft Hyper-V ja VMWare ESXI edustavat tyypin 1 hypervisoreita, kun taas VMWare Workstation- ja Oracle VirtualBox -ohjelmistot ovat tyyppiä 2. KVM puolestaan on Linux-ydinmoduuli, joka tarjoaa laitteistopohjaiset virtualisointiominaisuudet tyypin 1 hypervisorin tavoin ja emuloi muita laitteita käyttöjärjestelmätasolla QEMU-emulaattoria käyttäen, kuten tyypin 1 hypervisorit. (Manik ja Arora 2016; Eder

2016.)

Popek ja Goldberg (1974) määrittelivät vuonna 1974 julkaistussa artikkelissaan *Formal requirements for virtualizable third generation architectures* virtuaalikoneen ja virtuaalikonemonitorin (VMM) käsitteet. Virtuaalikone on tehokas, eristetty kopio aidosta tietokoneesta. Virtuaalikonemonitori puolestaan on ohjelmisto, joka mahdollistaa virtuaalikoneiden ajamisen. (Popek ja Goldberg 1974.) Virtuaalikonemonitorin tulee Popek ja Goldberg (1974) mukaan toteuttaa kolme vaatimusta:

- virtuaalikoneelle luotavan ympäristön on oltava olennaisilta osin identtinen aidon tietokoneen ympäristön kanssa,
- virtuaalikoneen suorituskyvyn tulee olla samankaltainen kuin aidon tietokoneen ja
- virtuaalikonemonitorin on pystyttävä hallitsemaan kaikkia järjestelmäresursseja. Virtuaalikoneet eivät saa päästä käyttämään muita kuin niille varattuja resursseja, ja virtuaalikonemonitorin tulee tietyissä olosuhteissa päästä ottamaan varatut resurssit hallintaansa.

Nämä määritelmät ovat yhä voimassa nykyisistä hypervisoreista puhuttaessa ja myöskin olennaisia tämän tutkielman tavoitteiden kannalta. Koska tavoitteena on luoda opiskelijoille mahdollisimman autenttinen, joskin pienimuotoinen “yritysverkko”, jossa palvelimet, työasemat ja verkkolaitteet toteuttavat tyypillisiä tehtäviään, on identtisyys aitojen laitteiden kanssa tärkeää. Jotta kurssiympäristössä harjoittelu olisi opiskelijoille mielekästä, ei järjestelmän suorituskyky saa myöskään jäädä merkittävästi jälkeen aidosta laitteistosta. Tämä korostuu erityisesti virtuaalisten työpöytäkoneiden käytössä, jossa pienetkin viiveet esimerkiksi hiiren kursorin liikkeissä voivat aiheuttaa turhautumista käyttäjälle. Virtuaalikonemonitorin täyden hallinnan vaatimus on myös olennainen turvallisuuden ja suorituskyvyn takia. Kyberhyökkäykseltä suojautumista harjoitellessa itse hyökkäysverkkoliikenteen tulee olla tehokkaasti kontrolloitavissa, jotta se voidaan tarvittaessa eristää ja pysäyttää.

2.1.2 Alustavirtualisointi

Alustavirtualisointi voidaan jakaa kolmeen eri pääkategoriaan: **emulaatioon/täysvirtualisointiin**, **paravirtualisointiin** ja **käyttöjärjestelmävirtualisointiin**.

Nämä kategoriat ovat melko löyhästi määriteltyjä ja monet nykyaikaiset virtualisointiratkaisut sisältävät ominaisuuksia/toimintoja useista kategorioista. Emulaatiolla tarkoitetaan konekielisten käskyjen useimmiten ajonaikaista kääntämistä arkkitehtuurista toiselle (kutsutaan myös nimellä *dynaaminen uudelleenkäännös, dynamic recompilation* (Stewart, Humphries ja Andel 2009). Emuloinnin avulla voidaan esimerkiksi ajaa ARM64-prosessoriarkkitehtuurille käännettyä koodia AMD64-pohjaisella alustalla.

Jos taas halutaan virtualisoida jokin alusta niin, että se pääsee käyttämään sille varattuja laitteistoresursseja suoraan ilman muutoksia käyttöjärjestelmään, puhutaan täysvirtualisoinnista. Virtuaalikonemonitorin tehtävänä on eristää nämä laitteistoresurssit ja jakaa ne vierasalustojen kesken. Täysvirtualisoitu alusta käyttäytyy siis kuin suoraan laitteistolle asennettu *bare metal*-alusta. Paravirtualisointi puolestaan tarkoittaa tekniikkaa, jossa virtualisoitava alusta on tietoinen siitä, että sitä ajetaan virtualisoituna: virtuaalikonemonitori muodostaa laitteiston ja virtualisoitavan alustan väliin kerroksen, joka kääntää alustan virtuaalikonemonitorille esittämät järjestelmäkutsut laitteistolle sopiviksi. Tämä tarkoittaa sitä, että virtualisoitava käyttöjärjestelmä on muokattava virtualisointia tukevaksi. (Babu ym. 2014.)

Käyttöjärjestelmävirtualisoinnilla tarkoitetaan virtualisointitekniikkaa, jossa useita toisistaan eristettyjä vieraskäyttöjärjestelmäinstansseja voidaan ajaa samaa käyttöjärjestelmäydintä käyttäen yhtäaikaisesti (Stewart, Humphries ja Andel 2009). Tämä lähestymistapa soveltuu erityisesti tilanteisiin, joissa vieraskäyttöjärjestelmillä ajetaan kevyempiä sovelluksia, eikä niistä jokainen tarvitse omaa käyttöjärjestelmäydintään. Käyttöjärjestelmävirtualisointia edustavat esimerkiksi Linux LXD ja Docker. (Plauth, Feinbube ja Polze 2017.)

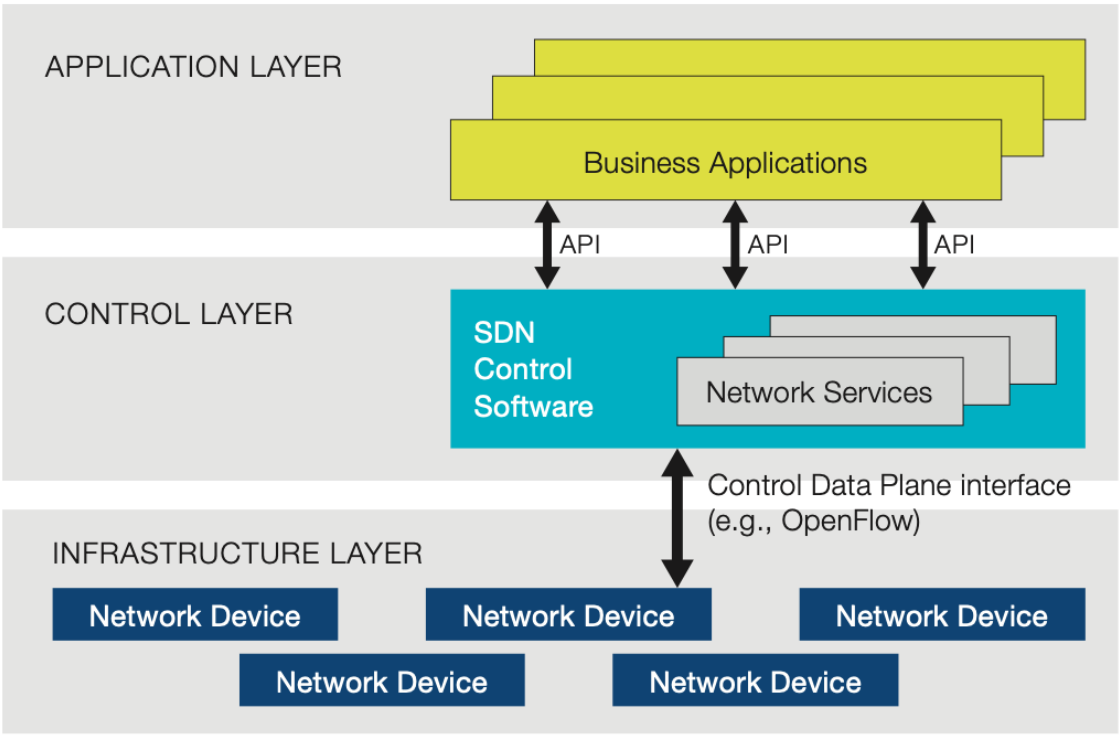
2.2 Hyperkonvergoitu infrastruktuuri (hyperconvergent infrastructure, HCI)

Hyperkonvergoidulla infrastruktuurilla tarkoitetaan ohjelmiston ja laitteiston kokonaisuutta, jossa tietojärjestelmän laskennalliset komponentit sekä verkko- ja tallennustilaresurssit toteutetaan samalla palvelinlaitteistolla ohjelmallisesti. Perinteisesti näistä komponenteista ovat vastanneet eri laitteisto- ja ohjelmistovalmistajien laitteet, joilla on usein omat erikoistuneet ylläpitäjänsä tai ylläpitäjätiiminsä. (Haag 2016.) Hyperkonvergoidussa ympäristössä

pyritään kytkemään nämä tietojärjestelmän osat toisiinsa niin tiiviisti, että niiden hallitseminen onnistuu keskitetysti yhteisten käyttöliittymien tai rajapintojen avulla. Hyperkonvergoitu infrastruktuuri soveltuu myös toteutettavaksi ns. pilvihybridimallina, jossa osa tietojärjestelmästä toteutetaan paikallisesti hyperkonvergoiduilla laitteilla, osa taas pilvipalveluita hyödyntäen. (Haag 2016.) Proxmox VE virtualisointiympäristönä mahdollistaa HCI:n toteuttamisen: se sisältää tietojenkäsittelyn, tallennustilan ja verkkoinfrastruktuurin virtualisoinnin ohjelmallisesti toteutettuina komponentteina (*Proxmox VE Administration Guide* 2019).

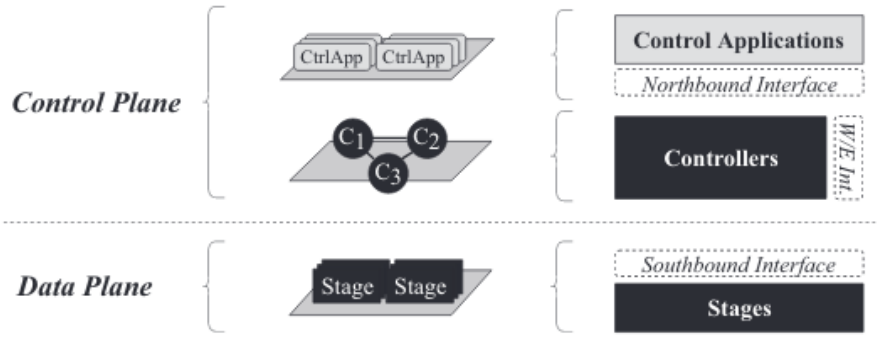
Hyperkonvergoidun infrastruktuurin “perusyksikkö"on **ohjelmistopohjainen palvelinkeskus** (*software-defined data center, SDD*). Ohjelmistopohjaisella palvelinkeskuksella tarkoitetaan sellaista palvelinkeskusta, jossa kaikki IT-infrastruktuurin perinteiset osat (laskennalliset resurssit, tallennustila, hallinta ja verkko) on toteutettu virtuaalisesti ja mahdollisimman automatisoidusti (Haag 2016). Hyperkonvergoitu infrastruktuuri pyrkii siis mahdollistamaan ohjelmistopohjaisen palvelinkeskustoteutuksen. Tavoitteena on luoda palvelinkeskus, jota voidaan hallita keskitetysti, jonka komponentteja voidaan ohjelmoida vastaamaan organisaation tarpeita ja jonka tarvitsemia resursseja voidaan skaalata tarpeen mukaan.

Ohjelmistopohjainen verkko (*software-defined networking, SDN* kuvassa 1) on verkkototeutusparadigma, jossa verkon hallintaosuus erotetaan sen toiminnallisesta paketinvälityksestä omaksi kerroksekseen (ONF 2012). Verkon hallinta ja konfigurointi toteutetaan ohjelmistopohjaisesti kontrollikerroksella (*control plane/control layer* kuvassa 1), joka järjestää tiedonvälityksen varsinaisten fyysisten verkkolaitteiden ja verkkoa käyttävien liiketoimintaohjelmistokerroksen (*application layer*) välillä. Kontrollikerros ohjaa datakerrosta (*data plane/infrastructure layer* kuvassa 1), joka suorittaa varsinaisen paketinvälityksen. Kontrollikerrosta voidaan ohjelmoida sovellustasolta sovellusrajapintoja käyttäen, jolloin verkkolaitteen toimintaa voidaan muokata dynaamisesti. (ONF 2012 ja Jain ja Paul 2013.) Proxmox VE mahdollistaa SDN-paradigman mukaisen verkkolaitetoteutuksen Open vSwitch -virtuaalikytkimien avulla mm. OpenFlow-rajapintaa käyttäen (*Proxmox VE Administration Guide* 2019). Tämä mahdollistaa Proxmox VE -alustalla toteutettujen virtuaalikoneiden verkkolaitteiden yhdistämisen organisaation muuhun verkkoinfrastruktuuriin ja mahdollistaa niiden hallitsemisen ohjelmallisesti.



Kuvio 1. Ohjelmistopohjaisen verkkoinfrastruktuurin osat (ONF 2012)

Ohjelmistopohjainen tallennustila (*software-defined storage, SDS* kuvassa 2) on ohjelmistopohjaisiin verkkoihin käsitteellisesti verrattavissa oleva tallennustilan toteutusparadigma, jossa tallennustila toteutetaan ohjelmallisesti niin, että tallennustilan hallitseva kontrollikerros ja datan I/O-toiminnallisuudesta huolehtiva datakerros erotetaan toisistaan. (Macedo ym. 2020.) Proxmox VE mahdollistaa SDS-paradigman mukaisen tallennustilan toteutuksen Ceph-tiedostojärjestelmää ja hallintatyökaluja käytettäessä (*Proxmox Virtual Environment 6.2 - Datasheet* 2020).



Kuvio 2. Ohjelmistopohjaisen tallennustilainfrastruktuurin osat (Macedo ym. 2020)

2.3 Proxmox Virtual Environment

Proxmox Virtual Environment (usein lyhennettynä **Proxmox VE**) on Proxmox Server Solutions GmbH:n kehittämä Debian GNU/Linux-pohjainen virtualisointialusta-käyttöjärjestelmäjakelu. Sen lähdekoodi on julkaistu GNU Affero General Public License-lisenssillä, ja on siten täysin avointa. Proxmox Server Solutions GmbH myy alustalle kaupallisia tukisopimuksia, mutta kaikki ominaisuudet ovat käytettävissä myös ilmaisversiossa. (*Proxmox VE Administration Guide* 2019.) Alustalla oli vuonna 2020 arviolta yli 350 000 asennusta (*Proxmox 6.2 press release* 2020).

Proxmox-virtuaalipalvelinympäristöllä tarkoitetaan tässä tutkielmassa itse fyysisen palvelimen, Proxmox-käyttöjärjestelmäjakelun ja sen sisäisen konfiguraation muodostamaa kokonaisuutta. Virtuaalipalvelinympäristön toiminnallisuutta laajennetaan tarvittaessa ulkoisilla laitteilla ja ohjelmistoilla, kuten fyysisillä verkkokytkimillä ja työasemilla. Vaihtoehtoisesti on mahdollista yhdistää palvelimet ja työasemat, verkkoinfrastruktuuri ja tallennustila yhdistämisen yhdeksi hyperkonvergoiduksi laitteeksi. Tätä laitetta voidaan käyttää täysin itsenäisenä järjestelmänä, tai se voidaan kytkeä osaksi muuta tietojärjestelmäinfrastruktuuria.

Virtualisointiteknologiat, hallinta ja klusterointi

Proxmox VE tukee KVM-pohjaista virtualisointia ja LXC-konttiteknologiaa, joita voidaan käyttää yhtäaikaisesti. Virtuaalikoneiden hallinta ja Proxmox-palvelimen ylläpito ta-

pahtuu joko selainpohjaisen graafisen käyttöliittymän, komentorivin tai ohjelmointirajapinnan (API) kautta. Virtuaalikoneista voidaan ottaa automaattisia varmuuskopioita tai live-tilannevedoksia (snapshot). Tarvittaessa virtuaalikoneista voidaan luoda mallipohjia (*template*), joista voidaan tarvittaessa kloonata uusia virtuaalikoneita. Kloonit voivat olla täydellisiä (virtuaalikoneen kiintolevytiedosto kopioidaan kokonaisuudessaan) tai linkitettyjä (kloonatun virtuaalikoneen kiintolevytiedostossa säilytetään ainoastaan muutokset alkuperäiseen virtuaalikoneeseen verrattuna). (*Proxmox Virtual Environment 6.2 - Datasheet* 2020 ja *Proxmox VE Administration Guide* 2019.)

Proxmox VE tukee myös usean palvelimen klusterointia, jotka mahdollistaa korkea saatavuus (*high availability, HA*) -toteutukset. Klustereille voidaan luoda jaettu tallennustila verkkotallennustilateknologioiden (esimerkiksi iSCSI tai NFS) avulla tai Ceph-hallintaohjelmistoa käyttäen. Tällaisessa kokoonpanossa käynnissä olevien virtuaalikoneiden siirtäminen klusterin sisällä (*live migration*) on mahdollista. (*Proxmox Virtual Environment 6.2 - Datasheet* 2020.)

Verkko-ominaisuudet

Verkkoliikenteeseen Proxmox VE tarjoaa kaksi vaihtoehtoista toteutusmallia: Linux-sillat tai Open vSwitch -ohjelmistokytkimet (*Proxmox Virtual Environment 6.2 - Datasheet* 2020). Open vSwitch -pohjaista verkkototeutusta on mahdollista laajentaa Proxmox-palvelimen ulkopuolisiin Open vSwitch:iä tukeviin verkkolaitteisiin SDN-verkkototeutusparadigman mukaisesti. Proxmox VE sisältää myös `netfilter`-pohjaisen palomuurin, joka voidaan määritellä klusteri- palvelin- tai virtuaalikonekohtaiseksi tarpeen mukaan (*Proxmox Virtual Environment 6.2 - Datasheet* 2020).

Laitteistovaatimukset

Proxmox Virtual Environment tarvitsee toimiakseen AMD64-käskykantaa tukevan prosessorin, joka tukee Intel VT- tai AMD-V-virtualisointiominaisuuksia. Käyttöjärjestelmä vaatii minimissään 1 gigatavun RAM-muistia, jonka lisäksi on varattava käyttötarkoituksen puolesta riittävästi muistia virtuaalikoneiden tai konttien käyttöön. Yhden palvelinyksikön asen-

nukselle riittää yksi verkkokortti, mutta useamman palvelimen klusterointiin ja/tai tallennustilan jakamiseen on varattava suorituskykysyistä omat verkkokorttinsa. Tallennustilaksi suositellaan nopeita SSD-levyjä laitteistopohjaisessa RAID-konfiguraatiossa. (*Proxmox VE Administration Guide* 2019.)

2.4 Yhteenveto

Tässä luvussa käsiteltiin tutkielman teknologista pohjaa käymällä läpi tutkimuksessa käytettyjä olennaisia käsitteitä ja periaatteita. Virtualisoinnin käsite laajennettiin koskemaan erityisesti alustavirtualisointia. Seuraavaksi käytiin läpi hyperkonvergoidun infrastruktuurin osat. Lopuksi esiteltiin tutkielmassa syntyvän suunnittelutieteellisen artefaktin implementaatioon käytettävä virtualisointialusta Proxmox Virtual Environment.

3 Kyberturvallisuusharjoitus

Tässä luvussa määritellään kyberturvallisuusharjoituksen käsite ja syvennetään sitä skenaarion, teknisen toteutuksen ja sääntöjen ja käytänteiden kautta. Tämän jälkeen esitellään virtualisoituihin kyberturvallisuuden opetusympäristöihin liittyvää aiempaa tutkimusta. Tutkimusten tarkastelu painottuu opetusympäristöissä tehtyihin arkkitehtuurivalintoihin ja opetusympäristöjä varten kehitettyihin suunnittelu- ja hallintatyökaluihin. Tavoitteena on löytää kirjallisuudesta ratkaisuja ja toimintatapoja, joista on hyötyä tässä tutkielmassa kehitettävän artefaktin suunnittelussa, toteutuksessa ja arvioinnissa.

3.1 Kyberturvallisuusharjoituksen määritelmä

Kyberharjoitus on tapahtuma, jossa organisaatio mallintaa ja testaa varautumistaan erilaisiin kyberhäiriöihin (Traficom 2019). Harjoituksessa luodaan kuvitteellinen kriisitilanne, jonka puitteissa on mahdollista havainnoida kyberhäiriön vaikutuksia ja niistä toipumista. Varsinainen harjoituksen toteutustapa riippuu organisaation tarpeista, toiveista ja resursseista. Harjoituksen tavoitteet, mittakaava, osallistujajoukko ja mahdolliset painopistealueet on hyvä määritellä ennen harjoitustyypin valitsemista. (Traficom 2019.)

Kyberharjoituksille ja kyberharjoitusmetodologioille ei ole olemassa yhtä ainoaa, yleisesti hyväksyttyä kategorisointia. Useat harjoitusohjeet pohjautuvat organisaatioturvallisuutta testaavia harjoituksia käsittelevään ISO 22398-standardiin (*Societal security - Guidelines for exercises* 2013). Makrodimitris ja Douligeris (2015) vertasivat kolmea eri harjoitusmetodologiaa ISO 22398-standardissa esitettyihin ohjeisiin, ja havaitsivat, että metodologioissa ehdotetut harjoitustyypit poikkesivat toisistaan. ENISA jakaa kyberharjoitukset keskustelu- ja operaatiopohjaisiin harjoituksiin. Keskustelupohjaisissa harjoituksissa harjoituksen sisältö käydään läpi teoriassa keskustelemalla, kun taas operaatiopohjaisissa harjoituksissa harjoitukseen kuuluvat toimenpiteet toteutetaan käytännössä. (Ouzounis 2009.)

Traficom (2019) jaottelee tarkemmin omiksi harjoitustyypeikseen työpöytäharjoitukset, juurisyyharjoitukset, toiminnalliset harjoitukset, tekniset harjoitukset, *capture the flag* -harjoitukset ja suuret yhteisharjoitukset. **Työpöytäharjoitukset** ovat keskustelullisia harjoi-

tuksia, jotka keskittyvät esitettyjen ongelmien dokumentointiin ja ratkaisuun kirjallisessa muodossa. **Juurisyyharjoitukset** ovat nekin eräänlaisia työpöytäharjoituksia, joissa keskitytään kyberhäiriöiden mahdollisten aiheuttajien ennakointiin ja dokumentointiin. Työpöytäharjoitukset ja juurisyyharjoitukset eivät edellytä teknistä harjoitteluympäristöä, ajastettuja syötteitä tai interaktiivisuutta. **Toiminnalliset harjoitukset** puolestaan ovat kriisitilanteessa toimimista ja viestintää testaavia harjoituksia, joissa osallistujat joutuvat reagoimaan ajallisesti etenevään kerrontaan ja syötteisiin. **Teknisillä harjoituksilla** tarkoitetaan yleisesti kyberharjoituksia, joissa harjoitellaan tietoteknisessä ympäristössä tai sen osassa. **Capture the flag -harjoitukset** ovat pelillisiä harjoituksia, joissa kilpaillaan yksin tai joukkueina. Osallistujat asettuvat kyberhyökkääjän asemaan tunkeutumalla harjoitusjärjestelmään ja keräämällä sieltä "lippuja" eli pisteitä. **Suurilla yhteisharjoituksilla** tarkoitetaan kyberharjoituksia, joissa useat organisaatiot, viranomaiset tai muut yhteistyökumppanit harjoittelevat yhdessä kyberhäiriötilanteissa toimimista. Niihin voidaan yhdistää elementtejä useista eri harjoitustyypeistä. (Traficom 2019.)

3.2 Skenaario

Skenaariolla tarkoitetaan kyberharjoituksista puhuttaessa kuvitteellista kehyskertomusta, jolla selitetään harjoituksen olosuhteita ja tapahtumia. Skenaariossa kuvataan halutunlainen poikkeustilanne, siihen johtaneet ja siitä seuranneet ongelmat (Traficom 2019). Ajallisesti etenevissä harjoituksissa skenaario muuttuu ja mukautuu osallistujien tekemien valintojen mukaan. Skenaario voi harjoitustyypistä riippuen sijoittua suoraan organisaation omaan toimintaympäristöön tai laajempaan fiktiiviseen maailmaan. Fiktiiviseen maailmaan sijoittuvaa skenaariota voidaan tukea erilaisella taustamateriaalilla, kuten kirjallisilla kuvauksilla tai esimerkiksi videoidulla uutislähetyksellä (Traficom 2019).

3.3 Tekninen toteutus

Kyberharjoituksella on harjoitustyypistä riippuen vaihtelevat tekniset toteutustarpeet. Teknisen osion sisältävä kyberharjoitus on mahdollista toteuttaa suoraan organisaation tietoteknisessä ympäristössä tai vaihtoehtoisesti halutulla tasolla aitoa ympäristöä jäljittelevässä,

harjoitusta varten rakennetussa simuloidussa järjestelmässä (Traficom 2019). Molemmilla toteutustavoilla on hyviä ja huonoja puolia. Aidossa järjestelmässä toteutetussa harjoituksessa on mahdollista päästä hyvin lähelle organisaation todellisia toimintamalleja ja prosesseja, ja laite/ohjelmistoympäristö on harjoittelijoille entuudestaan tuttu (Traficom 2019). Toisaalta tällöin harjoitus täytyy suunnitella niin, etteivät harjoituksessa tapahtuvat hyökkäystilannetta jäljittelevät syötteet aiheuta häiriöitä järjestelmän varsinaiseen toimintaan, vaurioita järjestelmän laitteistoa tai aiheuta tietoturvauhkaa organisaation ulkopuolisille järjestelmille. Esimerkiksi tuotantokäytössä olevassa järjestelmässä ei ole suotavaa levittää vakavia haitta- ja kiristysohjelmia harjoituksen yhteydessä. Harjoitusjärjestelmän ulkopuolelle karkaavat porttiskannaukset voivat myös aiheuttaa huolta verkkoja ja palvelimia ylläpitäville tahoille.

Simuloidussa harjoitusjärjestelmässä toteutuksen rajat ovat laajemmat. Harjoitusjärjestelmää voidaan muokata hyvin pitkälti harjoituksen tarpeisiin: järjestelmään voidaan syöttää halutunlaisia haavoittuvia ohjelmistoversioita, takaportteja ja puutteellisia konfiguraatioratkaisuja. Erityisesti pedagogisesti orientoituneissa harjoituksissa näistä voidaan generoida kiinnostavia opetustilanteita. Toisaalta simuloidussa järjestelmässä käytettävät työkalut ja eroavaisuudet tuotantoympäristön kanssa voivat vaikuttaa harjoituskokemukseen (Traficom 2019). Tällaisista eroista voi kuitenkin olla hyötyäkin, jos harjoituksen tavoitteena on harjaannuttaa osallistujia toimivaan poikkeavassa, vieraassa tietoteknisessä ympäristössä.

3.4 Säännöt ja käytänteet

Kyberharjoituksen säännöt ja järjestäytyminen käytännössä riippuvat pitkälti valitusta harjoitustyypistä. Keskustelulliset harjoitusten, kuten työpöytäharjoitusten ja juurisyyharjoitusten tarpeet poikkeavat toiminnallisista harjoitustyypeistä, joissa on mukana kerrontaa, ajastettuja syötteitä ja mahdollinen tietotekninen harjoitteluympäristö. (Traficom 2019.) On olennaista määrittää, toimitaanko harjoituksessa yksin, yhtenä ryhmänä vaiko joukkueina. Joukkueet voivat pyrkiä samaan päämäärään tai vaihtoehtoisesti kilpailla toisiaan tai muuta vastustajajoukkuetta vastaan.

Capture the flag -harjoitusten ja suurten yhteisharjoitusten osallistujajoukkueiden ja järjestävän tahon rooleille on olemassa tiettyjä vakiintuneita yleisnimityksiä, joita kuitenkin saa-

tetaan käyttää ristiin harjoituksesta riippuen, ja nimitysten alle voidaan yhdistellä useita harjoitusrooleja. Seker ja Ozbenli (2018) esittelevät seuraavan luokittelun kyberharjoituksen roolinimistölle.

- **Blue team** eli puolustajajoukkue valvoo ja suojaa harjoitusjärjestelmää hyökkääjiltä harjoituksen aikana. Puolustajajoukkue on vastuussa harjoitusjärjestelmän yksityisyydestä, eheydestä ja käytettävyydestä.
- **Red team** eli hyökkääjäjoukkue puolestaan pyrkii hyökkäämään puolustajajoukkueen suojelemaan harjoitusjärjestelmään, häiritsemään sen toimintaa ja varastamaan sieltä tietoja.
- **Green team** on vastuussa harjoituksen teknisten järjestelmien valmistelusta ja infrastruktuurin ylläpidosta.
- **White team** huolehtii harjoituksen käytännön järjestelyistä, skenaarion luonnista ja etenemisestä ja harjoitukseen kuuluvista ajoitetuista syötteistä harjoituksen aikana.
- **Yellow team** on harjoituksen tiedonvälittäjäjoukkue, joka välittää tietoa joukkueiden välillä esimerkiksi raporttien tai uutisjuttujen avulla. (Seker ja Ozbenli 2018.)

3.5 Virtualisoidut kyberturvallisuuden opetusympäristöt käytännössä

Virtualisoidulla opetusympäristöllä tarkoitetaan tämän tutkielman kontekstissa tiettyyn käyttötarkoitukseen suunniteltua laboratorioympäristöä, joka mahdollistaa opiskeltavan asian harjoittelun käytännössä. Toisin sanoen tutkielman puitteissa ei olla kiinnostuneita organisaation tavanomaisen IT-infrastruktuurin (esimerkiksi työasemapalvelut tai sähköposti- tai tiedostonjakopalvelimet) virtualisoinnista. Seuraavaksi käydään läpi joitakin esimerkkejä toteutetuista kyberturvallisuuden opetusympäristöistä ja niissä käytetyistä työkaluista. Tutkielman tavoitteiden kannalta erityisen kiinnostavia ovat teknisten kyberturvallisuustaitojen harjoitteluun rakennettujen virtualisoitujen ympäristöjen arkkitehtuurivalinnat: näihin perehtymällä voidaan ymmärtää paremmin, miten ITKST55-kurssin tarpeisiin soveltuva virtualisointiympäristö on mahdollista rakentaa. Arkkitehtuurivalintojen lisäksi tarkastellaan joitakin työkaluja ja tekniikoita, joilla teknisen kyberturvallisuusharjoituksen suunnittelua voidaan helpottaa. Tarkasteltavaksi rajataan teknisten IT-taitojen harjaannuttamiseen pyrkivät kyberturvallisuuden opetusympäristöt: näin ollen esimerkiksi yleiset verkko-

oppimisympäristöt, joissa voidaan järjestää ei-teknistä kyberturvallisuusopetusta jäävät tutkielman ulkopuolelle.

3.5.1 Kyberturvallisuuden opetuskäytön arkkitehtuurivalinnat

Virtualisoitu kyberturvallisuuden opetusympäristö ei ajatuksena ole kovinkaan uusi. Ragsdale, Lathrop ja Dodge (2003) esittelivät United States Military Academy:ssa kehitetyn IWAR-laboratorion, jossa VMWare Workstation-ohjelmistolla varustettuihin työasemiin rakennettiin useista virtuaalikoneista ja virtuaalisista verkkolaitteista koostuva harjoitteluverkko. Harjoitteluverkko jaettiin *punaiseen* verkkoon, jossa hyökkäyksiä tekevät virtuaalikoneet sijaitsivat ja *siniseen verkkoon*, jonka koneisiin hyökkäykset kohdistuivat. Järjestelmän suurimpina etuina pidettiin sen joustavuutta ja siirrettävyyttä: jokaista IWAR-työasemaa voitiin joko ajaa erillään tai ulkoisiin verkkoihin kytkettynä, ja se voitiin tarvittaessa asentaa kannettavaan tietokoneeseen, jolloin sitä voitiin liikutella helposti ympäriinsä (nk. *IWAR-in-a-box*). IWAR-laboratorion työasemat olivat kuitenkin toisistaan erillisiä yksiköitä, eli virtualisoitu opetusympäristö oli hajautettu opiskelijoiden työasemille. (Ragsdale, Lathrop ja Dodge 2003.) Nykyään vastaava tilanne voi syntyä esimerkiksi yliopistokursseilla, joilla opiskelijoita edellytetään tekemään kurssitehtäviä henkilökohtaisilla tietokoneilla ajettavilla virtuaalikoneilla.

Hajautetuissa virtualisointiratkaisuissa voi ilmetä useita ongelmia: opiskelijoiden edistymistä tehtävissä on vaikea seurata opettajien toimesta, tehtävien tarkastaminen on hankalaa ja oppilaiden välisen plagioinnin estäminen saattaa olla myös vaikeaa (Thompson ja Irvine 2018). Lisäksi opiskelijoilta ja henkilökunnalta saattaa kulua runsaasti tehtäville varattua aikaa virtualisointiin liittyvien teknisten ongelmien selvittämiseen. Thompson ja Irvine (2018) esittelevät ratkaisuksi näihin tyypillisiin kompastuskiviin Labtainer-rajapinnan, joka on kontti- eli käyttöjärjestelmävirtualisaatioon pohjautuva, teknisten kyberturvallisuustaitojen kuten verkkoliikenne-analyysin harjoitteluun luotu alusta. Labtainer-rajapinnalla luodut harjoitustehtävät paketoidaan Docker-konteiksi, jotka asennetaan opiskelijan työasemalla Python-skriptillä. Opiskelija pääsee tekemään harjoitustehtävän. Tämän jälkeen hän ajaa toisen Python-skriptin, joka kerää tehtäväympäristöstä tiedot tehtävän suoritustavasta ja tuloksista. Opettaja voi tarkastaa tehtävän ja analysoida suoritustapaa rajapinnasta löytyvien

työkalujen avulla. Labtainer-rajapinnalla luodut harjoitustehtävät on mahdollista satunnaistaa opiskelijakohtaisesti ja suoritetut tehtävät vesileimataan yksilöllisesti tarkastusskriptin toimesta. (Thompson ja Irvine 2018.)

Hajautettujen ratkaisujen vastakohtana voidaan pitää keskitettyä palvelinratkaisua, jossa opiskelijat ottavat yhteyttä keskustietokoneeseen tai keskustietokoneklusteriin. Tunc ja Hariri (2015) esittävät tällaiselle ratkaisulle nimeä CLaaS (Cybersecurity Lab as a Service). ClaaS-palvelu tarjoaa verkkoselainpohjaisen käyttöliittymän, jonka kautta opiskelija voi tehdä erilaisia kyberturvallisuusharjoituksia, kuten salasanan murtamista, DDOS-hyökkäyksiä ja puskuriylivuotoja. ClaaS-ympäristö koostuu edustapalvelimesta, joka tarjoaa käyttöliittymän ja virtuaalikone-konsolin sekä taustapalvelinklusterista, joka luo ja hallinnoi harjoitustehtäviin tarvittavia virtuaalikoneita. Opiskelija kirjautuu järjestelmään sisään, jonka jälkeen hän valitsee haluamansa harjoitustehtävän. Taustapalvelin valitsee harjoitustehtävää varten vähiten kuormitetun palvelimen klusterista ja luo sinne tarvittavan virtuaalikoneen verkkokonfiguraatioineen. Luodun virtuaalikoneen tiedot välitetään edustapalvelimelle, joka avaa selainpohjaisen VNC-yhteyden virtuaalikoneelle ja opiskelija pääsee aloittamaan tehtävän. (Tunc ja Hariri 2015.)

Toisena esimerkkinä keskitetystä pilvipalveluympäristöstä Chen ym. (2017) rakensivat kokeellisen verkkoselaimella käytettävän kyberturvallisuuden opetusympäristön Massive Open Online Course (MOOC) -kursseja varten. Opetusympäristö koostui Proxmox VE-palvelinklusterista, Laravel-PHP-rajapinnasta ja noVNC-konsolijärjestelmästä. Järjestelmä havaittiin joustavaksi opiskelijoiden ja opettajien kannalta – opiskelijat pystyvät harjoittelemaan selaimella paikasta ja ajasta riippumatta, ja opettajat voivat rakentaa monipuolisia harjoituksia useilla eri käyttöjärjestelmillä, laitteisto- ja verkkokonfiguraatioilla. Selainpohjaisuus voi kuitenkin aiheuttaa ongelmia vasteajan ja web-palvelimen rinnakkaisten pyyntöjen kestävyyden suhteen. Opetusympäristö salli useiden virtuaalikoneiden verkkokonfiguraatioiden luomisen harjoitustehtäviä varten. (Chen ym. 2017.)

3.5.2 Kyberturvallisuusharjoituksen suunnittelu- ja toteutustyökalujen kehityssuuntia

Tekniset kyberturvallisuusharjoitukset ovat perinteisesti vaativia toteuttaa: harjoituksen rakentaminen vaatii huomattavaa teknistä osaamista ja runsaasti työtunteja. Lisäksi kehitysprosessin lopputuloksena on usein varsin lineaarinen harjoitustehtävä, jonka muokkaaminen uudeksi tehtäväksi vaatii jälleen manuaalista työtä harjoituksen laatijalta. Tämä voi olla ongelmallista erityisesti CTF-tyylisiä kyberturvallisuusharjoituksia järjestettäessä. Schreuders ym. (2017) kehittämä Security Scenario Generator eli SecGen on Ruby-sovellus, joka pyrkii helpottamaan harjoitustehtävien luontia ja erityisesti niiden variointia. Harjoituksen järjestäjät luovat XML-kielisen korkean tason kuvauksen toivotusta harjoitustehtävästä, jonka jälkeen SecGen lukee skenaariomäärittelyn, satunnaistaa sen toteutuslogiikan ja luo skenaariossa tarvittavat virtuaalikoneet. Skenaario voisi olla esimerkiksi seuraavanlainen:

“Luo virtuaalikone, jossa on seuraavat haavoittuvuudet:

- yksi etäkäytettävä haavoittuvuus, jonka hyväksikäyttö aiheuttaa järjestelmän käyttäjätason kaappauksen, ja
- yksi paikallisesti hyväksikäytettävissä oleva haavoittuvuus, joka voi aiheuttaa järjestelmän täydellisen kaappauksen.”(Schreuders ym. 2017)

SecGen valikoi skenaariomäärittelyn 3 perusteella sopivat satunnaiset haavoittuvuudet, ja luo VirtualBox-virtuaalikoneen, joka konfiguroidaan Puppet- ja Vagrant-provisioinnin kautta. Skenaariossa voidaan määritellä tiettyjä palveluita tai protokollia, joita harjoituksessa halutaan käyttää. Sovellukseen voidaan kehittää ja lisätä uusia haavoittuvuusmoduuleja tarpeen mukaan. SecGen:in lähdekoodi on saatavilla GitHubissa, ja sitä kehitetään edelleen. (Schreuders ym. 2017.)

```
<?xml version="1.0"?>
<scenario [snip]>
  <!-- an example remote storage system, with a
    remotely exploitable vulnerability that can then
    be escalated to root -->
  <system>
   <system_name>storage_server</system_name>
    <base platform="linux"/>
    <vulnerability privilege="user_rwx"
      access="remote"/>
    <vulnerability privilege="root_rwx"
      access="local"/>
    <service/>
    <network type="private_network" range="dhcp"/>
  </system>
</scenario>
```

Kuvio 3. SecGen-skenaariomäärittely XML-kielellä (Schreuders ym. 2017)

Toinen esimerkki skenaariopohjaisesta kyberharjoitusten luontityökalusta on Pham ym. (2016) kehittämä CyRIS. CyRIS pyrkii erityisesti tekemään kyberharjoitusten vaatimista verkkotopologioista yksinkertaisempia toteuttaa. Harjoituksen järjestäjä luo YAML-kielisen skenaariomäärittelyn, jonka jälkeen CyRIS luo määritelmän mukaisen virtuaalikoneen tai virtuaalikonekokonaisuuden. CyRIS käyttää KVM-virtualisointia ja CentOS-virtuaalikonepohjia, jotka muokataan skenaarioon sopiviksi Python-skriptillä. Skenaariomäärittelyssä 4 määritellään virtuaalikoneen verkkokonfiguraatio, asennettavat sovellukset ja koneeseen kohdistettavat hyökkäykset ja haittaohjelmat. (Pham ym. 2016.)

```
- host_settings:
  - id: host_1
    mgmt_addr: 172.16.1.2
    account: crond

- guest_settings:
  - id: desktop
    ip_addr: 192.168.122.100
    basevm_name: basevm_desktop
    tasks:
    - install_package:
      - pacakge_manager: yum
        name: wireshark

    - emulate_attack:
      - attack_type: ddos_attack
        target_account: daniel

    - emulate_traffic_capture_file:
      - format: pcap
        file_name: /home/trainee/traffic.pcap
        attack_type: ssh_attack
        attack_source: 2.95.120.235
        noise_level: medium

    - emulate_malware:
      - name: spyeye
        mode: dummy_calculation

- clone_settings:
  - range_id: 123242
    mgmt_network: 10.1.1.1/16
    - guests:
      - guest_id: desktop
        host: host_1
        mgmt_addr_list: 10.1.1.2; 10.1.2.2;
```

Kuvio 4. CyRIS-skenaariomäärittely YAML-syntaksilla (Pham ym. 2016)

CyRIS pystyy emuloimaan verkkopohjaisia hyökkäyksiä pakettikaappausten avulla. Tarvittaessa hyökkäysliikenteen sekaan voidaan upottaa “normaalia” verkkoliikennettä. CyRIS toteuttaa verkkokonfiguraation Linux-siltoja käyttäen: jokaisella harjoitusinstanssilla ja siten harjoittelijalla on oma erillinen siltansa, johon kaikki harjoitusinstanssiin kuuluvat virtuaalikoneet kytketään. Siltojen välistä liikennettä ei siis ole mahdollista toteuttaa CyRIS:illä suoraan. (Pham ym. 2016.)

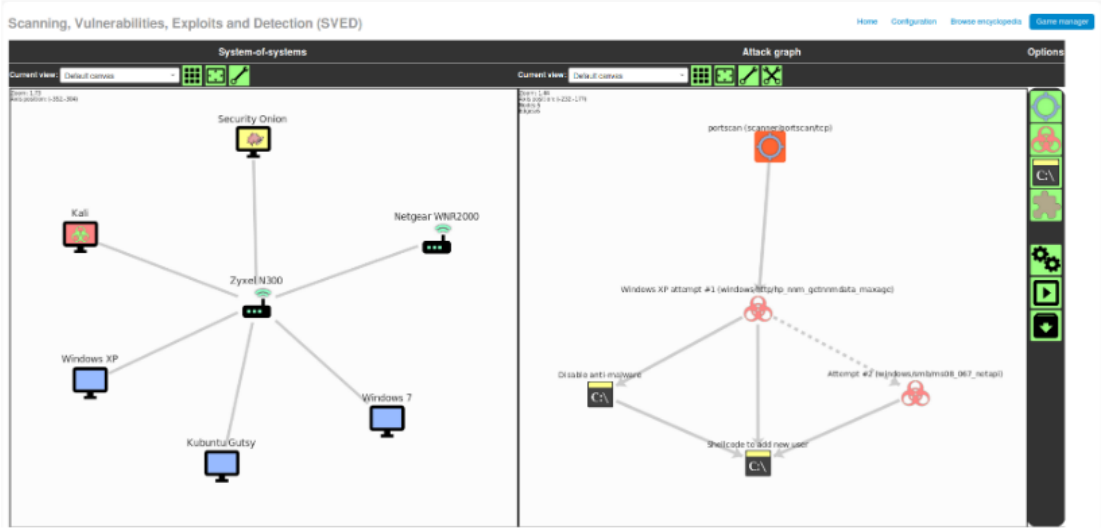
Skenaariosuunnittelua voidaan lähestyä myös formaalin mallintamisen keinoin. Russo, Costa ja Armando (2018) esittelevät Scenario Definition Language (SDL) -laajennoksen TOSCA-määrittelykieleen, joka on IaaS-järjestelmien mallintamiseen yleisesti käytetty määrittelykieli. SCD-laajennos lisää TOSCA-kieleen tietoturvapoikkeamiin liittyviä sääntöjä, reunaehtoja, käyttäytymismalleja ja tavoitteita. Formaalilla mallintamisella pyritään pääsemään eroon kyberharjoituksen rakentamiseen liittyvästä yritysten ja erehdysten prosessista: jos mallin logiikka voidaan vahvistaa pätevästi ennen implementaatiota, on se pätevä myös käytännössä toteutettuna. Tavoitteena on luoda formaali malli kyberharjoitusskenaa-

riosta, joka voidaan antaa IaaS-palveluntarjoajalle toteutettavaksi. SCD-malli ei ota harjoituksen implementaatioyksityiskohtiin paljoakaan kantaa: sen sijaan mallin pohjalta luodaan automaattiset testit, joilla mallin toimivuus voidaan testata käytännössä. (Russo, Costa ja Armando 2018.)

Russo, Costa ja Armando (2020) kehittivät aiemmassa artikkelissaan Russo, Costa ja Armando (2018) esitellystä SDL-laajennoksesta implementaation nimeltä CRACK, jota testattiin käytännössä kokeilemalla sitä realistiseen kyberharjoitusskenaarioon. CRACK:n lähdekoodi on avointa, ja se on saatavilla GitHub:ssa. Implementaatiota vertailtiin muihin kyberlaboratoriokäyttöön kehitettyihin työkaluihin, muun muassa Pham ym. 2016 esittelemään CyRIS:n ja Schreuders ym. 2017 SecGen:n. Russo, Costa ja Armando (2020) havainnoivat, että vertaillut työkalut eivät joko ota kantaa skenaarion testaukseen ja sen toiminnan verifiointiin tai nämä vaiheet on toteutettu puutteellisesti. CRACK mahdollistaa testauksen ja verifioinnin tehokkaan automatisoinnin, ja on siten myös varteenotettava vaihtoehto kyberharjoituksen skenaarion suunnitteluun. (Russo, Costa ja Armando 2020.)

Yamin, Katt ja Gkioulos (2020) mainitsevat autonomiset joukkueet yhtenä kyberlaboratorioiden tutkimuksen kiinnostavista kehityssuunnista. Autonomisilla joukkueilla tarkoitetaan ihmisistä koostuvan kyberharjoitukseen osallistuvan joukkueen korvaamista automattisella työkalulla. Esimerkiksi Schreuders ym. (2017) esittelemä SecGen korvaa kyberharjoituksessa teknisestä infrastruktuurista vastaavan *green team*:n (Seker ja Ozbenli 2018; Schreuders ym. 2017; Yamin, Katt ja Gkioulos 2020).

Holm ja Sommestad (2016) puolestaan pyrkivät automatisoimaan harjoituksessa hyökkäävän *red team*:n toiminnan kehittämällään SVED-työkalukokonaisuudella. SVED koostuu hyökkäysten suunnittelutyökalusta, toteuttajasta ja lokienkerääjästä. Suunnittelutyökalulla luodaan graafi hyökkäyksen kohteista, käytettävistä hyökkäystekniikoista ja verkossa olevista sensoreita, joilla hyökkäyksiä verkossa havainnoidaan. Suunnittelutyökalua (kuvassa 5) voidaan käyttää joko graafisella käyttöliittymällä tai REST-rajapinnan kautta. (Holm ja Sommestad 2016.)



Kuvio 5. SVED-suunnittelutyökalun graafinen käyttöliittymä (Holm ja Sommestad 2016)

Suunnitelma syötetään toteuttajatyökalulle, joka varmistaa hyökkäyskohteiden saavutettavuuden, valmistelee verkossa olevat lokilähteet seuraamaan hyökkäystä ja lopulta suorittaa hyökkäysgraafin askel askeleelta. Lokienkerääjä ottaa vastaan lokeja hyökkäykseen osallistuvilta koneilta ja verkkoon sijoitetuilta sensoreita, jotta avulla hyökkäyksen edistymistä ja onnistumista voidaan tarkkailla. Hyökkäyksen askeleiden toteutukseen SVED tarjoaa moduulipohjaisen järjestelmän, joka tukee useita eri ulkoisia palveluita kuten yhteensopivuuden Metasploit-hyökkäysrajapintaa ja OpenVAS-haavoittuvuusskanneria. SVED mahdollistaa *red team*-toiminnan virtaviivaistamisen ja hyökkäysten toistamisen aina samalla tavalla harjoitusinstanssista riippumatta. (Holm ja Sommestad 2016.)

3.6 Yhteenveto

Tässä luvussa on käsitelty kyberturvallisuusharjoituksen ja skenaarion määritelmää, pohdittu harjoitusten teknistä toteutusta ja esitelty kyberharjoituksissa yleisesti käytettyjä sääntöjä, käytänteitä ja käsitteitä. Seuraavaksi tarkasteltiin virtualisoitujen kyberturvallisuuden opetusjärjestelmien toteutusta erityisesti suhteessa järjestelmissä tehtyihin infrastruktuurivalintoihin. Aiemman tutkimuksen perusteella virtualisointiratkaisut voidaan jakaa karkeasti keskitettyihin ja hajautettuihin ratkaisuihin. Havaittiin, että kyberturvallisuusharjoitusten suunnitteluun ja toteutukseen on kehitetty useita menetelmiä, joiden kehityssuunnista voidaan

eritellä skenaariomäärittelyn perusteella infrastruktuuria konfiguroivat työkalut, skenaarion formaali mallintaminen ja kyberharjoituksen autonomiset joukkueet.

4 Tutkimusmenetelmä

Tutkimus toteutetaan suunnittelutieteellisen tutkimuksen keinoin hyödyntäen joitakin elementtejä konstruktiivisesta tutkimuksesta. Suunnittelutieteellinen tutkimus on tutkimusta, jossa pyritään ratkaisemaan haastavia ongelmia uusilla, innovatiivisilla tavoilla. Innovaatiot on liitettävä niitä ympäröivään liiketoimintaympäristöön ja aiemman tutkimuksen muodostamaan teoriapohjaan. Suunnittelutieteellisessä tutkimuksessa rakennetaan aina artefakti, jolla ongelma pyritään ratkaisemaan. Suunnittelutieteellistä tutkimusta käytetään laajasti erityisesti tietojärjestelmätieteen sovellusaloilla. (March ja Smith 1995; Von Alan ym. 2004; Piirainen ja Gonzalez 2014).

Konstruktiivinen tutkimusote on erityisesti Skandinaviassa suosittu tutkimusmenetelmä, joka on alun perin kehitetty liiketaloustieteen alalle. Sittemmin sitä on sovellettu onnistuneesti muun muassa teollisuuden, logistiikan ja tietojärjestelmätieteen alalla. Konstruktiivinen tutkimus muistuttaa erittäin läheisesti suunnittelutieteellistä tutkimusta, mutta lähestymistavoissa on joitakin eroja esimerkiksi tutkimusprosessin vaiheiden, testauksen toteutuksen ja pohjaolettamuksien suhteen. (Piirainen ja Gonzalez 2014.) Seuraavaksi esitellään konstruktiivisen tutkimuksen ja suunnittelutieteellisen tutkimuksen prosessit pääpiirteittäin ja kerrotaan, miten niitä sovelletaan tässä tutkielmassa.

4.1 Konstruktiivinen ja suunnittelutieteellinen prosessi

Konstruktiivisen tutkimuksen prosessi alkaa ensimmäisessä vaiheessa ratkaistavan ongelman tarkkarajaisesta määrittelystä, sen rajoitusten ja sääntöjen asettamisesta. Tämän jälkeen ongelman ratkaisemiseksi täytyy toisessa vaiheessa organisoida projekti, johon sitoutetaan ongelman ratkaisemisen kannalta olennaiset sidosryhmät. Kolmannessa vaiheessa yhteistyössä sidosryhmien kanssa analysoidaan ongelmaa ja sen teoreettista taustaa kirjallisuuden avulla. Neljännessä vaiheessa suunnitellaan varsinainen konstruktio. Tämä vaihe on luonteeltaan innovatiivinen ja vapaamuotoinen. Käytännössä tutkija ehdottaa ongelmaan ratkaisua prosessin aiempien vaiheita, omaa käytännön kokemustansa ja soveltuvaa teoriaa/kirjallisuutta hyödyntäen. Viidennessä vaiheessa suunniteltu artefakti implementoidaan ja evaluoidaan ho-

listisessa markkinatestissä, joka pyrkii arvioimaan artefaktia kokonaisvaltaisesti. Holistisen markkinatestin on tarkoitus paljastaa, toimiiko artefakti aidossa organisaatiossa vaiko ei. Seuraavaksi siirrytään kuudenteen vaiheeseen eli reflektiovaiheeseen, jossa tulisi pohtia artefaktin sovellettavuutta ja sen mahdollisia rajoituksia. Prosessin seitsemännessä vaiheessa täytyy tehdä ilmi artefaktin teoreettinen kontribuutio. (Piirainen ja Gonzalez 2014.)

Holistinen markkinatesti jaetaan useimmiten eri vaiheisiin. Kasanen, Lukka ja Siitonen (1993) esittävät artefaktin validointiin soveltuvat kolme erilaista testaustyyppiä. Nämä ovat heikko markkinatesti (*weak market test*), keskivahva markkinatesti (*semi-strong market test*) ja vahva markkinatesti (*strong market test*). Heikossa markkinatestissä selvitetään, ovatko liiketoimintayksikön taloudellisista ratkaisuista vastuussa olevat henkilöt kiinnostuneita ottamaan kehitetyn artefaktin osaksi päätöksentekoaan. Keskivahvassa markkinatestissä selvitetään, onko artefakti otettu laajasti käyttöön eri organisaatioissa. Vahvassa markkinatestissä katsotaan, ovatko kehitettyä artefaktia systemaattisesti hyödyntäneet liiketoimintayksiköt tehneet parempaa tulosta kuin ne liiketoimintayksiköt, joissa artefaktia ei ole käytetty. (Kasanen, Lukka ja Siitonen 1993; Piirainen ja Gonzalez 2014.)

Peffers ym. (2007) esittelevät suunnittelutieteelliselle tutkimukselle kuusivaiheisen prosessimallin. Prosessi ei ole välttämättä lineaarinen ja sen vaiheet voivat tapahtua limittäin tai eri järjestyksessä tutkimuksen lähtökohdista ja tavoitteista riippuen. Prosessin ensimmäinen vaihe on ongelman identifioiminen ja motivointi. Tässä vaiheessa tutkimusongelma määritellään tarkkarajaisesti ja sen ratkaiseminen oikeutetaan. Ongelman ratkaiseva artefakti kehitetään vastaamaan ongelmaan. Määrittely auttaa tutkijaa kehittämään ratkaisua ja tuo ilmi yleisölle ratkaisun taustalla olevan ajatuskulun. Toisessa vaiheessa määritellään ratkaisun tavoitteet. Määritellyn ongelman pohjalta selvennetään tavoitteet, jotka kehitettävän artefaktin tulee täyttää. Tavoitteet voivat olla laadullisia tai määrällisiä artefaktin laadusta ja sovellusalueesta riippuen. Kolmannessa vaiheessa artefakti suunnitellaan ja kehitetään. Artefakti voi olla konstruktio, malli, metodi tai instantiaatio, tai kokoelma teknisiä, sosiaalisia ja/tai tiedollisia resursseja. Toisin sanoen suunnittelutieteellinen artefakti voi olla mikä tahansa suunniteltu objekti, jossa on mukana tieteellinen kontribuutio. Artefaktille määritellään aiemmassa vaiheessa tehtyjen tavoitteiden pohjalta haluttu toiminnallisuus ja arkkitehtuuri, ja nämä toteutetaan. Neljännessä vaiheessa artefakti demonstroidaan. Käytännössä tämä tar-

koittaa tilannetta, jossa artefaktilla ratkaistaan yksi tai useampi ongelman aito instanssi. Demonstraatio voidaan tehdä artefaktista riippuen esimerkiksi kokeellisesti, simuloimalla, tapaustutkimuksena, todistamalla tai millä tahansa muulla soveltuvalla menetelmällä. (Peffers ym. 2007.)

Viidennessä vaiheessa artefakti evaluoidaan, eli havainnoidaan ja mitataan, miten tehokkaasti artefakti onnistuu ratkaisemaan ongelman. Ratkaisun tavoitteita vertaillaan demonstraatiovaiheessa havaittuihin tuloksiin soveltuvilla analyysitekniikoilla. Evaluoinnissa voidaan esimerkiksi vertailla artefaktin toteutunutta toiminnallisuutta sille asetettuihin tavoitteisiin, käyttää erilaisia kvantitatiivisia mittareita kuten budjetointidataa, kerätä asiakkailta palautetta tai ajaa simulaatioita. Evaluoinnin päätyttyä tutkijoiden tulee päättää, palataanko takaisin suunnitteluvaiheeseen artefaktin parantamiseksi vai siirrytäänkö seuraavaan vaiheeseen ja jätetään parantaminen tuleville projekteille. Kuudes ja viimeinen vaihe on ratkaisun kommunikointi. Tässä vaiheessa ongelma ja siihen kehitetty ratkaisu esitellään ja oikeutetaan eri soveltajayleisöille. Kommunikaatiossa on syytä painottaa artefaktin hyödyllisyyttä ja uutuusarvoa, sen suunnitteluprosessin kurinalaisuutta ja itse artefaktin toimivuutta. (Peffers ym. 2007.)

Itse tutkielman ja siten tutkimuksen korkean tason tavoitteet ja varsinaiset tutkimuskysymykset esiteltiin luvussa 1.1. Tutkimuksessa halutaan siis selvittää, onko ITKST55-kurssi mahdollista järjestää virtualisoidussa kurssiympäristössä, soveltuuko Proxmox virtualisointialustana tähän ja lisäksi muodostaa käsitys siitä, onko hyperkonvergoidun infrastruktuurin periaatteita mahdollista hyödyntää kyberturvallisuusharjoituksen järjestämisessä. Ymmärtääksemme näiden kysymysten muodostamaa ongelmakokonaisuutta täytyy hahmottaa, millaisia tavoitteita ja vaatimuksia ITKST55-kurssilla on, millaisia ominaisuuksia Proxmox VE-virtualisointijärjestelmästä löytyy ja miten nämä tavoitteet ja vaatimukset voisivat kohdata kyberturvallisuusharjoituksen kontekstissa. Tutkimuksen teoreettinen ja käsitteellinen taustoitus toteutetaan luvuissa 2 ja 3. Tämä taustoitus on osa Piirainen ja Gonzalez (2014) esittelemän konstruktiivisen prosessin ensimmäistä, toista ja kolmatta vaihetta. Osana taustoitusta tehtävä teknisten ratkaisujen kartoitus voidaan myös laskea osaksi mallin neljättä vaihetta. Ratkaisuksi suunniteltavassa artefaktissa täytyy ottaa eri sidosryhmien tarpeet artefaktin suhteen. Tämä on huomioitu vaatimusmäärittelyssä jakamalla vaatimukset eri ryhmiä erityi-

sesti koskeviin vaatimuskategorioihin. Nämä vaatimukset esitellään alaluvussa 5.2. Peffers ym. (2007) esittelemästä prosessimallista tavoitteiden ja vaatimusten määrittely ja taustoitus voidaan puolestaan laskea osaksi ensimmäistä ja toista vaihetta.

Tutkielman luvussa 6 esitellään varsinaisen ITKST55-kurssia varten rakennetun virtuaalipalvelinjärjestelmän toteutus teknisellä tasolla. Toisin sanoen tässä luvussa esitellään tutkielmassa kehitetyn artefaktin implementaatio. Luku jatkaa Piirainen ja Gonzalez (2014) esittelemän mallin neljättä vaihetta ja aloittaa viidennen vaiheen eli implementaation ja markkinatestauksen. Peffers ym. (2007) mallissa luku täyttää kokonaisuudessaan neljännen vaiheen määritelmän. Luvussa 7 käydään läpi kurssin aikana tehtyjä havaintoja virtuaalipalvelinjärjestelmän toiminnasta, joiden perusteella muodostetaan lopullinen evaluointi vertaamalla toteumaa luvussa 5.2 määriteltyihin vaatimuksiin. Tässä luvussa siis jatketaan Piirainen ja Gonzalez (2014) mukaista vaihetta viisi ja edetään vaiheisiin kuusi ja seitsemän. Peffers ym. (2007) kannalta luvussa 7 käydään läpi prosessin viides vaihe ja aloitetaan kuudennen vaiheen läpikäyntiä. Tutkielman viimeinen osuus on pohdinta luvussa 8, jossa aiemmissa luvuissa tehdyt havainnot, teoreettinen tausta ja käytännön kokemukset kytketään yhteen. Tässä luvussa viedään siis loppuun Piirainen ja Gonzalez (2014) vaiheet kuusi ja seitsemän ja viimeistellään Peffers ym. (2007) mukainen kuudes vaihe.

4.2 Artefaktin evaluointi

Edellä esitellyissä kahdessa prosessikuvauksessa kummassakin on mukana laajamittainen testaus/evaluointivaihe. Piirainen ja Gonzalez (2014) määrittelevät konstruktiivisen tutkimuksen implementaatiovaiheeseen kuuluvan holistisen markkinatestauksen ja sen osat, joilla artefaktin toteutusta voidaan testata eri käytännön asteilla. Peffers ym. 2007 puolestaan ottavat evaluoinnin omaksi vaiheekseen ja määrittelevät sille useita mahdollisia lähestymistapoja. March ja Smith (1995) määrittelivät arvioinnin tarkoittavan kriteerien määrittelyä ja artefaktin suoritumisen arvioimista suhteessa kriteereihin. Iivari (2007) kuitenkin huomauttaa, että suunnittelutieteellisen tutkimuksen tuotoksena syntyvät artefaktit voivat olla usein varsin löyhästi kytkettyjä teorioihin, joihin niiden väitetään perustuvan. Suunnitteluorientoitunut tutkimus arvioi artefaktin onnistumista artefaktin hyväksynnän perusteella, ja tällöin suunnittelutieteellisen tutkimuksen teoreettinen kontribuutio voi jäädä rajalliseksi (Piirainen

ja Gonzalez 2014). Tästä syystä evaluoinnin kurinalaisuutta on syytä korostaa, mutta toisaalta on tärkeää myös valikoida sellaiset evaluointitekniikat, joilla toteutetun artefaktin ominaisuudet, onnistumiset ja epäonnistumiset voidaan tuoda mahdollisimman hyvin esiin.

Piirainen ja Gonzalez (2014) esittelemää kolmiportaista holistista markkinatestausta ei sovelleta sellaisenaan tämän tutkielman laajuudessa. Tutkimuksen aikana kehitettiin prototyyppi virtuaalipalvelinjärjestelmästä, jonka toimintaa testattiin käytännössä yhden kurssiinstanssin puitteissa. Suoritettu testaus voitaisiin mieltää keskivahvaksi markkinatestaukseksi: testattiin, että artefakti on otettu reaalimaailman käyttöön. Kolmivaiheista markkinatestausmallia voitaisiin soveltaa jatkokehitystä pohdittaessa: halutaanko prototyypin kaltaista virtuaalipalvelinjärjestelmää mahdollisesti hyödyntää muilla kursseilla, ja miten sen suorituskykyä voitaisiin systemaattisesti verrata muilla teknisillä alustoilla toteutettuihin kursseihin.

Peffers ym. (2007) esittivät artefaktin evaluoinnille useita laadullisia ja määrällisiä toteutustapoja, ja korostivat soveltuvien tapojen huolellista pohtimista. Prat, Comyn-Wattiau ja Akoka (2014) määrittelevät suunnittelutieteellisen tutkimuksen kirjallisuudesta löytämänsä kuusi "geneeristä" evaluointitekniikkaa, joita soveltamalla evaluointi voidaan esimerkiksi toteuttaa:

1. Artefaktin käytön esittely yhden tai useamman esimerkin kautta.
2. Artefaktin käytön esittely yhdessä tai useammassa reaalimaailman tilanteessa.
3. Artefaktin arviointi sitä käyttävien oppilaiden suoritusten arvioinnin kautta.
4. Artefaktin arviointi sitä käyttäneiden oppilaiden hyödyllisyysnäkemysten kautta.
5. Artefaktin käytöstä kerätty laadullinen palaute sen käyttäjiltä.
6. Jos kehitetty artefakti on algoritmi, sen suorituskyvyn vertailu muihin vastaaviin algoritmeihin. (Prat, Comyn-Wattiau ja Akoka 2014.)

Tämä tutkielma painottuu erityisesti artefaktin teknisen toteutuksen esittelyyn ja sen arviointiin. Evaluointi voidaan siis toteuttaa esimerkiksi edellä esitetyllä tavalla numero kaksi. Luvussa 7 esitellään seikkaperäisesti ITKST55-kurssin vuoden 2019 etenemisen aikana tehtyjä havaintoja rakennetun virtuaalipalvelinjärjestelmän toiminnasta ja siinä eri tahojen havaitsemista mahdollisuuksista ja puutteista. Kuvauksessa on tapaustutkimuksellisia piirteitä.

Tämän läpikäynnin perusteella kuvaus kytketään alaluvussa 5.2 esiteltyihin vaatimuksiin, jotta voidaan päätellä, täyttikö virtuaalipalvelinjärjestelmä sille asetetut tavoitteet. Piirainen ja Gonzalez (2014) huomauttavat, että evaluointiprosessi ei aina ole lineaarinen; prosessin aikana voi syntyä uusia evaluointitarpeita ja toisaalta sen aikana voidaan myös havaita kokonaan uusia ongelmia. Tällaisia havaittuja ongelmia ja niiden mahdollisia ratkaisuja ja/tai jatkokehitysehdotuksia käsitellään luvussa 8.

5 ITKST55-kurssi

Tässä luvussa kuvataan ITKST55-kurssin perusajatus sekä opintojakson sisältö ja toteutustavat. Sen jälkeen määritellään tämän kuvauksen perusteella vaatimukset kurssiympäristön varsinaiselle toteutukselle, jotta tutkielmassa tavoitellun kurssilla käytettävän virtuaalipalvelinympäristön toteuttaminen on mahdollista. Luku vastaa Peffers ym. (2007) esittämän suunnittelutieteellisen prosessimallin vaiheita kolme ja neljä eli suunnittelu- ja demonstraatiovaiheita.

5.1 Yleiskuvaus kurssista

ITKST55 - Kyberhyökkäys ja sen torjunta on Jyväskylän yliopistossa vuodesta 2018 järjestetty informaatioteknologian tiedekunnan organisoima syventävän opintojakso. Kurssi on suunnattu erityisesti kyberturvallisuuden maisteriohjelman opiskelijoille. Opintojakson suoritustavoitteiden mukaisesti opiskelija kurssin suoritettuaan *"tuntee keskeisimmät tietoverkkoympäristön analysoinnin ja turvaamisen työkalut ja menetelmät, tunnistaa tietoverkko- ja informaatio-operaatioita, osaa analysoida niitä sekä valita niihin reagointiin sopivimmat tekniset ja muut toimenpiteet"* (*ITKST55 - Kyberhyökkäys ja sen torjunta. Opintoesite.* 2020) Opintojakson sisältö koostuu Jyväskylän yliopistolla järjestettävistä luennoista ja harjoituksista ja erillisestä pelivaiheesta. Varsinainen pelivaihe oli syksyllä 2019 MPK:n Jyväskylän varuskunnassa Tikkakoskella järjestetty Kyberturvallisuuden erikoiskurssi. Kurssisuorituksen saadakseen opiskelijoiden tuli osallistua sekä Jyväskylän yliopistolla järjestettyyn kontaktiopetukseen että Kyberturvallisen erikoiskurssin intensiiviseen pelivaiheeseen, ja lisäksi koota näiden aikana opintopäiväkirja. Tässä tutkielmassa käsitellään tarkemmin ainoastaan Jyväskylän yliopistolla järjestettyä kontaktiopetusvaihetta, jossa tutkielman aikana kehitetty virtuaalipalvelinjärjestelmä oli käytössä.

Kontaktiopetusvaihe järjestettiin viikon mittaisena intensiivijaksona Jyväskylän yliopiston Agora-rakennuksen oppimistila Montussa 9.9.2019 - 13.9.2019. Intensiivijakson alussa maanantaina opiskelijat jaetaan kolmeen puolustajajoukkueeseen (harjoituksen *blue team* -joukkueet), jotka esittävät skenaariossa määriteltyjen kolmen yrityksen palkkaamia tieto-

turvakonsultteja. Tietoturvakonsultit saapuvat järjestelmiin täysin ummikkoina; heidän tehtävänään on selvittää, millainen "yritysten" tietojärjestelmäinfrastruktuuri on, millaisia digitaalisia palveluita ne tarjoavat asiakkailleen ja henkilökunnalleen sekä millaisilla teknisillä ratkaisuilla nämä palvelut on toteutettu. Sen jälkeen tietoturvakonsultit pyrkivät parhaansa mukaan suojaamaan edustamiensa yritysten tietojärjestelmät havaitsemalla ja paikkaamalla löytyneitä haavoittuvuuksia, kuten päivittämällä haavoittuvia sovelluksia ja määrittelemällä palomuurisääntöjä suojaamaan järjestelmää.

Kun opiskelijat ovat ryhmäytyneet ja tutustuneet alustavasti edustamiinsa yrityksiin ja niiden tekniseen infrastruktuuriin, alkaa harjoituksen hyökkääjäjoukkue (*red team*) tehdä erilaisia hyökkäyksiä ja häiriöitä puolustajajoukkueiden suojaamiin tietojärjestelmiin. Puolustajajoukkueiden on havaittava hyökkäys ja pyrittävä torjumaan se parhaansa mukaan, minimoiden sen vaikutukset järjestelmän normaaliin toimintaan. Tämän "pelin" lomassa järjestetään yhteisiä luentosessioita, joilla käsitellään esimerkiksi käytettäviä työkaluja ja termistöä, kyberharjoituksessa toimimista ja tilannekeskustoimintaa. Jokainen kurssipäivä päättyy *hotwash* -sessioon, jossa käsitellään yhteisesti päivän aikana suoritetut toimenpiteet, onnistumiset ja ongelmat.

ITKST55 ei vuoden 2019 toteutuksessa sisällä pisteytystä tai kilpailullisia elementtejä. *Blue team*:it ovat harjoituksessa samassa kaupungissa toimivien yksityisten yritysten palkkaamia kyberturvallisuuskonsultteja. Itse yritykset eivät ole velvoitettuja toimimaan yhteistyössä kyberturvallisuusasioissa, ja harjoituksen skenaarion puitteissa ne saattavat toimia toisiaan vastaan, mutta *blue team*:eilla on yhteinen päämäärä - kyberuhalta suojautuminen - ja he saavat tässä tehtävässä avustaa toisiaan.

5.2 Kurssin asettamat vaatimukset kurssiympäristölle

Seuraavaksi kerrotaan, millainen kurssilla toteutettavan virtuaalipalvelinympäristön tulee olla, jotta se täyttää kurssin vaatimukset. Toisin sanoen tässä kappaleessa esitellään konstruktiivisen tutkimuksen tuloksena syntyvän artefaktin evaluointikriteerit. Vaatimuksella tarkoitetaan tässä yhteydessä seikkaa tai ehtoa, joka rakennettavan kurssiympäristön tulee voida toteuttaa. Vaatimukset pohjautuvat löyhästi Topham ym. (2016) esittämiin kyberturvallisuus-

laboratorion vaatimuksiin, jotka ovat seuraavat:

1. Kyberturvallisuuslaboratorion täytyy olla joustava ja ominaisuuksiltaan monipuolinen, jotta opetushenkilökunta voi sen puitteissa toteuttaa realistisia harjoitusskenaarioita.
2. Kyberturvallisuuslaboratoriossa tulee voida luoda ja toteuttaa harjoitustehtäviä useille opiskelijoille samankaltaisesti.
3. Laboratorioverkossa olevat tietokoneet ja ohjelmistot täytyy eristää ulkoisista verkoista.
4. Laboratorio-olosuhteissa tulee voida antaa opiskelijoille pääkäyttäjä-oikeuksia laboratoriokoneille harjoituksen niin vaatiessa.
5. Laboratorion tulee tarjota mahdollisuus työn tallentamiseen ja työn varmuuskopiointi- ja palautusmahdollisuudet opiskelijoille. (Topham ym. 2016.)

ITKST55-kurssin asettamat vaatimukset voidaan jakaa tutkielmassa toteutettavan evaluoinnin helpottamiseksi kolmeen kategoriaan: kurssin opetussisällön muodostamiin, opetusteknisiin ja teknisiin vaatimuksiin. Kurssin opetussisällön muodostamat vaatimukset ovat asioita, joita kurssiympäristössä tulee voida suorittaa, jotta kurssin opetustavoitteet täyttyvät. Opetustekniset vaatimukset puolestaan ovat asioita, joiden tulee olla mahdollisia kurssin organisaation ja käytännön opettamisen onnistumiseksi. Tekniset vaatimukset taas ovat ominaisuuksia, jotka kurssijärjestelmässä käytettävästä ohjelmisto/laitteistoalustasta tulee löytyä, jotta kurssi on mahdollista järjestää. Ensimmäiset ovat kiinnostavia erityisesti opiskelijoiden, toiset opettajien ja kolmannet esimerkiksi *red team*in, järjestelmän rakentajien ja ylläpitäjien näkökulmasta. On huomioitava, että kaikki kolme vaatimuskategoriaa käsittelevät sellaisia kurssiin liittyviä asioita, joiden toteuttamiseen virtuaalipalvelinympäristö vaikuttaa tavalla tai toisella. Kurssiin liittyvät vaatimukset, jotka eivät vaadi virtuaalipalvelinympäristön toteutumista täyttyäkseen jäävät tämän tutkielman ulkopuolelle.

Seuraavaksi käydään läpi edellä luetellut kolme vaatimuskategoriaa. Jokainen kategoria alkaa vapaamuotoisella vaatimuskuvauksella, jonka pohjalta muodostetaan varsinaiset vaatimukset. Näitä vaatimuksia verrataan myöhemmin tutkielman luvussa 7.3 läpikäytävään vuoden 2019 ITKST55-kurssitoteutukseen kehitetyn prototyyppijärjestelmän evaluoimiseksi.

5.3 Opetussisällölliset vaatimukset

Kurssin kulku sisältää luentoja sekä ryhmätyöskentelyä ja itsenäistä työskentelyä peliympäristössä. Luokkatila on suunniteltava niin, että nämä kaikki on mahdollista toteuttaa. Opiskelijoiden tulee saada käyttöönsä työasemat, joilta löytyvät intensiivijakson kyberharjoituksessa tarvittavat työkalut ja joilta voi käyttää harjoitukseen (“peliin”) kuuluvia järjestelmiä. Kurssin tekniseen järjestäytymiseen ja järjestelmän haltuunottoon täytyy varata tarpeeksi aikaa, sillä virtuaalipalvelinjärjestelmä voi olla vaikeampi hahmottaa kuin perinteinen fyysisiin tietokoneisiin ja verkkolaitteisiin nojaava järjestelmä.

Virtuaalipalvelinjärjestelmän täytyy olla käytettävyydeltään mahdollisimman paljon aitoja palvelinympäristöjä vastaava. Tämä tarkoittaa esimerkiksi sitä, että virtuaalipalvelinten etäkäytön tulee onnistua tavanomaisilla työkaluilla, kuten verkkoselaimilla, pääte-emulaattoreilla ja etätyöpöytäsovelluksilla. Virtualisointiympäristön käytön opiskelu ei kuulu kurssin opetussisältöön, ts. sen täytyy olla kurssilaisille mahdollisimman läpinäkyvä: kurssilaisten ei täydy välittää siitä, millaisilla teknisillä ratkaisuilla kurssin harjoitteluympäristö on toteutettu. Tämä voidaan sitoa Topham ym. (2016) esittämään vaatimukseen 1 kyberturvallisuuslaboratorion joustavuudesta: järjestelmän täytyy olla tarpeeksi realistinen, jotta skenaariokerronta sen puitteissa onnistuu uskottavasti.

Vaatimuskuvauksen perusteella määriteltiin seuraavat opetussisällölliset vaatimukset:

- **OS1**. Luokkatilan tulee sallia peliympäristöön pääsy kaikille sitä tarvitseville osapuolille.
- **OS2**. Opiskelijoiden tulee saada käyttöönsä fyysiset työasemat, joilta löytyvät tarvittavat työkalut ja joilla voidaan käyttää harjoitukseen liittyviä peliverkon järjestelmiä.
- **OS3**. Virtuaalipalvelinten etäkäytön tulee onnistua tavanomaisilla työkaluilla, kuten verkkoselaimilla, pääte-emulaattoreilla ja etätyöpöytäsovelluksilla.
- **OS4**. Virtuaalipalvelinympäristön tulee olla kurssilaisille läpinäkyvä.

5.4 Opetustekniset vaatimukset

ITKST55 on vuonna 2019 täysin lähiopetuksena järjestettävä kurssi. Kaikki osallistuvat tahot työskentelevät paikan päällä samassa yliopiston tilassa, joten etäyhteyksiä ei tarvitse toteuttaa. Opetustila tulee jakaa niin, että joukkueilla ja opettajilla on tilaa työskennellä. *Red team* on erotettava puolustajajoukkueista niin, että se pystyy kommunikoimaan keskenään rauhassa. Tästä "erotetusta tilasta" on oltava tietoliikenneyhteydet peliverkkoon.

Kurssin aikana peliverkossa tapahtuvia ilmiöitä täytyy voida havainnollistaa koko ryhmän kesken yhteisesti. Tätä varten täytyy rakentaa niin sanottu tilannekuva-työasema, joka näkee kaiken peliverkossa tapahtuvan verkkoliikenteen. Tilannekuva-työaseman on käytettävä Security Onion -käyttöjärjestelmäjakelulla toteutettua valvontapalvelinta, kuten kurssilaisten käytössä olevat valvontatyöasemienkin. Työasema on kytkettävä luokkatilassa sijaitsevaan televisioon tai projektoriin, jonka avulla verkkoliikennettä ja verkossa havaittuja tapahtumia voidaan visualisoida Security Onion -jakelusta löytyviä työkaluja käyttäen. Tilannekuvaa voidaan käyttää osana skenaariokerrontaa, joten se voidaan kytkeä Topham ym. (2016) vaatimukseen numero 1.

Vaatimuskuvauksen perusteella määriteltiin seuraavat opetustekniset vaatimukset:

- **OT1**. Harjoituksen *red team* on erotettava puolustajajoukkueista niin, että se pystyy kommunikoimaan rauhassa.
- **OT2**. *Red team*:in tilasta on oltava tietoliikenneyhteydet peliverkkoon.
- **OT3**. Peliverkon tapahtumia täytyy voida havainnollistaa tilannekuvatyöasemalla, joka on kytkettävä yleiseen valvontavirtuaalikoneeseen.
- **OT4**. Kurssilaisten käytössä olevilta valvontatyöasemilta täytyy voida käyttää tiimikohtaisia Security Onion -valvontavirtuaalikoneita.

5.5 Tekniset vaatimukset

ITKST55:n teknisiä työkaluja sisältävä osuus toteutetaan ns. peliverkossa. Peliverkossa on julkisia, kaikille joukkueille ja henkilökunnalle saatavilla olevia palveluita (yleinen infrastruktuuri, "Common Infrastructure"). Näitä palveluita voivat olla esimerkiksi yhteinen wiki

tiedonjakoa varten ja jokin sosiaalisen median palvelu, jota käytetään osana skenaariokerrontaa. Lisäksi jokaisella *blue team*:illa on vastuullaan yhden yrityksen sisäverkko palvelimineen ja työasemineen ("BT-verkot"). Jokaisella *blue team*:illa tulee olla toiminnallisesti identtinen BT-verkko, sisältäen samoilla ohjelmistoversioilla toteutetut työasemat ja palvelimet. Tämä vastaa Topham ym. (2016) vaatimusta numero 2. BT-verkossa ajettavien sovellusten, kuten web-palvelinten näyttämä sisältö voi vaihdella joukkuekohtaisesti skenaariosyistä, kunhan itse palvelimet on konfiguroitu samalla tavalla ja ohjelmistoversiot ovat samat.

Harjoituksen *red team*:in käyttämien työasemien täytyy voida kytkeytyä harjoituksen BT-verkkoihin peliverkon Internetiä vastaavalta alueelta eli "Common Infrastructure"-verkosta. *Blue team*:ien fyysiset työasemat puolestaan kytkeytyvät verkkoon edustamansa yrityksen "intranet"-alueelta. Tällä pyritään jäljittelemään tilannetta, jossa hyökkääjät hyökkäävät yrityksen järjestelmiin internetin yli ja puolustajat pyrkivät torjumaan hyökkäyksen yrityksen tiloista käsin.

Opiskelijoilla tulee olla intensiivijakson aikana käytettävissään Internet-yhteys, jonka avulla hakea ratkaisuja teknisiin ongelmiin ja asentaa päivityksiä virtuaalikoneisiin. Internet-yhteyttä tulee voida käyttää suoraan BT-työasemalta. Harjoituksessa käytetään kurssisisällön mukaisia, etukäteen päätettyjä ja esiasennettuja työkaluja, mutta mahdollisuus lisätyökalujen asentamiseen kurssin aikana on oltava. Internet-yhteys on toteutettava niin, etteivät harjoituksessa käytetyt syötteet ja hyökkäykset pääse vahingossa tai tarkoituksella peliverkon ulkopuolelle julkiseen Internetiin. Tämä vastaa Topham ym. (2016) esittämää vaatimusta numero 3.

Virtuaaliverkkoinfrastruktuurissa on oltava erilliset valvontavirtuaalikoneet ja näiden käyttöön tarkoitetut valvontatyöasemat, joilla kurssin osallistujat ja henkilökunta voivat tarkkailla ja visualisoida verkkoliikennettä. Opetustekniset vaatimukset-alaluvussa kuvatun tilannekuva-työaseman käyttöön tuleva Security Onion -valvontavirtuaalikone on asennettava niin, että sinne peilataan kaikki peliverkossa käytävä verkkoliikenne. BT-valvontavirtuaalikoneille on peilattava blue teamien intranet- ja DMZ-verkoissa tapahtuva verkkoliikenne.

Jokainen BT-joukkue saa täydelliset pääkäyttäjäoikeudet joukkuekohtaisen BT-verkkonsa

virtuaalikoneisiin, jotta järjestelmän tarkastelu ja suojaaminen on mahdollista. Lisäksi joukkueille voidaan tarvittaessa antaa erilliskäyttöoikeuksia käytettävään virtualisointijärjestelmään, jos se on skenaarion mukaisesti tarpeen. Nämä vaatimukset vastaavat Topham ym. (2016) vaatimusta numero 4. Lisäksi Topham ym. (2016) vaatimuksen numero 5 mukaisesti harjoituksessa käytettäville virtuaalikoneille tulee luoda säännöllisesti palautuspisteitä, joihin virtuaalikone voidaan palauttaa ylitsepääsemättömissä teknisissä ongelmatapauksissa.

Vaatimuskuvauksen perusteella määriteltiin seuraavat opetustekniset vaatimukset:

- **T1**. Peliverkossa on oltava oma verkkoalue yleistä infrastruktuuria varten.
- **T2**. Jokaista blue team:ia kohden on oltava toiminnallisesti identtinen BT-verkko.
- **T3**. *Red team*:in fyysisten työasemien tulee kytkeytyä BT-verkkoihin yleisen infrastruktuurin verkkoalueelta.
- **T4**. *Blue team*:ien fyysisten työasemien tulee kytkeytyä BT-verkkoihin intranet-alueelta.
- **T5**. Opiskelijoilla tulee olla kurssin aikana fyysisiltä BT-työasemilta käytettävä Internet-yhteys.
- **T6**. Peliverkosta ulospäin menevät Internet-yhteydet tulee toteuttaa niin, etteivät harjoituksessa käytettävät syötteet pääse peliverkon ulkopuolelle.
- **T7**. Peliverkossa on oltava yleinen valvontavirtuaalikone, joka havainnoi koko peliverkon liikennettä.
- **T8**. Jokaisella *blue team*:illa on oltava käytössään oma valvontavirtuaalikone, johon peilataan BT-verkon DMZ- ja Intranet -liikenne.
- **T9**. Jokaisella *blue team*:illa on oltava täydelliset pääkäyttäjäoikeudet BT-verkkonsa virtuaalikoneisiin.
- **T10**. Joukkueille tulee voida tarvittaessa sallia pääsy itse virtualisointialustaan.
- **T11**. Kaikille harjoitukseen kuuluville virtuaalikoneille on luotava säännöllisesti palautuspisteitä.

6 Kurssilla käytettävän virtuaalipalvelinympäristön toteutus

Tässä luvussa kuvataan, miten edellisessä luvussa määritellyt vaatimukset täyttävä virtuaalipalvelinympäristö, kutsumanimeltään **Proomu** rakennettiin käytännössä. Ensin esitellään käytössä ollut palvelinlaitteisto ja Proxmox-alusta ohjelmineen. Sen jälkeen käsitellään Proxmox-alustan, virtuaalikoneiden ja fyysisten verkkolaitteiden yhteistä verkkokonfiguraatiota ja kerrotaan tarkemmin virtuaalisten reitittimien toteutuksesta pfSense- ja Ubuntu -virtuaalikoneita käyttäen. Lopuksi käsitellään verkkoliikenteen peilauksen toteutus ja kerrotaan lyhyesti kyberharjoitukseen kuuluvien palvelinten ja työasemien kokoonpanoista ja niiden tarjoamista palveluista.

6.1 Käytetty laitteisto ja ohjelmistoversiot

Virtuaalipalvelinympäristö koostuu itse fyysisestä palvelimesta, siihen kytketyistä fyysisistä verkkolaitteista ja fyysisistä työasemista. Laitteiston ominaisuudet olivat seuraavat:

Dell R740-palvelin

- 2x Intel Xeon Platinum 8160-prosessori, 24c/12t @ 2.10GHz
- 756GiB DDR4 RAM
- 2x 256GiB M.2 SATA SSD, RAID1
- 8x 1Tb SSD 3,5Tb RAID60, PERC H740P-RAID-ohjain
- Intel I350-verkkokortti, 4 porttia (1GbE)
- Intel X710-verkkokortti, 4 porttia (10GbE, SFP+)

Fyysiset verkkolaitteet

- Dell EMC S3124P-kytkin, 24GbE-porttia + 2 SFP+-porttia
- Linksys WLAN-reititin
- 3x Zyxel-kytkin, 8 porttia

Fyysiset työasemat

- 12x Dell-kannettava tietokone (BT-työasemat)
- 3x BT-Security Onion-käyttöön tarkoitetut työasemat
- 3x valvontatyöasemat
- 1x Dell-kannettava, yleinen valvontatyöasema runkoverkossa

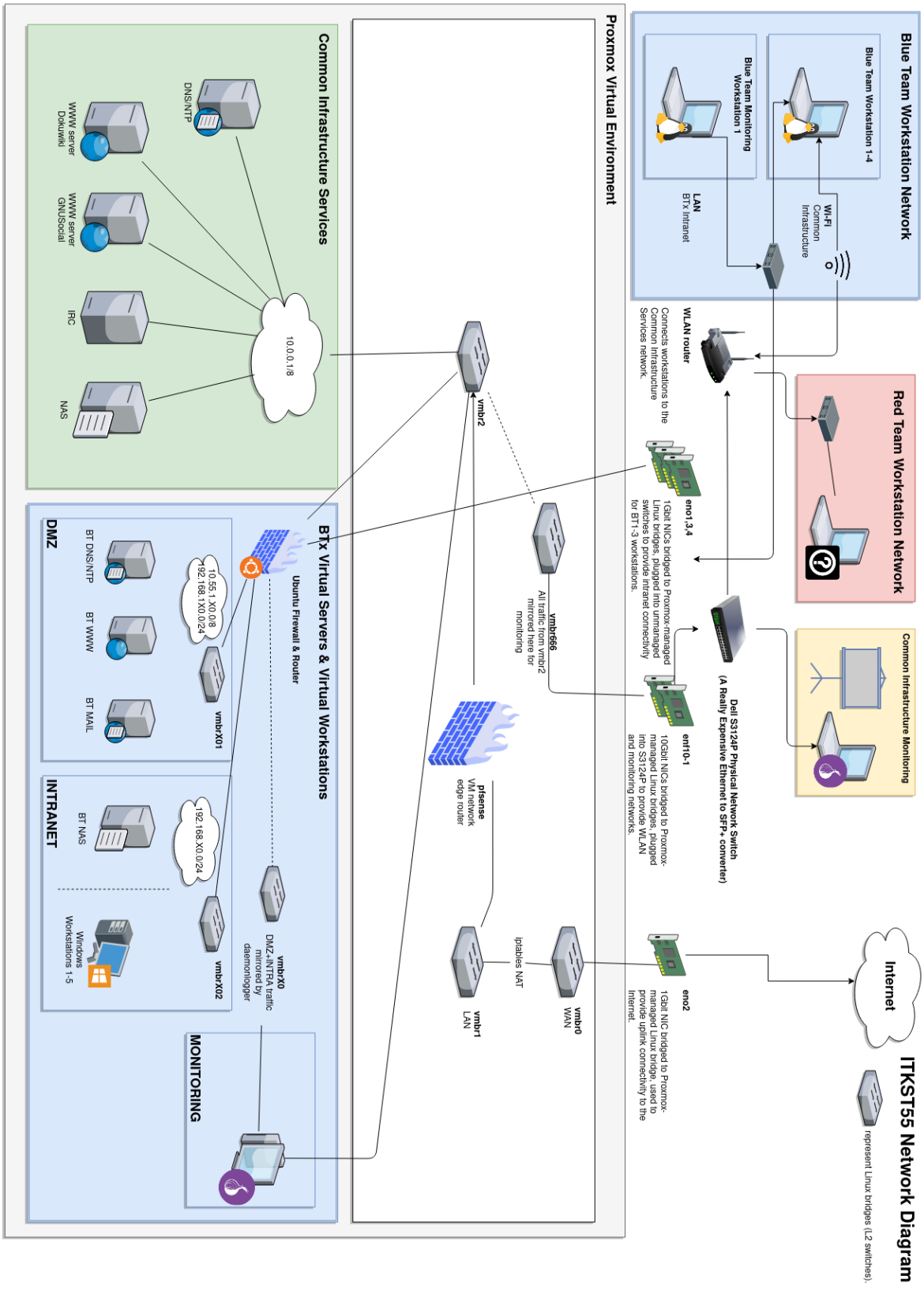
Kurssitoteutus järjestettiin Proxmox Virtual Environment 6.0-versiolla. Harjoituksessa käytössä oleva Linux-kerneliversio oli pve-kernel-4.15.18-18. Verkkoliikenteen tarkkailuun asennettiin erikseen paketit `htop`, `nload` ja `net-tools`. Lisäksi asennettiin verkkoliikenteen peilausta varten `daemonlogger`. Kaikki sovellukset asennettiin suoraan Debian-projektin Debian 10 (`buster`)-tietovarastosta. Proxmox-päivitykset asennettiin Proxmox Server Solutions GmbH:n ylläpitämästä ilmaista yhteisöversiota tarjoavasta `no-subscription`-tietovarastosta.

6.2 Verkon toteutus

Proomu-virtuaalipalvelinjärjestelmän verkkokonfiguraatio voidaan jakaa loogisesti kahteen osaan: Proxmox VE:n hallinnoimaan virtuaaliverkkoon ja fyysisiin verkkolaitteisiin itse palvelimessa ja sen ulkopuolella. Palvelimen ulkopuolisia fyysisiä verkkolaitteita ovat WLAN-reititin, Dell S3124P-kytkin ja hallinnoimattomat verkkokytkimet. Alaluvussa 5.5 käsiteltyjen kurssivaatimusten mukaisesti blue team-verkkoja on kolme rinnakkaista ja toiminnallisesti identtistä kappaletta. Jokaisella blue teamilla on käytössään 4 työasemaa, joita voidaan käyttää kahdella eri tavalla: *WLAN-yhteys* kaikille kurssilaisille yhteisiin Common Services-palveluihin ja *Ethernet-yhteys* BTx-sisäverkkoihin. Langatonta yhteyttä käytettäessä työasemilta pääsee Internetiin. RT hyökkää BT-verkkoihin Common Infrastructure-verkkoalueelta (“peli-internetistä”) käsin, myöskin virtuaaliverkkoon sillattuja työasemia käyttäen. Jokaisella BT:llä on myös käytössään BT-työasema, joka on tarkoitettu BT-kohtaisen Security Onion-valvontapalvelimen käyttöön. Lisäksi kaikki peliverkossa tapahtuva liikenne peilataan yleiselle Security Onion -valvontapalvelimelle.

Kuvassa 6 esitellään Proomu-virtuaalipalvelinjärjestelmä kokonaisuudessaan. Vaaleanharmaan “Proxmox Virtual Environment”-laatikon sisälle jäävät verkkolaitteet ovat virtuaalisia Linux-siltoja/kytkimiä *(Linux bridge)*. Loput verkkolaitteista ovat fyysisiä. Tässä luvussa

käsitellään verkkokuvan komponentit: virtuaaliset kytkimet, kaksi eri virtuaalireititintyyppiä (pfSense ja Ubuntu), verkkoliikenteen peilaus monitorointia varten, fyysiset verkkolaitteet ja lopuksi verkkokuvan virtuaaliset ja fyysiset palvelimet ja työasemat yleisellä tasolla.



Kuvio 6. Kuva verkon toteutuksesta yleisellä tasolla

6.2.1 Virtuaaliset kytkimet

vmbr0 on virtuaaliverkon hallintakytkin *(management interface)* ja samalla reitti Internetiin. vmbr0:lle on omistettu fyysinen verkkoliitäntä **eno2**. vmbr0:n kautta Proomua on mahdollista etähallita Internetin välityksellä, eikä Proomun pääsy internetiin katkea muissa virtuaaliverkkokytkimissä tehtyjen muutosten myötä.

```
iface vmbr0 inet static
address  172.2x.xx.xx
netmask  255.255.0.0
gateway  172.xx.0.1
bridge-ports eno2
bridge-stp off
bridge-fd 0
```

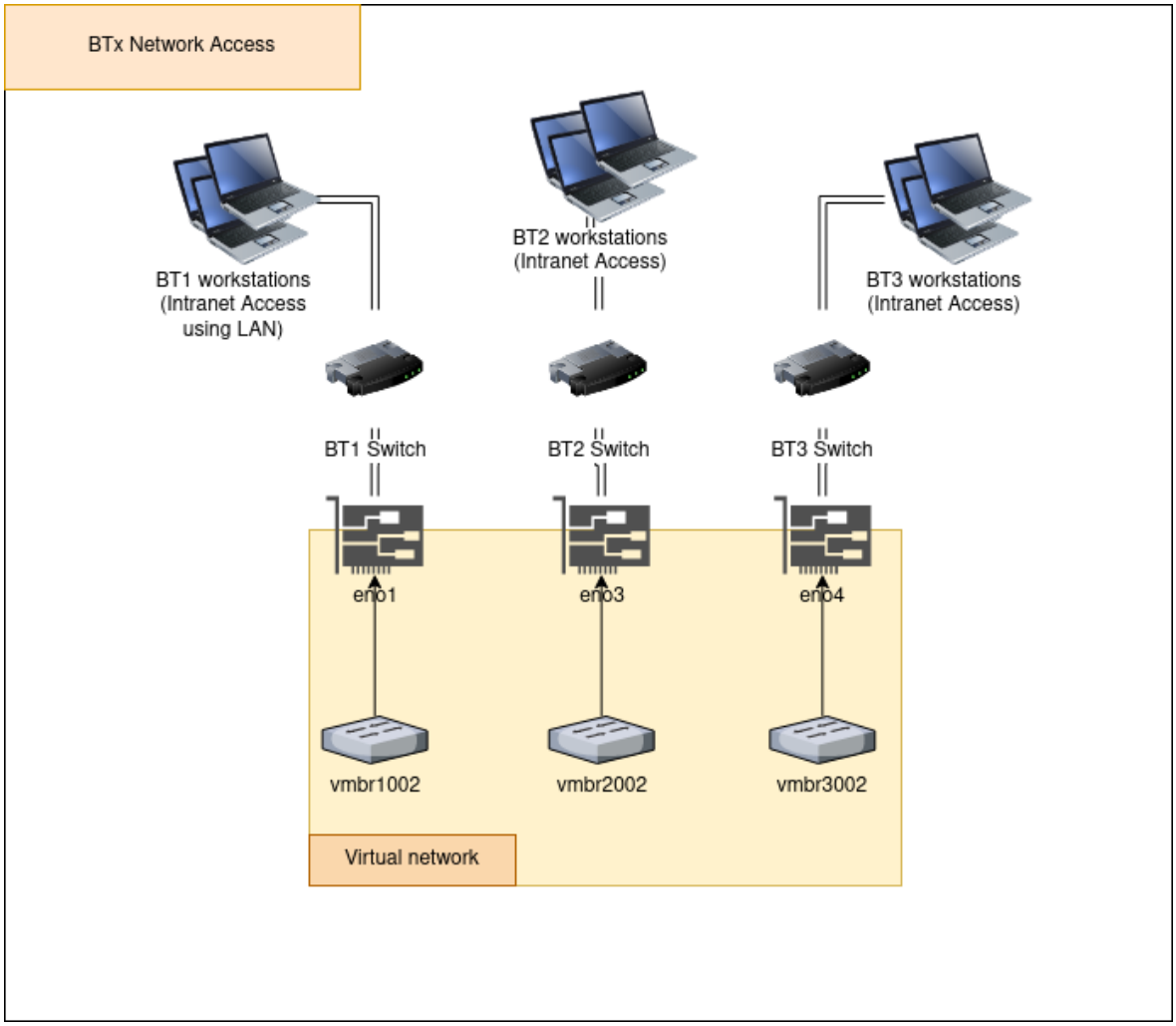
vmbr1 on virtuaaliverkon "sisäverkkoon" eli LANiin vievä kytkin. Kaikki virtuaaliverkon sisäverkosta tuleva Internetiin suuntautuva liikenne viedään vmbr1:sta vmbr0:n iptables MASQUERADE-säännön avulla. vmbr1:lle on varattu fyysinen verkkoliitäntä **ens1f0**, joka on kytketty Dell S3124P-kytkimeen. Dell-kytkimeen on kytketty WLAN-reititin, joka mahdollistaa opiskelijoiden pääsyn verkkokuvassa vihreällä taustalla olevaan Common Infrastructure Services-palvelimiin. vmbr1-kytkimelle määriteltiin oma IP-osoitealue, jotta kytkimeen voitaisiin tarvittaessa liittää laitteita, joiden täytyy päästä sekä koko peliverkkoon että Internetiin. Tätä osoitealuetta käytettiin fyysiselle Security Onion -työasemalle, jonka täytyi saada yhteys virtuaaliselle CI-Security Onion -palvelimelle.

```
iface vmbr1 inet static
address  172.3x.xx.xx
netmask  255.255.255.0
bridge-ports ens1f0
bridge-stp off
bridge-fd 0
bridge-vlan-aware yes
bridge-vids 2-4096
```

```
post-up echo 1 > /proc/sys/net/ipv4/ip_forward
post-up   iptables -t nat -A POSTROUTING -s '172.3x.xx.xx/24' -o
vmbr0 -j MASQUERADE
post-down iptables -t nat -D POSTROUTING -s '172.3x.xx.xx/24' -o
vmbr0 -j MASQUERADE
```

vmbr0 ja vmbr1 ovat virtuaaliverkon ainoat kytkimet, joille on määritelty IP-osoitteet. Kaikki muut kytkimet käsittelevät ainoastaan layer 2 -liikennettä, joiden IP-reitityksestä huolehtivat Proomun tapauksessa pfSense- ja Ubuntu-käyttöjärjestelmillä varustetut reititin-virtuaalikoneet.

Kurssin kyberharjoituksen skenaariossa haluttiin jäljitellä tilannetta, jossa yrityksen palvelukseen palkatut kyberturvallisuuskonsultit kytkevät työasemansa yrityksen intranet-verkkoon ja alkavat työskennellä sieltä käsin. Näin ollen fyysiset BT-työasemat piti sillata virtuaaliseen intranet-verkkoon. Yhdistäminen päädyttiin lopulta toteuttamaan hyvin yksinkertaisella tavalla siltaamalla fyysiset verkkoliitännät virtuaalisiin Linux-siltoihin **vmbr*002** kuvassa 7 esitellyllä tavalla.



Kuvio 7. BT-verkkojen virtuaalisen intranetin yhdistäminen fyysisille työasemille

Fyysisistä verkkoliitännöistä vietiin Ethernet-kaapeli BT-kohtaiseen hallintaominaisuudettomaan kytkimeen (*unmanaged switch*, kuvassa 7 “BTx Switch”), jolla verkko jaettiin BT-työasemien kesken. BT-työasemat kytkettiin hallintaominaisuudettomien kytkinten Ethernet-liitäntöihin. Näin ollen BT-työasemilta saattoi käyttää intranet-alueella sijaitsevia virtuaalikoneita Ethernet-yhteyttä käyttäen. WLAN-yhteyttä käyttämällä kurssilaiset pääsivät BT-työasemilta Internetiin. Tämä konfiguraatio esitellään alaluvussa 6.2.5. Virtuaalisille Linux-silloille ei määritelty `interfaces`-tiedostossa IP-osoitteita:

```
iface vmbr*002 inet manual
bridge-ports eno1
bridge-stp off
```

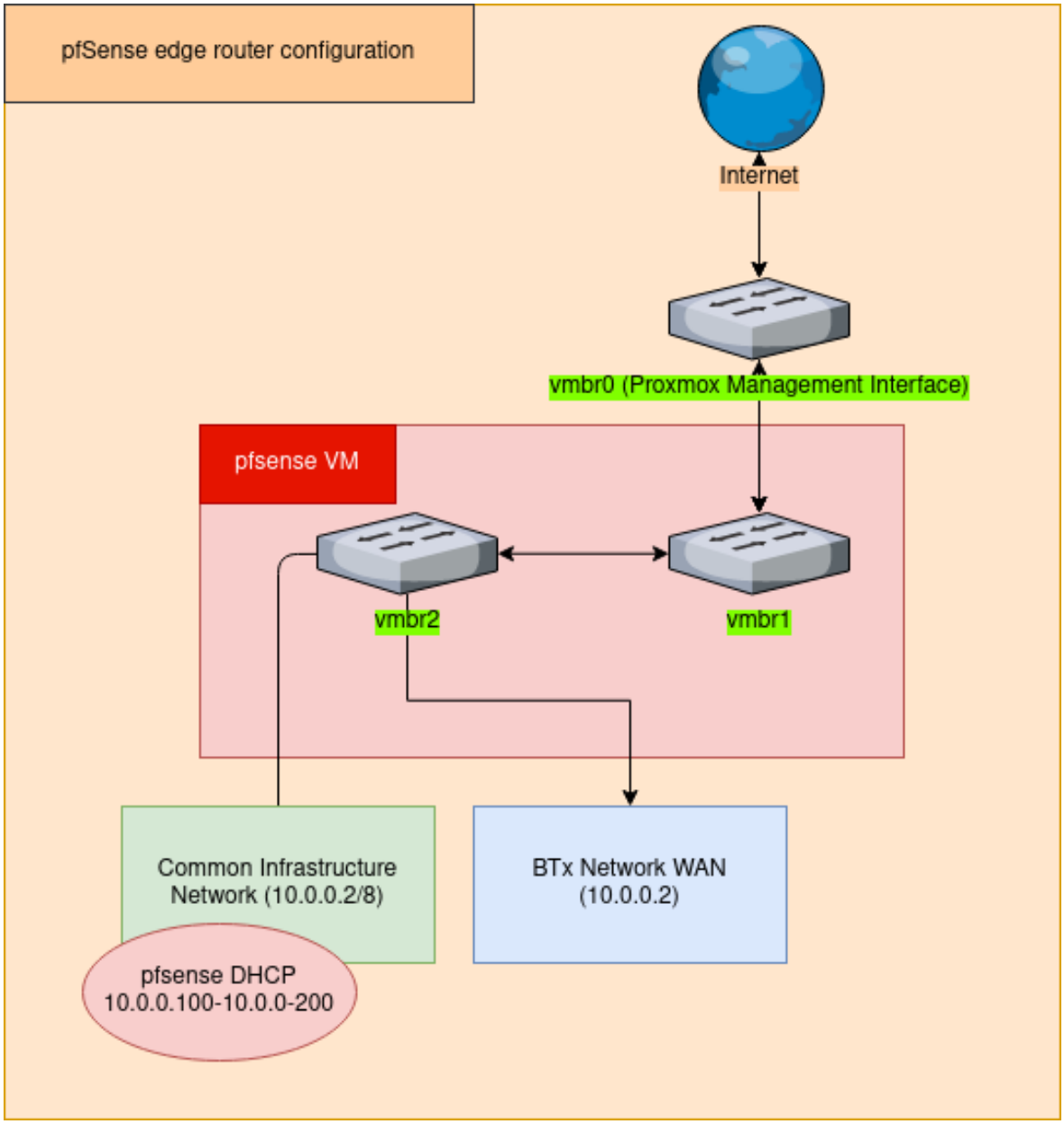
```
bridge-fd 0
```

Näin ollen **vmbr*002**-sillat huolehtivat ainoastaan L2-liikenteen välityksestä. L3-reitityksestä huolehtivat BT-verkkojen Ubuntu-reitittimet, jotka toteuttivat intranetin ja DMZ:n segmentoinnin. Näiden toteutus esitellään alaluvussa 6.2.3. Vastaavasti konfiguroitiin myös **vmbr*0**-sillat, joihin peilattiin liikennettä verkkovalvontaa varten luvussa 6.2.4 läpikäytävällä tavalla.

6.2.2 Virtuaaliset reitittimet: pfSense

pfSense on FreeBSD-käyttöjärjestelmäjakeluun perustuva verkkolaitekäyttöjärjestelmä. Se tarjoaa selaimesta käytettävän graafisen käyttöliittymän FreeBSD:n paketinsuodatusjärjestelmä `pf`:lle ja muille verkkotyökaluille. pfSense on mukautettavissa useisiin eri käyttötarkoituksiin: sitä voidaan käyttää esim. erillisenä palomuurina, WAN- tai LAN-reitittimenä, VPN- tai verkonvalvontalaitteena tai DHCP-palvelimena. pfSense on mahdollista asentaa joko fyysiseen tai virtuaaliseen palvelimeen. (Netgate 2019.)

Peliverkon reunareitittimenä käytettävä pfSense asennettiin Proxmox VE:n sisään virtuaalikoneeseen, jolle annettin kaksi virtuaalista verkkokorttia. Nämä verkkokortit yhdistettiin virtuaalisiin verkkokytkimiin **vmbr1** (WAN) ja **vmbr2** (LAN) 8. pfSense konfiguroitiin välittämään liikennettä NAT-tilassa “ulkoverkkoa” edustavasta vmbr1:sta (jota vastaavalle Linux-sillalle oli annettu IP-osoite alaluvussa 6.2 esitellyllä tavalla) LAN-verkkoalueelle 10.0.0.2/8. Tämä alue on ns. “Common Infrastructure Network”, joka näkyy kuvassa 6 vihreällä pohjalla. pfSense konfiguroitiin lisäksi tarjoamaan DHCP:llä IP-osoitteita alueelta 10.0.0.100-10.0.0.200 kuvan 8 mukaisesti. Näin ollen Common Infrastructure-verkkoon kytketyt RT-työasemat ja muut henkilökunnan koneet saivat tarvittaessa IP-osoitteen ja pääsyn peliverkkoon.



Kuvio 8. pfSense-reunareitittimen toiminnallisuus

6.2.3 Virtuaaliset reitittimet: Ubuntu

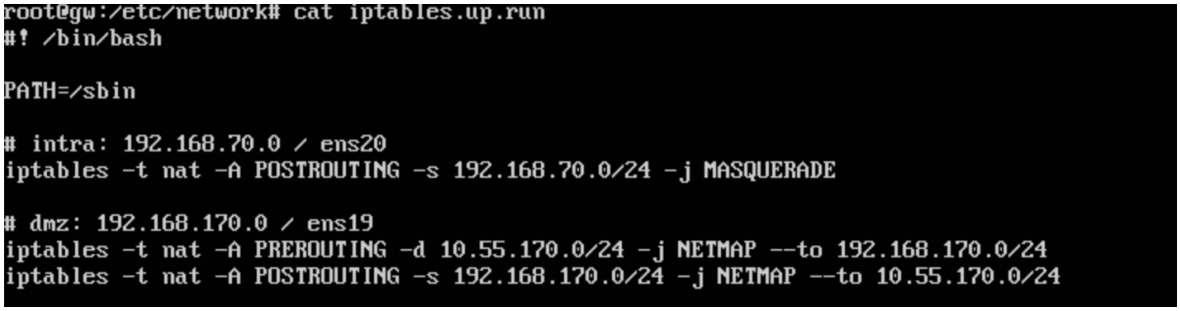
BT-verkkojen toteutuksessa havaittiin kolme uniikkia haastetta: BT-kohtaisten Security Onion -valvontapalvelinten liikennepeilauksen toteutus, fyysisten BT-työasemien siltaaminen virtuaaliverkkoon ja BT-verkkojen segmentointi DMZ- ja intranet-verkkoihin. Näistä aiheista Security Onion-peilaus käsitellään alaluvussa 6.2.4, kun taas BT-työasemien siltaami-

nen esiteltiin alaluvussa 6.2. Tässä alaluvussa käsitellään erityisesti reitityksen, verkkosegmentoinnin ja palomuurin toteutusta Ubuntun `ifupdown`:n ja `iptables`:n yhdistelmällä.

BT-verkkojen reitittimenä ja palomuurina oli aluksi tarkoitus käyttää samankaltaista pfSense-reititintä kuten koko verkon reunareitittimessäkin. Näiden reitittimien ja palomuurien toiminnan opettelu ja palomuurisääntöjen muokkaaminen kuului kurssin tavoitteisiin. Verkkosegmentoinnista aiheutuneiden teknisten ongelmien takia pfSense jouduttiin vaihtamaan Ubuntu-virtuaalikoneeseen, jossa oli mahdollista käyttää tutumpia, komentorivipohjaisia työkaluja.

BT-reitittimenä toimiville Ubuntu-virtuaalikoneille annettiin kolme verkkokorttia: **vmbr2** WAN-liitännäksi, **vmbr[10,20,30,70]01** DMZ-verkkoa ja **vmbr[10,20,30,70]02** intranet-verkkoa varten. WAN-liitännän yhdyskäytäväksi määriteltiin 10.0.0.2 eli reunareititin-pfSensen LAN-IP 8. Lisäksi **vmbr[10,20,30,70]02**-verkkoliitäntä sillattiin Proxmoxissa fyysisiin verkkoliitäntöihin **eno[1,3,4]**, jolloin intranet-verkot saatiin yhdistettyä fyysisiin BT-työasemiin luvussa 6.2 esitellyllä tavalla.

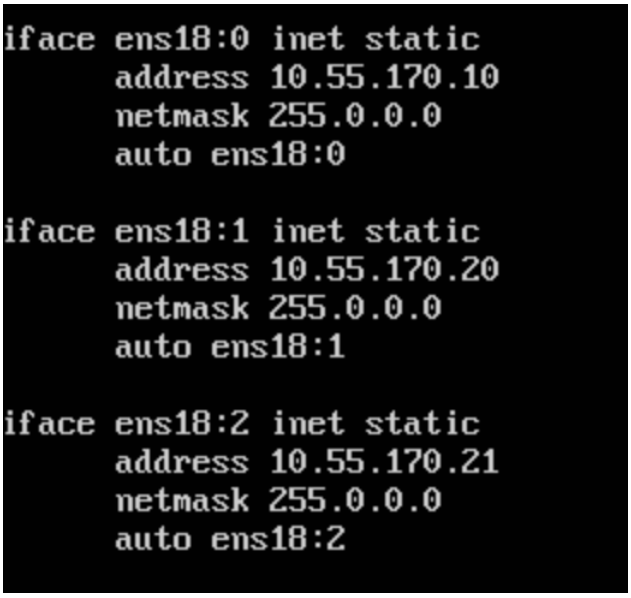
Verkkosegmentointi osoittautui varsin haastavaksi virtuaalipalvelinympäristön verkkoinfrastruktuurin puitteissa. BT-verkkojen DMZ-verkkosegmentissä sijaitsevien palvelinten tuli olla käytettävissä “ulkoverkosta” eli Common Infrastructure-verkon puolelta, jolloin Ubuntu-reitittimen tuli pystyä ohjaamaan liikennettä DMZ-verkossa oleville palvelimille. Intranet-verkossa olevien palvelinten ja työasemien puolestaan ei kuulunut näkyä ulkoverkkoon. Näihin ongelmiin keksittin ratkaisuksi kuvassa 9 esitelty `iptables`-skripti ja kuvassa 10 esitelty virtuaalikoneen `ifupdown`-verkkokorttikonfiguraatio.



Kuvio 9. iptables-skripti BT-verkkojen reitityksen toteuttamiseen

BT-verkkojen reititys toteutettiin `iptables`-skriptinä, joka sijoitettiin käynnistyksen yhteydessä ajettavaksi `rc.local`-tiedostoon. Intranet-reititys onnistui yksinkertaisesti MASQUERADE-säännöllä (ks. kuva 9), joka muuntaa intranet-verkosta tulevan verkkoliikenteen lähtevän IP-osoitteen reitittimen WAN-IP-osoitteeksi (NAT). DMZ-verkkosegmentin liikenteen täytyy kuitenkin olla molemminpuolista: siellä sijaitseviin palvelimiin täytyy päästä ulkoverkosta. Näin ollen tarvitaan NETMAP-sääntöä. NETMAP-sääntö mahdollistaa 1:1–osoitteenmuunnoksen kokonaisille verkkoalueille: toisin sanoen kuvassa 9 koko 10.55.170.0/24-verkkoalue käännetään vastaamaan verkkoalueen 192.168.170.0/24 osoitteita (Linuxtopia 2020). Tämä ei kuitenkaan yksinään riitä. Miten **vmbr[10,20,30,70]01**-verkkokortti saadaan palvelemaan DMZ-alueelle kahta eri IP-avaruutta niin, että DMZ-alueelle pääsee niin ulkoverkosta kuin intranet-verkostakin?

Ratkaisuksi päädyttiin käyttämään IP-aliasointia kuvassa 10 esitetyllä tavalla. Verkkokortille **ens18** luodaan `/etc/network/interfaces`-tiedostossa kolme aliasta, yksi jokaista DMZ-alueella sijaitsevaa palvelinta varten. Tällöin näille palvelimille ulkoverkosta tuleva liikenne pääsee reitittymään perille, ja intranet-verkosta on mahdollista myös päästä palvelimille.



Kuvio 10. IP aliasing ifupdownissa

Lisäksi Ubuntu-reittimiin asennettiin ISC `dhcpd`-palvelinohjelmisto hoitamaan BT-intranetin IP-osoitteiden jakelua DHCP:llä kurssilaisten työasemia ja virtuaalisia työasemia varten. Tähän riitti sovelluksen oletuskonfiguraatio, johon määriteltiin DHCP-palvelin toimimaan ainoastaan intranet-verkossa (verkkoliitäntä **ens20** kuvassa 9) ja määriteltiin jaettava IP-osoiteavaruus (192.168.[10,20,30,70].10-192.168.[10,20,30,70].20).

6.2.4 Verkkoliikenteen monitorointi ja peilaus: Daemonlogger ja Security Onion

Verkon valvonnan ja monitoroinnin toteuttamiseksi tarvitaan täydellinen kopio verkkoliikenteestä, jota voidaan analysoida erillään. Vaihtoehtoja tämän toteutukseen on useita: kaiken verkkoliikenteen kaikkiin portteihin jakava vanhanaikainen keskitin *(hub)*, verkkoliikenteen peilaus eli liikenteen kopiointia toiseen verkkokytkinporttiin *(SPAN port* tai *port mirroring)*, fyysinen liikenteen peilauslaite eli *network tap* tai jopa liikenteen uudelleenohjaaminen ARP-myrkytyksellä (Hjelmvik 2011). Virtualisoidussa ympäristössä työskenneltäessä erillisiä fyysisiä laitteita hyödyntävät kopiointitavat lisäävät verkkoinfrastruktuurin monimutkaisuutta ja hankaloittavat ylläpidettävyyttä.

Proxmoxin OVS-toteutus tukee ohjelmallisesti toteutettavia SPAN-portteja, kuten Vext.info (2018) toteaa. Linux-siltoihin pohjautuvaa verkkokonfiguraatiota käytettäessä vaihtoehdot

ovat rajallisempia: joko

1. porttipeilauksen toteuttaminen Proxmoxin ulkopuolisella fyysisellä verkkokytkimellä,
2. liikenteenohjaus käyttöjärjestelmäydintasolla Linux `tc`-työkalua käyttäen tai
3. erillisen porttipeilausohjelmiston (esimerkiksi Daemonlogger) käyttö.

Proomun Linux-siltojen peilaamiseen valittiin aikataulujen ja käytössä olevien laitteistoresurssien takia kolmas vaihtoehdoista eli Daemonlogger. **Daemonlogger** on Martin Roeschin kehittämä avoimen lähdekoodin paketinkaappausohjelmisto. Sen avulla on mahdollista kaapata verkkoliikennettä tietystä verkkoliitännästä ja kopioida se sellaisenaan toiseen verkkoliitäntään (ns. *soft tap*). (Roesch 2017.) Daemonloggerin käyttö onnistuu yksinkertaisimmillaan seuraavasti:

```
daemonlogger -d -i <MITÄ_PEILATAAN> -o <MIHIN_PEILATAAN>
```

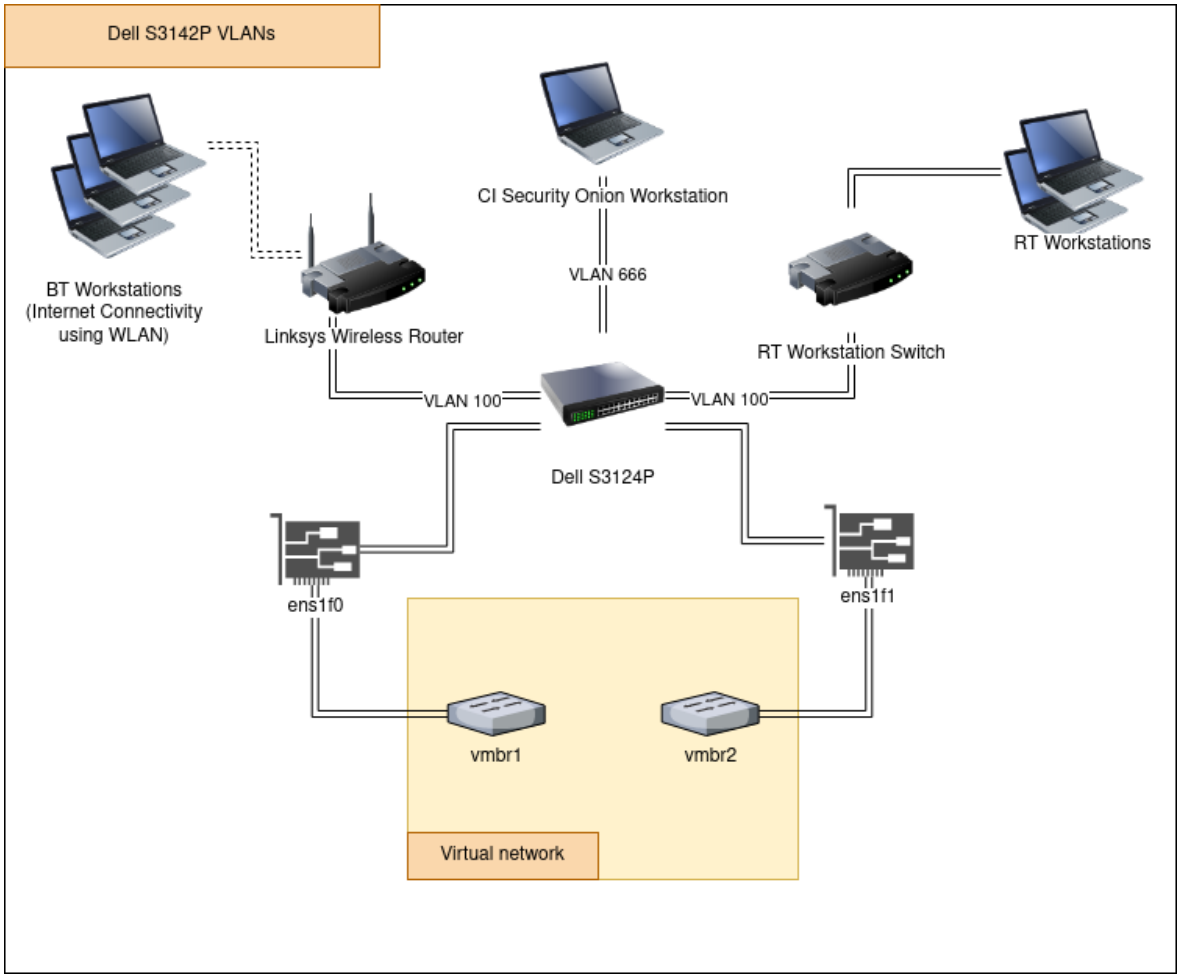
`-d`-parametri käynnistää Daemonloggerin taustalle daemon-tilassa. Käynnistyksen jälkeen sovellus peilaa kaiken verkkoliitäntään sisäänpäin tulevan liikenteen haluttuun ulostuloon. Jos peilaus halutaan käynnistää automaattisesti järjestelmän käynnistyksen yhteydessä, voidaan se toteuttaa `rc.local`-tiedostosta ajettavalla yksinkertaisella skriptillä, kuten Proomun tapauksessa tehtiin.

```
#!/bin/bash
#BT-verkkojen INTRA ja DMZ (vmbr*00*) peilataan vmbr**-liitän-
töihin
daemonlogger -d -i vmbr1001 -o vmbr10
daemonlogger -d -i vmbr1002 -o vmbr10
daemonlogger -d -i vmbr2001 -o vmbr20
daemonlogger -d -i vmbr2002 -o vmbr20
daemonlogger -d -i vmbr3001 -o vmbr30
daemonlogger -d -i vmbr3002 -o vmbr30
#vmbr2-verkkoliikenne (Common Infra + BT-verkot) monitorointi-
peilaus
daemonlogger -d -i vmbr2 -o vmbr666
```

Kun peilaus on toteutettu, voidaan peilatut verkkoliitännät (`vmbr*00*`) liittää Proxmoxissa Security Onion-virtuaalikoneeseen, josta ne voidaan konfiguroida kuunteluliittymiksi (*packet sniffing interface*). Näistä kuunteluliittymistä Security Onion saa runko- ja BT-verkkojen verkkoliikennettä, jota on mahdollista prosessoida ja analysoida Security Onion:in sisältämillä työkaluilla.

6.2.5 Dell S3124P ja muut fyysiset verkkolaitteet

ITKST55-kurssilla käytettiin Dell R740-palvelimen Intel X710-verkkokortin SFP+-liitäntöjen jakamiseen Dell S3124P-kytkintä. Kytkimeen konfiguroitiin 2 VLANia: kuvan 11 mukaiset **VLAN 100** ja **VLAN 666**. VLAN 100 tarjosi RT:n työasemille reitin sisälle peliverkkoon ja Internetiin, ja BT-työasemille reitin Internetiin WLAN-yhteyttä käyttäen. WLAN-yhteyden tarjoava Linksys-reititin konfiguroitiin toimimaan siltaavassa tilassa, jolloin siihen kytkeytyneet fyysiset työasemat saivat IP-osoitteet Common Infrastructure -alueen DHCP-palvelimelta (pfSense-reunareititin, esitelty alaluvussa 6.2.2). RT-työasemat kytkettiin hallintaominaisuudettomaan kytkimeen, joka puolestaan liitettiin S3124P-kytkimeen VLAN 100:lle määriteltyyn porttiin. Tällöin RT-työasemat toimivat samalla tavoin kuin muutkin WLAN:iin kytketyt työasemat. VLAN 666 puolestaan tarjosi Security Onion -työasemalle pääsyn peliverkkoon, jotta se pystyi käyttämään virtuaalista Security Onion -palvelinta ja esittämään sen liikennettä luokan edessä olevalla televisiolla visualisointitarkoituksessa.



Kuvio 11. Dell S3124P-kytkimen VLANit ja niiden käyttö

VLAN 100 liitettiin verkkokortin liitäntään **ens1f1**, joka puolestaan oli kytketty virtuaaliseen verkkokytkimeen **vmbr2**. Näin ollen VLAN 100:n luokiteltuun porttiin kytkeytymällä pääsi käyttämään peliverkon sisäverkkoa ja Internetiä. VLAN 666 puolestaan kytkettiin liitäntään **ens1f0**, joka kytkettiin virtuaaliseen verkkokytkimeen **vmbr1**, jolloin Security Onion -työasema on mahdollista yhdistää vmbr1:n kytkettyyn CI- Security Onion -palvelimeen. Security Onion -työasemalle määriteltiin kiinteä IP-osoite vmbr1:n osoitealueelta tämän mahdollistamiseksi.

On olennaista huomioida, että vuoden 2019 ITKST55-toteutuksen virtuaalipalvelinympäristö ei erityisesti vaatinut Dell D3124P -kytkimen ominaisuuksia. Voidaan jopa sanoa, että kytkintä käytettiin yksinkertaisena SFP+-RJ45 -liitinmuuntimena - jos Dell R740-palvelimessa

olevassa verkkokortissa olisi ollut useampi Ethernet-liitäntä, olisi ympäristö voitu toteuttaa kokonaan ilman kytkimen käyttöönottoa. Kytkintä päädyttiin käyttämään Proxmoxin verkkokonfiguraatiossa ilmenneiden VLAN-konfiguraation teknisten ongelmien vuoksi. Tulevilla kurssitoteutuksilla ja muissa tulevissa palvelimen käyttötilanteissa Proxmoxin verkkokonfiguraatiota ja VLAN:ien siltaamista fyysisiin verkkolaitteisiin on syytä tarkastella huolella, jotta Dell S3124P:n ominaisuuksia voidaan käyttää tehokkaammin.

6.3 Virtuaalipalvelimet ja -työasemat

Virtuaalipalvelinympäristön opiskelijoille näkyvin osa lienee varsinaiset virtuaalikoneet, joita kurssin aikana käytetään. Seuraavaksi esitellään lyhyesti ne virtuaaliset työasemat ja palvelimet, jotka toteutettiin Proomu-alustalla ITKST55-kurssin tarpeita varten. Kurssin käytännön opetussisällön suojelemiseksi palvelimille asennettujen sovellusten ja käyttöjärjestelmien tarkkoja ohjelmistoversioita ja haavoittuvuuksia ei voida käsitellä syvällisemmin tämän tutkielman puitteissa. Virtualisoidut verkkolaitteet ja niiden toiminta on esitelty alaluvussa 6.2.2.

Kyberharjoituksessa tarvitaan usein joitakin yleisiä verkkopalveluja, jotka on hyvä toteuttaa peliverkon sisällä. Vuoden 2019 ITKST55-kurssilla yleiset verkkopalvelut toteutettiin nk. *Common Infrastructure*-alueelle 6, joka esitti BT-verkkojen näkökulmasta ulkoverkkoa eli Internetiä. Tarjottuja verkkopalveluja olivat globaali DNS/NTP-palvelin, WWW-palvelin, yleinen sähköpostipalvelin, tiedostojakopalvelin ja IRC-palvelin. WWW-palvelimella ajettiin Dokuwiki-palvelua, johon kurssilaiset pitivät kirjaa BT-verkoissa tekemistään toimenpiteistä ja Gnusocial-nimistä sosiaalisen median palvelua, jossa luotiin kyberharjoituksen skenaarioon kuuluvia syötteitä ja elävöitettiin skenaarion kerrontaa. IRC-palvelin pystytettiin harjoituksen aikana opiskelijoiden toiveesta välittömäksi kommunikointikanavaksi, jossa BT-joukkueet saattoivat jakaa toistensa kanssa esimerkiksi järjestelmistä löytyneitä lokiviestejä. Kaikki Common Infrastructure-palvelimet toteutettiin **Ubuntu Server**-käyttöjärjestelmällä.

Pelikäyttöön tarkoitettujen palvelujen lisäksi Common Infrastructure-alueella sijaitsi nk. yleinen Security Onion -valvontapalvelin, johon peilattiin kaikki Proxmox-

virtuaalipalvelinympäristön sisällä tapahtuva verkkoliikenne luvussa 6.2.2 kuvatulla tavalla. Yleinen Security Onion oli ensisijaisesti visualisointi- ja opetustyöväline, jota pidettiin jatkuvasti näkyvillä opetustilan edessä olevassa televisiossa. Toissijaisesti valvontapalvelinta käytettiin verkon tilanteen ja kuormituksen tarkkailuun kurssin henkilökunnan toimesta.

Vuonna 2019 ITKST55-kurssilla jokainen BT sai suojattavakseen neljä palvelinta, jotka toteuttivat infrastuktuuripalveluja skenaariossa määritellyn yrityksen henkilökunnan ja asiakkaiden käyttöön. Palvelimet jaettiin kahteen verkkosegmenttiin: kuvassa 6 DMZ-vyöhykkeellä sijaitsivat yrityksen DNS/NTP-palvelin, yrityksen verkkosivuja ajava WWW-palvelin ja sähköpostipalvelin. Yrityksen intranetissä 6 puolestaan sijaitsi yrityksen sisäinen tiedostonjakopalvelin. Näiden lisäksi BT:t saivat käyttöönsä joukkuekohtaisen Security Onion-valvontapalvelimen, jonka verkkoliitäntään oli peilattu BT-verkossa käytävä verkkoliikenne alaluvussa 6.2.4 esitellyllä tavalla. Kaikki DMZ-vyöhykkeen palvelimet toteutettiin **Ubuntu Server**-käyttöjärjestelmällä. Intranet-alueen tiedostonjakopalvelin puolestaan rakennettiin **Windows Server 2012 R2 Standard**-käyttöjärjestelmälle.

Palvelinten lisäksi BT-kohtaisissa intranet-verkossa sijaitsi viisi virtuaalista Windows 7-työasemaa, jotka esittivät yrityksen työntekijöiden tietokoneita. Näiltä työasemilta ajettiin verkkoliikennettä generoivaa skriptiä, joka simuloi yrityksen työntekijöiden tavanomaista verkkoliikennettä, kuten internet-sivujen selaamista. Kurssilaiset pääsivät käyttämään työasemia Proxmoxin verkkoselainkäyttöliittymästä löytyvästä VNC-konsolista.

7 Vuoden 2019 kurssitoteutus käytännössä

Tässä luvussa käydään läpi kurssin intensiivijakson kulku ja kurssin aikana Proomu-virtuaalipalvelinympäristössä havaitut ongelmat. Lisäksi esitellään kurssilla syntyneitä toiveita ja ajatuksia tulevien kurssitoteutusten varalle. Tämän luvun sisältö on osa Peffers ym. (2007) esittämän prosessimallin demonstraatio- ja evaluointivaiheita.

7.1 Kurssin eteneminen intensiivijakson aikana

Kurssi järjestettiin viikon mittaisena intensiivikurssina 9.9.2019-13.9.2019. Intensiivijakson aikana opetusta järjestettiin klo 8.00–20.00. Kurssipäivät koostuivat luennoista, tehtävistä ja aktiivisesta pelivaiheesta, jossa BT:t saivat toteuttaa järjestelmän puolustukseen tähtääviä toimia ja RT sai hyökätä BT:iden suojaamiin järjestelmiin. Tietyt ajanjaksot sovittiin "hyökkäyksettömäksi" ajaksi valmistautumista, tehtävien tekoa tai luentoja varten.

Kurssi alkoi maanantaina aloitusluennolla ja opiskelijoiden ryhmäytymisellä BT-joukkueisiin. Tämän jälkeen aloitettiin järjestelmän haltuunotto: BT:ille esiteltiin heidän edustamansa yritykset ja näiden ydintoiminta, jonka jälkeen he pääsivät tutustumaan yrityksen tietotekniseen ympäristöön. Koska kurssijärjestelmä oli vuonna 2019 ensimmäistä kertaa virtualisoitu, varattiin järjestelmän haltuunottoon enemmän aikaa kuin aiemmin. BT:t muodostivat kuvan yritysverkossa toimivista palvelimista ja työasemista ja tekivät alustavan suunnitelman sen suojaamiseksi. Tämän jälkeen BT:ille annettiin aikaa suojaamissuunnitelman toteuttamiseen käytännössä: esimerkiksi ohjelmistopäivityksien asentamiseen, salasanojen vaihtamiseen ja palomuurisääntöjen tiukentamiseen.

Ennakkosuojaamisen jälkeen RT:lle annettiin "lupa" aloittaa hyökkääminen. Tästä alkoi nk. varsinainen pelivaihe, jossa RT hyökkäsi ja BT:t pyrkivät tunnistamaan hyökkäykset ja minimoimaan niiden vaikutuksen yrityksen arkipäiväiseen toimintaan. BT:t saivat vapaasti jakaa keskenään järjestelmistä löytyneitä merkkejä väärinkäytöksistä (*IoC, indication of compromise*) ja pohtia, miten hyökkäyksiin pitäisi reagoida. Skenaarioon liittyvää tarinankerrontaa toteutettiin esimerkiksi peliverkosta löytyneessä GNUSocial-nimisessä sosiaalisen median palvelussa, jossa yritysten asiakkaat saattoivat esimerkiksi ihmetellä yritysten verkkosivujen

toimintaongelmia. Muut kurssipäivät etenivät vastaavalla kaavalla: luentoja erilaisista SoC-toimintaan liittyvistä aiheista, kirjallisia tehtäviä ja aktiivista pelivaihetta.

Kurssipäivät päättyivät niinkutsuttuun *hotwash* -sessioon, jossa käytiin läpi RT:n päivän aikana tekemät hyökkäykset, BT:iden tekemät suojaustoimet ja keskusteltiin yleisesti päivän tapahtumista ja kokemuksista. Näissä sessioissa myös tuotiin esille päivän aikana ilmenneitä puutteita ja toiveita niin kurssin teknisissä kuin muissakin puitteissa.

7.2 Toteutuksessa havaitut ongelmat ja toiveet tuleville kursseille

Suurimpia Proomu-virtuaalipalvelinympäristössä ilmenneitä ongelmia oli, ettei *red team*:ille ehditty rakentaa ajoissa omaa BT-verkon kopiota testiympäristöksi. Syynä tähän olivat ongelmat ulkoisen Dell EMC S3124P -kytkimen porttikonfiguraation kanssa, joita ei saatu ratkaistua aikataulun puitteissa. Tämän vuoksi RT:lla ei ollut mahdollisuutta testata suunniteltuja hyökkäyksiä, vaan varsinainen testaus tapahtui suoraan “tuotantoympäristössä”. Kurssin seuraavia toteutuksia suunniteltaessa testiympäristön toteuttaminen täytyy ottaa huomioon jo varhaisessa vaiheessa, jotta vastaavaa ei pääse tapahtumaan.

Toinen suuri ongelma Proomu-järjestelmässä ilmeni virtuaalisen ja fyysisen verkon segmentoinnissa. Alaluvussa 6.2 esitelty verkkoratkaisu vaati palvelimelta kaksi verkkokorttia ja yhteensä kuusi yksilöllistä verkkoliitäntää. Tämä tekee ratkaisusta erityisen heikosti skaalautuvan. Esimerkiksi verkon VLAN-pohjainen segmentointi Proxmox VE:n sisällä mahdollistaisi verkkoliitäntöjen kapasiteetin tehokkaamman hyödyntämisen. Tämä riittäisi luultavasti ITKST55:n mittakaavan pienelle peliverkolle, mutta laajemmassa harjoitus- tai kurssikäytössä on syytä pohtia muitakin ratkaisuja. Kim, Maeng ja Jang (2019) toteavat, että VLAN-segmentoinnin sijaan skaalautuvuutta olisi hyvä tavoitella NFV (*network function virtualization*)- ja SDN -verkkoratkaisuilla. Yamin, Katt ja Gkioulos (2020) mainitsevat myös SDN-verkot ratkaisuna kyberlaboratorioiden skaalautuvuusongelmiin. ITKST55:n ja Proomun rakennusaikataulujen puitteissa skaalautuvaa verkkoratkaisua ei kuitenkaan valitettavasti saatu toteutettua. Tulevilla kurssitoteutuksilla tähän olisi hyvä perehtyä mahdollisimman alusta alkaen.

Virtuaalisten Windows-työasemien käyttö vaati Proomu-virtuaalipalvelinympäristössä

Proxmoxiin sisäänrakennetun selainpohjaisen noVNC-konsolin käyttöä. Selainkonsoli toimi olennaisilta osin, mutta työaseman ja virtuaalikoneen välisen leikkaa/liimaa-toiminnallisuuden puute havaittiin ongelmalliseksi. Lisäksi kurssin tavoitteiden kannalta ei välttämättä ole mielekästä, että kurssilaiset joutuvat käyttämään aikaa kurssin tavoitteiden ulkopuolella olevan ohjelmiston käytön opetteluun. Tämä vaikuttaa myös alaluvussa 5.3 esiteltyyn läpinäkyvyyden vaatimuksen toteutumiseen. Tulevilla kursseilla virtuaalisten työasemien etäkäyttö tulee suunnitella paremmin: esimerkiksi Windows RDP- tai SPICE -pohjainen ratkaisu voisi olla mahdollinen.

Kurssin aikana havaittiin myös, ettei tiedostojen siirtoa virtuaalisille Windows 7-työasemille oltu suunniteltu kunnolla. Virtuaalityöasemissa ei siis oltu muistettu lisätä hallinnollista "takaporttia", jonka kautta niihin voitaisiin tarvittaessa lisätä sovelluksia tai skenaarion kannalta tarpeellisia asiakirjoja ym. kurssin aikana. BT-reitittimien harjoituksen osana kiristetyt palomuurit aiheuttivat sen, ettei tiedostojen siirtäminen verkon välityksellä onnistunut suunnitellulla tavalla. Näin ollen virtuaalikoneet jouduttiin hetkellisesti piilottamaan kurssilaisten näkyviltä Proxmoxin hallintaliittymästä ja tarvittavat tiedostot siirtämään virtuaalikoneisiin syötetyillä ISO-levykuvilla. Lisäksi virtuaalikoneiden ylläpidossa havaittiin useita tilanteita, joissa tietyt yksinkertaiset toimenpiteet, kuten IP-osoitteiden ja isäntänimen vaihdon olisi voinut automatisoida. Yleisesti ottaen järjestelmän suunnitteluun olisi syytä ottaa alusta alkaen mukaan työasemiin ja palvelimiin tavalla tai toisella toteutettavat "ylläpitäjien takaportit", joiden kautta tällaisia muutoksia voitaisiin tehdä. Yksi mahdollinen ratkaisu olisi konfiguraationhallintaohjelmiston, kuten Ansible tai Puppet käyttöönotto virtuaalikoneiden hallinnan helpottamiseksi.

Runkoverkon Security Onion -palvelimen dashboard-työkalu Kibana kaatui kerran kurssin aikana tuntemattomasta syystä. Palvelimesta oli kuitenkin luotu palautuspiste (*snapshot*), joten hajonneen virtuaalikoneen tilalle voitiin kloonata uusi toimiva versio palautuspisteen pohjalta. On huomautettava, että Security Onion -virtuaalikone säilöi verkosta kerättyä lokidataa omalla kiintolevyllään. Uuden virtuaalikoneen kloonaamisesta ja vanhan sammuttamisesta aiheutui siis pieni katkos verkosta kerättyyn valvontadataan, ja osa valvontadatasta jäi hajonneen virtuaalikoneen kiintolevylle. Tulevien kurssitoteutusten kannalta voisi olla mielekästä säilöä Security Onionin keräämää dataa toisella tiedostopalvelin-virtuaalikoneella,

jotta palvelin voidaan palauttaa tarvittaessa ilman katkoksia ja datan silpoutumista useiden virtuaalikoneiden tai niiden palautuspisteiden välille.

Aivan kaikkia virtuaalipalvelinympäristöön toivottuja palveluja ja niihin kohdistuvia hyökkäyksiä ei saatu toteutettua aikataulujen ja puutteellisen osaamisen takia. Näistä merkittävin oli Windows Active Directory -hakemistopalvelun puuttuminen kokonaan. Kaupallisten ohjelmistojen lisensointi kyberturvallisuuden opetuskäyttöön voi olla poikkeuksellisen vaikeaa - erityisesti silloin, jos tarvitaan vanhoja tai haavoittuvaisia ohjelmistoversioita. Proomu-järjestelmän tapauksessa lisenssiavainten saannissa kesti kauan, eikä jäljelle jääneessä aikataulussa ehditty rakentaa halutunlaista hakemistopalvelua. Näin ollen tulevilla kurssitoteutuksilla on syytä varmistaa varhaisessa vaiheessa, että erityisosaamista vaativilla laajemmilla ohjelmistokokonaisuuksilla on selkeästi määritelty vastuuhenkilö ja toteutusaikataulu, ja ohjelmistokokonaisuuden vaatimat tekniset resurssit, kuten lisenssit ovat saatavilla.

7.3 Kurssitoteutuksen evaluointi

Tässä alaluvussa käydään läpi luvussa 5 esitellyt ITKST55-kurssin asettamat vaatimukset ja tiivistetään niiden toteutuminen Proomu-virtuaalipalvelinjärjestelmässä aiempiin alalukuihin viitaten. Tavoitteena on luoda yhteenveto siitä, miltä osin toteutettu artefakti täytti sille alaluvuissa 5.3, 5.4 ja 5.5 asetetut vaatimukset. Vaatimukset käsitellään alaluvun 5.2 esittelemässä kolmessa pääkategoriassa: opetussisällön muodostamat, opetustekniset ja tekniset vaatimukset.

7.3.1 Opetussisällön muodostamien vaatimusten toteutuminen

Luokkatilaan saatiin rakennettua verkkokuvan kaltainen fyysisten työasemien ja verkkolaitteiden kokonaisuus (vaatimus **OS1**). Opiskelijat saivat käyttöönsä työasemat, joilta löytyivät intensiivijaksolla tarvittavat työkalut ja lisäksi mahdollisuus lisäohjelmistojen asentamiseen (vaatimukset **OS2** ja **OS3**). Lisäohjelmistojen asennusmahdollisuutta työasemiin käytettiin esimerkiksi IRC-asiakasohjelmien asennukseen kurssin aikana pystytettyä IRC-palvelinta varten. Järjestelmän haltuunotossa ei havaittu merkittäviä ongelmia - verkon skannaus ja kartoitus eivät vieneet merkittävästi enempää aikaa aiempiin kurssitoteutuksiin verrattuna.

Virtualisointiympäristön läpinäkyvyys (vaatimus **OS4**) ei onnistunut aivan täysin alaluvussa 7.2 selitettyjen tiedostonsiirtoon liittyvien teknisten ongelmien takia. BT-virtuaalityöasemia jouduttiin siirtämään hetkellisesti pois opiskelijoille näkyvästä Proxmox-selainkäyttöliittymästä, jotta niihin saatiin siirrettyä harjoitukseen kuuluvaa haittadataa. Tämä oli lopulta varsin pieni ja lyhytkestoinen ongelma, mutta sen olemassaolo kertoo laajemmalti järjestelmän hallittavuudesta ja ylläpidettävyydestä - virtuaalipalvelinjärjestelmään tarvitaan paremmin suunniteltu hallintaprosessi. Tähän aiheeseen palataan luvussa 8.

7.3.2 Opetusteknisten vaatimusten toteutuminen

Kurssin kaikki osallistujat olivat paikan päällä ja tila saatiin jaettua järkevästi niin, että joukkueiden työasemat saatiin kytkettyä Proomuun ja verkkolaitteisiin. *Red team* sijoitettiin verholla muusta tilasta erotettuun tilaan, josta saatiin toteutettua tietoliikenneyhteydet Proomu-järjestelmään vaatimusten **OT1** ja **OT2** mukaisesti. Tulevilla kurssitoteutuksilla voisi olla mahdollista, että RT osallistuu harjoitukseen etäyhteydellä yliopiston ulkopuolelta tai toisesta tilasta yliopiston sisällä, jolloin joukkueet saadaan paremmin eroteltua toisistaan. Tämä voi olla tärkeää, jos mukaan halutaan tulevaisuudessa pelillisiä/kilpailullisia elementtejä.

Peliverkon havainnollistus saatiin toteutettua vaatimuksen **OT3** mukaisella tavalla tilannekuva-työasemalla ja Security Onion -käyttöjärjestelmäjakelulla. Valvontatyöaseman ja -palvelimen toteutus käytiin läpi alaluvuissa 6.2.1 ja 6.2.4. Tilannekuvatyöasema kytkettiin televisioon ja sitä käytettiin aktiivisesti peliverkossa tapahtuvan liikenteen havainnollistamiseen. Kurssin aikana Security Onion -palvelin kaatui kerran alaluvussa 7.2 kerrotulla tavalla, mutta se saatiin palautettua nopeasti toimintakuntoon virtuaalipalvelinjärjestelmässä otettujen palautuspisteiden ansiosta.

Kurssilaiset pääsivät käyttämään BT-kohtaisia valvontavirtuaalikoneita valvontatyöasemilta vaatimuksen **OT4** mukaisesti, mutta valvontavirtuaalikoneita ei oltu sijoitettu omaan verkkosegmenttiinsä tai VLAN:iinsa, joten valvontavirtuaalikoneen toiminta oli ei-toivotulla tavalla sidoksissa BT-verkon toimintaan. Tästä syystä vaatimus **OT4** lasketaan osittain onnistuneeksi.

7.3.3 Teknisten vaatimusten toteutuminen

ITKST55:n vaatima virtuaaliverkon segmentointi saatiin toteutettua toiminnallisesti yleistä infrastruktuuria varten (vaatimus **T1**). Jokainen BT sai käyttöönsä oman BT-verkon, joka oli toteutettu samoista virtuaalikonepohjista, täyttäen täten vaatimuksen **T2**. Sekä BT- että RT-joukkueet pääsivät yhdistymään vaatimusten 5.5 mukaisiin verkkosegmentteihin vaatimusten **T3** ja **T4** mukaisesti. Common Infrastructure -verkko eli peliverkon Internet saatiin toteutettua luvussa 6.2 tarkemmin esitellyllä tavalla. CI-verkkoalueelle saatiin rakennettua kuvan 6 mukaiset palvelut, joita käytettiin harjoituksen aikana. On kuitenkin huomioitava, että verkkosegmentoinnin toteutustapa ei ollut ihanteellinen - jokainen BT-verkko tarvitsi tässä toteutuksessa oman verkkoliitännän fyysisten BT-työasemien kytkemiseksi BT-virtuaaliverkkoon.

Peliverkon internet-yhteys saatiin toteutettua halutulla tavalla BT-työasemille vaatimuksen **T5** edellyttämällä tavalla. Työasemilta pääsi Ethernet-yhteyttä käyttäen BT-intranet-verkkoihin ja WLAN-yhteyttä käyttäen Internetiin. Peliverkon ja Internetin välillä oli palomuuri estämässä haitallisen liikenteen peliverkosta ulospäin, täyttäen vaatimuksen **T6**. Opiskelijat pystyivät siis hakemaan ratkaisuja ongelmiin ja asentamaan päivityksiä virtuaalikoneisiin. Opiskelijoille ei kuitenkaan ehditty luoda omaa tietovarastoa, josta päivityksiä olisi voinut asentaa ilman Internet-yhteyttä. Tällaista tietovarastoa voitaisiin käyttää verkkoliikenteen määrän rajoittamiseen ja lisäksi mahdollisesti myös kurssilaisten työskentelyn ohjaamiseen: jos varastossa on saatavilla ainoastaan tiettyjä työkaluja ja päivityksiä, täytyy niitä myös käyttää harjoituksessa ilmenevien ongelmien ratkaisuun.

Yleinen valvontavirtuaalikone saatiin toteutettua vaatimusta **T7** vastaavasti. BT-valvontavirtuaalikoneille saatiin peilattua BT-intranet- ja DMZ-verkoissa tapahtuva verkkoliikenne vaatimuksen **T8** mukaisesti. Opiskelijat pystyivät seuraamaan näitä Security Onion -työkaluja käyttäen BT-työasemilla (vrt. luvussa 7.3.2 käsitelty vaatimus **OT4**). Kuitenkin intranet-verkon toteutustavasta johtuen BT-työasemat sijaitsivat itseasiassa intranet-verkkosegmentissä, eivätkä esimerkiksi omassa verkkosegmentissään tai omassa VLAN:ssaan. Tämä teki niiden toiminnan riippuvaiseksi intranet-verkon normaalista toiminnasta, mikä ei ole valvontaa toteutettaessa toivottavaa. Tulevissa kurssitoteutuksissa BT-valvontapalvelin ja BT-työasema on sijoitettava omaan VLAN:iinsa tai segmentoitava

muuten erilleen. Näin ollen kuten vaatimus **OT4**, myös vaatimus **T8** lasketaan osittain onnistuneeksi.

BT-joukkueet saivat pääkäyttäjäoikeudet kaikkiin joukkuekohtaisen BT-verkkonsa virtuaalikoneisiin (vaatimus **T9**). Virtualisointialustaan pääsy jouduttiin sallimaan alaluvussa 7.2 esitetyllä tavalla Windows-virtuaalityöasemien käyttöä varten, mikä täytti vaatimuksen **T10**. Kaikista peliverkossa olleista virtuaalikoneista otettiin säännöllisesti palautuspisteitä ennen kurssia, sen aikana ja sen jälkeen, jolloin vaatimus **T11** täyttyi. Palautuspisteitä jouduttiin käyttämään kurssin aikana muutamia kertoja, merkittävimpänä esimerkkinä alaluvussa 7.2 esitelty Security Onion -palvelimen kaatuminen.

7.3.4 Evaluoinnin yhteenveto

Evaluoinnin perusteella todettiin, että Proomu-virtuaalipalvelinjärjestelmä täytti valtaosan sille asetetuista vaatimuksista. Kaksi toisiinsa läheisesti kytkeytyvää vaatimusta, **OT4** (tiimikohtaisten valvontavirtuaalikoneiden käyttö BT-työasemilta) ja **T8** (BT-valvontavirtuaalikoneisiin on peilattava BT-verkon DMZ- ja intranet -liikenne) laskettiin osittain toteutuneiksi, sillä vaikka toteutunut toiminnallisuus riitti kurssin opetustavoitteiden täyttöön, vaatimusten tekninen toteutus paljastaa järjestelmässä merkittävän suunnitteluvirheen. Vaatimus **OS4** (virtuaalipalvelinympäristön läpinäkyvyys) puolestaan ei toteutunut Proomu-järjestelmässä lainkaan, sillä kurssin aikana jouduttiin hetkellisesti estämään kurssilaisten pääsy tietyille virtuaalikoneille luvussa 7.2 kuvatulla tavalla. Tämäkin puute voidaan laskea suunnitteluvirheeksi, johon on syytä kiinnittää huomiota mahdollisilla tulevilla kurssi-instansseilla.

8 Pohdinta

Teknisen kyberharjoituksen, pienimuotoisenkin sellaisen kuten ITKST55:n kaltaisen järjestämisessä joudutaan tasapainottelemaan monien tekijöiden välillä: realismi, aikataulurajoitteet, opetukselliset tavoitteet sekä henkilö-, laitteisto- ja ohjelmistoresurssit. Tässä luvussa pyritään tuomaan ilmi, millaisena ITKST55:n lopullinen toteutus näyttäytyy toteutuneen evaluoinnin ja aiemman tutkimuksen valossa.

Traficom (2019) mukainen simuloitu kyberharjoitusympäristö soveltui erinomaisesti ITKST55:n tarpeisiin. Käyttötarkoitusta varten rakennetussa järjestelmässä oli mahdollista toteuttaa monipuolisia hyökkäys- ja puolustustapahtumia. Erityisesti kiitosta sai järjestelmän joustavuus: virtuaalikoneiden suorituskykyä voitiin kasvattaa lennossa ja palautuspisteiden käytöllä voitiin ratkaista vakavia ongelmatilanteita. Joustavuudesta kertoo myös se, että kurssin aikana syntyneitä kurssilaisten toiveita pystyttiin toteuttamaan nopeassa aikataulussa muun kurssitoiminnan rinnalla. Yksi näistä oli yleinen keskustelukanava, jolla blue team:it pystyisivät jakamaan linkkejä ja tekstinpätkiä. Ratkaisuna päädyttiin unrealIRCD-IRC-palvelimeen ja blue teamien työasemille asennettuihin IRC-asiakassovelluksiin, joita kurssilaiset käyttivät aktiivisesti.

ITKST55-kurssin organisointi

Projektinhallinnallisesta näkökulmasta ITKST55-kurssin vuoden 2019 toteutuksessa oli useita seikkoja, joihin on syytä kiinnittää huomiota mahdollisilla tulevilla kurssitoteutuksilla. Ensisijaisesti olisi syytä kiinnittää huomiota kurssin vaatimusmäärittelyprosessiin. Vykopal ym. (2017) ehdottavat kyberharjoituksen järjestämiseen kahta varsinaista testausvaihetta. Kun kyberharjoituksen skenaario on määritelty ja siihen liittyvät tekniset järjestelmät rakennettu, järjestetään työpaja/hackathon-sessio, jossa syntyneiden havaintojen ja ideoiden perusteella voidaan vielä tehdä muutoksia skenaarioon. Näiden muutosten jälkeen järjestelmä “lyödään lukkoon” ja sen toiminta testataan toteuttamalla pilottiversio harjoituksesta, jossa osallistujina ovat erilliset testaaja-*blue team*:it. Tämänkaltainen tiukkaan määritelty muutosaikataulu ei kuitenkaan välttämättä sovellu sellaisenaan pääasiallisesti opetuskäyttöön ra-

kennettuun kyberharjoitukseen, johon ei sisälly myöskään pelillisiä elementtejä. Opettavassa kyberharjoituksessa saatetaan kaivata enemmän joustavuutta. Tiettyyn aihepiiriin halutaan ehkä käyttää enemmän aikaa tai uusia injektioita saatetaan toivoa lisättäviksi järjestelmään kurssin aikana.

ITKST55:n tapauksessa voisi olla mielekästä määritellä tietyt "leikkauspäivämäärät" järjestelmän kriittisille komponenteille, joiden on pakko onnistua kurssin toteutumisen kannalta. Leikkauspäivämäärien jälkeen voitaisiin järjestää Vykopal ym. (2017) mukainen testaustyöpaja/hackathon uusien ideoiden varalle. Tämä edellyttää luonnollisesti kriittisten komponenttien tarkkarajaista määrittelyä. Esimerkiksi toimiva verkkosegmentointi on virtuaalipalvelinjärjestelmän kannalta kriittinen vaatimus, mutta kurssilaisten väliseen kommunikaatioon tarkoitettu IRC-palvelin on pienemmän prioriteetin vaatimus.

Vuoden 2019 kurssitoteutuksen jälkeen tiedetään tarkemmin, millaisia teknisiä ja järjestäytymisellisiä ongelmia kurssitoteutuksella voidaan havaita. Nämä ongelmatyypit voidaan nyt yleistää ja muuntaa selkeämmiksi toimintaohjeiksi tulevaisuutta varten. Toimintaohjeet mahdollistavat nopean reagoinnin harjoituksen aikana ilmenneisiin ongelmiin ja estävät kurssijärjestelmiin liittyvän tietopohjan siiloutumista tietyille ydinhenkilöille. Ohjeisiin tehtyjä muutoksia seuraamalla voidaan lisäksi tutkia kurssiin liittyvän kollektiivisen tietämyksen kasvua, joka voi olla itsessään kiehtova tutkimusaihe esimerkiksi laajojen kurssikokonaisuuksien järjestämistä tutkivalle.

Esimerkkinä tällaisesta ohjeesta voisi olla ohjeistus siihen, miten toimitaan Security Onion -virtuaalikoneen lakatessa toimimasta kesken harjoituksen. Ohjeistus voisi tässä tapauksessa pitää sisällään vianetsintäneuvoja ja ohjeet virtuaalikoneen palauttamiseen palautuspisteestä. Jos ohjeen noudattamisesta aiheutuu mahdollisia sivuvaikutuksia esimerkiksi uusien teknisten ongelmien muodossa tai niistä muodostuu uhka jollekin/joillekin kurssin tavoitteiden toteutumiselle, täytyy nämä riskit kirjata osaksi ohjetta. Toimintaohjeet tulee jakaa kurssin järjestäjien kesken mahdollisimman helposti saavutettavalla ja myös päivitettävällä tavalla. Tällainen voi olla esimerkiksi wikialusta tai kollaboratiivinen dokumentinmuokkausympäristö.

Proomu-virtuaalipalvelinjärjestelmässä tehdyt tekniset valinnat

Tässä tutkimuksessa virtuaalipalvelinympäristön toteuttamiseen käytettiin Proxmox VE-käyttöjärjestelmäjakelua, mutta se ei ole ainut mahdollinen toteutustapa vastaavalle ympäristölle. Proxmox VE:n tarjoamia hallintatyökaluja ja ominaisuuksia voisi olla kiinnostavaa vertailla muihin samankaltaisiin KVM-pohjaisiin virtualisointiohjelmistojakeluihin. Tällaisia ovat esimerkiksi oVirt, OpenNode ja Ganeti (Linux-KVM.org 2021). Vertailuun voisi olla mahdollista ottaa mukaan myös muita virtualisointialustoja: kyberturvallisuuslaboratorioita on toteutettu esimerkiksi VMWare vSphere- ja VirtualBox-alustoilla (Tunc ja Hariri 2015 ja Schreuders ym. 2017). Yamin, Katt ja Gkioulos (2020) huomauttavat, että erityisesti kyberlaboratorioiden vertailemiseen tarkoitettuja suorituskykytestejä (*benchmark*) ei ole juuri olemassa, ja nostavat niiden kehittämisen merkittäväksi tulevaisuuden tutkimussuuntauksek-si.

Virtuaalipalvelinjärjestelmän sisälle rakennettujen virtuaalikoneiden yhtenevä hallinta voi olla hyvin vaikeaa. Alaluvussa 7.2 esiteltiin Proomu-järjestelmän ongelmia virtuaalisten työasemien tiedonsiirrossa. Vykopal ym. (2017) suosittelevat Ansiblen kaltaisen automaatiotyökalun käyttöä virtuaaliympäristönkonfiguraatioiden hallintaan, sillä pelkissä staattisissa ohjedokumenteissa voi olla virheitä ja ne vanhenevat nopeasti. Bica, Unc ja Țurcanu (2020) rakensivat virtuaalipalvelinympäristön, jonka virtuaalikoneiden ja verkon hallinta toteutettiin Ansible-työkalulla ja Python-skripteillä. Muita virtuaalipalvelinympäristön hallintaan ja ylläpitoon soveltuvia ohjelmistoja ovat esimerkiksi Puppet, Chef ja Saltstack (Bica, Unc ja Țurcanu 2020). Pham ym. (2016) kuitenkin huomauttavat, että monet yleiskäyttöiset automaatiotyökalut saattavat olla vaikeita käyttää kyberharjoituksen vaatimaa monimutkaista ja useita riippuvuussuhteita sisältävää verkkotopologiaa suunniteltaessa. Näin ollen konfiguraatiotyökalun käyttöönottoon ja valintaan on syytä varata aikaa osaksi virtuaalipalvelinjärjestelmän rakennusprosessia.

Tutkimuksen virtuaaliverkon toteutuksessa kiinnostavaa on myös Daemonlogger-ohjelman käyttö verkkoliikenteen peilaamiseen virtuaalisesta Linux-sillasta toiseen. Tutkitusta kirjallisuudesta ei löytynyt viitteitä Daemonloggerin käyttöön kyberturvallisuusharjoitusten yhteydessä. Ohjelmisto havaittiin kuitenkin toimivaksi ja erittäin vakaaksi Proomu-järjestelmässä vuoden 2019 ITKST55-kurssilla, ja jatkotutkimus sen käyttömahdollisuuksista voisi olla

mielekästä. Lisäksi Daemonloggerin suorituskykyä voisi vertailla alaluvussa 6.2.4 esiteltyihin kahteen muuhun peilaustapaan, fyysisellä verkkokytkimellä tehtävään peilaukseen ja Linux *tc*-pohjaiseen liikenteenohjaukseen. Vertailuun tulisi ottaa myös Open vSwitch-pohjainen verkkototeutus, jossa verkkoliikenteen peilaus voidaan toteuttaa ohjelmallisella SPAN-portilla (Vext.info 2018).

Kaupallisten ohjelmistojen käyttö osana kyberharjoitusta on usein varsin monimutkaista: toisaalta harjoitukseen halutaan mukaan organisaatioissa yleisesti käytettyjä kaupallisia työkaluja, mutta niiden käyttöönotto voi vaatia erityisosaamista ja joissakin tapauksissa tietynkaltaista laitteistoa. Brilingaitė, Bukauskas ja Kutka (2017) havaitsivat teknisestä kyberharjoituksesta keräämässään palautteessa, että 25 prosenttia harjoituksen osallistujista eivät olleet halukkaita omaksumaan harjoitukseen kuuluvia työkaluja ja järjestelmiä, koska ne erosivat liikaa kotiorganisaatioissa käytetyistä järjestelmistä. Lisäksi monien kaupallisten käyttöjärjestelmien ja ohjelmistojen käyttö voi olla vaikeaa lisensoinnin kannalta: jokaiselle opiskelijalle tai opiskelijaryhmälle ei välttämättä voida hankkia omaa lisenssiä budjettisyistä. Joissakin tapauksissa voi olla mahdollista järjestää osa harjoituksesta käytön mukaan laskutettavassa pilvipalvelussa, jossa lisenssin käyttö sisältyy hintaan (Topham ym. 2016). Pilvipalveluntarjoajien käyttöehdot voivat kuitenkin rajoittaa pilviympäristössä toteutettavaa kyberharjoitustoimintaa, ja monimutkaisempien verkkoratkaisujen toteuttaminen voi olla vaikeaa.

ITKST55-kurssin vuoden 2019 toteutuksessa ei ollut mukana pelillisiä elementtejä, kuten pisteytystä tai kilpailua joukkueiden kesken, eikä sellaisten puute ollut kurssin oppimistavoitteiden puitteissa oleellista. Pelillistämisellä voitaisiin mahdollisesti parantaa harjoituksen immersiivisyyttä ja opetusteknisestä näkökulmasta helpottaa kurssin arviointia. Toisaalta pelillistäminen kasvattaa harjoituksen oppimiskäyrää ja saattaa lisätä suoriutumispaineita erityisesti yksittäisten opiskelijoiden kohdalla (Brilingaitė, Bukauskas ja Kutka 2017). Harjoituksessa toteutettava pisteytys voitaisiin toteuttaa automaattisesti kurssijärjestelmää monitoroimalla, manuaalisesti osallistujien toimintaa tarkkailemalla tai näiden yhdistelmällä. Kim, Maeng ja Jang (2019) kehottavat kuitenkin huolellisuuteen automaattisia arviointijärjestelmiä suunniteltaessa ja toteutettaessa: puutteellisesti testattu pisteytysjärjestelmä aiheuttaa merkittävästi lisätyötä harjoituksen *white team*:lle itse pelivaiheen aikana. Pelillistämisen teknisen toteutuksen pohtiminen Proomun kaltaisessa virtuaalipalvelinjärjestelmässä voisi

olla ITKST55:n lisäksi kiinnostavaa erityisesti CTF-harjoitusten tai CTF-elementtejä sisältävien yliopistokurssien toteutusta suunniteltaessa.

Hyperkonvergoitu infrastruktuuri Proomu-järjestelmässä

ITKST55-kurssin vuoden 2019 toteutus rakennettiin Jyväskylän yliopiston monitoimitilaan, josta se purettiin pois kurssin päätyttyä. Teknisen kurssiympäristön siirrettävyys oli siis yksi tärkeimpiä kriteerejä virtuaalipalvelinympäristön valinnassa. Tästä yksinkertaisesta oivalluksesta syntyi ajatus kyberharjoituksessa käytettävän palvelinympäristön toteuttamisesta hyperkonvergoidun infrastruktuurin periaatteita hyödyntäen.

Proomu-virtuaalipalvelinjärjestelmän voidaan laskea toteuttaneen osittain hyperkonvergoidun infrastruktuurin periaatteita Haag (2016) esittämällä tavalla: laskennalliset resurssit, tallennustila ja verkkolaitteet toteutettiin pääosin yhden palvelinlaitteiston sisällä, ja niitä pystyttiin konfiguroimaan Proxmox VE:n tarjoamilla hallintatyökaluilla keskitetysti. Kurssitoteutukseen valitut Proxmox VE:n sisäiset toteutusratkaisut eivät kuitenkaan varsinaisesti toteuta hyperkonvergoidun infrastruktuurin edellyttämää skaalautuvuutta. Esimerkiksi verkkoliikenteen peilausta ei voida rakentaa Linux-siltoja käyttävässä verkkokonfiguraatiossa ilman ulkopuolisten sovellusten tai laitteiden käyttöä, kun taas Open vSwitch-verkkokonfiguraatiossa tämä onnistuisi suoraan Vext.info (2018) esittelemällä tavalla (*Proxmox VE Administration Guide* 2019; Vext.info 2018). Open vSwitchin käyttöönotto myös mahdollistaisi ohjelmistopohjaisen verkon periaatteiden käyttöönoton Proomu-järjestelmässä. Ohjelmistopohjainen tallennustila Proxmox VE:ssa puolestaan voitaisiin toteuttaa Ceph-ohjelmistolla. Näiden Proxmox VE:n HCI-ominaisuuksien vertailu ja käyttöönotto voisi olla yksi mahdollinen reitti Proomu-järjestelmän jatkokehitykselle.

Todelliseen mittaansa hyperkonvergoitu infrastruktuuri kuitenkin pääsisi tilanteessa, jossa se otettaisiin osaksi pysyvämmän kyberlaboratorioympäristön suunnittelua. Lesko Jr (2019) esittelivät yliopistokäyttöön suunnitellun kyberturvallisuuden monimuotolaboratorion, jonka tekninen toteutus pohjautuu hyperkonvergoituun infrastruktuuriin. Tällaisen laboratorion on mahdollistettava kyberturvallisuuden opetuksen moninaiset tarpeet, mukaan lukien verkon yli tehtävät tekniset harjoitukset, monimuotokoulutukset ja erilaiset kilpailulliset har-

joitukset. Laboratorioympäristössä tulee pystyä siirtymään nopeasti eri kurssien sisältöjen välillä. Järjestelmän tulee myös tukea useita virtuaalikoneita ja monimutkaisia verkkoratkaisuja edellyttäviä rinnakkaisia opiskelijakohtaisia harjoitusympäristöjä. Monimuotolaboratorion kaikki virtualisoidut laitteet toteutetaan toisessa yliopiston tilassa sijaitsevalla hyperkonvergoidulla laitteella, jota käytetään varsinaiseen laboratoriotilaan sijoitetuilta työasemilta. (Lesko Jr 2019.) Kyberharjoituskäytössä HCI-pohjainen laboratorioarkkitehtuuri voisi soveltua erinomaisesti kyberharjoituskäyttöön tarkoitettujen konfiguraationhallintatyökalujen, kuten Pham ym. 2016 esittelemän CyRIS:n kanssa käytettäväksi.

9 Johtopäätökset ja tuleva tutkimus

ITKST55-kurssi on mahdollista toteuttaa virtualisointiteknologian avulla tutkimuksen aikana kehitetyn prototyypin eli Proomu-virtuaalipalvelinympäristön perusteella. Prototyypin evaluointi osoitti, että järjestelmä täytti suurimman osan sille asetetuista vaatimuksista. Toisaalta evaluoinnissa havaittiin myös suunnittelullisia ongelmia, joihin on syytä perehtyä varhain mahdollisista tulevista kurssitoteutuksista puhuttaessa.

Proomu-ympäristö näytti toteutuksellaan myös, että Proxmox VE voisi soveltua virtualisointialustana kyberturvallisuuden ja tietotekniikan opetuskäyttöön Jyväskylän yliopistossa laajemmaltikin. Proxmox VE:n avulla saatiin toteutettua virtuaalipalvelinjärjestelmä, jolla pystyttiin toteuttamaan ITKST55:n kaltainen kurssi. Jatkokehityksen myötä Proxmoxin ominaisuuksia voitaisiin hyödyntää entistä paremmin ja kurssiympäristön rakentaminen ja ylläpitäminen tehdä helpommaksi. Proomu on tutkimuksen tekijän tietojen mukaan ensimmäinen tutkittu Proxmox VE:tä teknisen kyberharjoituksen järjestämiseen käyttänyt virtuaalipalvelinjärjestelmä. Proomu toimii myös erinomaisena esimerkkinä fyysisten ja virtuaalisten verkkolaitteiden ja työasemien yhdistämisestä ja näiden toteutuksen haasteista, jotka käytiin läpi tutkielman luvussa 7.2. Tutkija toivoo toteutuskuvauksellaan luoneensa uutta suunnittelutieteellistä tietämystä, josta voisi mahdollisesti olla hyötyä vastaavia käyttökohteita suunnitteleville ja samoja työkaluja käyttäville henkilöille.

Hyperkonvergoitu infrastruktuuri (HCI) tarjoaa kiehtovia mahdollisuuksia virtualisoidun kyberturvallisuusopetuksen järjestämiseen. Proomu-järjestelmä toteutti HCI:ta osin: laskenta, tallennustila ja verkkolaitteet toteutettiin ohjelmallisesti, mutta aivan kaikkia Proxmox VE:ssa tarjolla olevia HCI-ominaisuuksia ei ehditty ITKST55:n puitteissa ottaa käyttöön. Tutkimuksessa yhdistettiin ensimmäistä kertaa hyperkonvergoidun infrastruktuurin ja kyberturvallisuusharjoituksen käsitteet, ja aihepiiri tarjoaa useita tutkimusaiheita niin teorian kuin käytännön toteutuksienkin kannalta.

Muista mahdollisista jatkotutkimusreiteistä ensisijaisesti voisi olla mielekästä rakentaa Proomu-virtuaalipalvelinjärjestelmästä uusi iteraatio, joka korjaa luvussa 7.2 käsitellyt puutteet ja luo järjestelmälle joustavamman, paremmin dokumentoidun ja helpommin ylläpidet-

tävämmän jatkumon. Lisäksi voisi olla kiinnostavaa vertailla, miten ITKS55:n virtuaalipalvelinympäristön toteutus onnistuisi muilla vastaavilla virtualisointialustoilla - mitkä ympäristön osat olisivat helpompia toteuttaa ja mitkä niistä taas mahdollisesti vaativat erityisesti Proxmoxin ominaisuuksia.

Tämä tutkimus ei ottanut kantaa rakennetun virtuaalipalvelinympäristön suorituskykyyn liittyviin seikkoihin. Proxmox-ympäristöä voisi vertailla muihin vastaaviin ilmaisiin ja kaupallisiin virtualisointiohjelmistoratkaisuihin. Erityisen mielekästä voisi olla vertailla KVM-pohjaisia virtualisointiratkaisuja keskenään. Vuosi 2020 toi myös konkreettisesti ilmi, että etäopetuskäyttöön joustavasti muuntautuvia järjestelmäratkaisuja tullaan tarvitsemaan tulevaisuudessakin. Olisi siis syytä pohtia, miten ITKST55:n kaltainen kurssi tai muu pienimuotoinen kyberturvallisuusharjoitus olisi mahdollista toteuttaa etäyhteyksien välityksellä tai monimuoto-toteutuksena. Tällaisella toteutuksella on omat vaatimuksensa, joiden määrittelyyn voitaisiin soveltaa esimerkiksi tässä tutkielmassa esiteltyä jakoa opetuksellisiin, opetusteknisiin ja teknisiin vaatimuksiin.

Lähteet

Aken, Joan E van. 2004. “Management research based on the paradigm of the design sciences: the quest for field-tested and grounded technological rules”. *Journal of management studies* 41 (2): 219–246.

Babu, Anish, MJ Hareesh, John Paul Martin, Sijo Cherian ja Yedhu Sastri. 2014. “System performance evaluation of para virtualization, container virtualization, and full virtualization using xen, openvz, and xenserver”. Teoksessa *2014 Fourth International Conference on Advances in Computing and Communications,* 247–250. IEEE.

Bica, Ion, Roxana Larisa Unc ja Ștefan Țurcanu. 2020. “Virtualization and Automation for Cybersecurity Training and Experimentation”. Teoksessa *International Conference on Information Technology and Communications Security,* 227–241. Springer.

Brilingaitė, Agnė, Linas Bukauskas ja Eduardas Kutka. 2017. “Development of an Educational Platform for Cyber Defence Training”. Teoksessa *European Conference on Cyber Warfare and Security,* 73–81. Academic Conferences International Limited.

Chen, Lei, Wei Huang, Aina Sui, Deqin Chen ja Chengsheng Sun. 2017. “The online education platform using Proxmox and noVNC technology based on Laravel framework”. Teoksessa *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS),* 487–491. IEEE.

Eder, Michael. 2016. “Hypervisor-vs. container-based virtualization”. *Future Internet (FI) and Innovative Internet Technologies and Mobile Communications (IITM)* 1.

Haag, M. 2016. *Hyper-Converged Infrastructures for DUMMIES.*

Hjelmvik, Erik. 2011. *Sniffing Tutorial Part 1 - Intercepting Network Traffic.* `https://www.netresec.com/index.ashx?page=Blog&month=2011-03&post=Sniffing-Tutorial-part-1---Intercepting-Network-Traffic`. Noudettu 10.4.2021.

Holm, Hannes, ja Teodor Sommestad. 2016. “Sved: Scanning, vulnerabilities, exploits and detection”. Teoksessa *MILCOM 2016-2016 IEEE Military Communications Conference,* 976–981. IEEE.

Iivari, Juhani. 2007. “A Paradigmatic Analysis of Information Systems As a Design Science”. *Scandinavian Journal of Information Systems* 19 (2): 5.

Societal security - Guidelines for exercises. 2013. Standard. Geneva, CH: International Organization for Standardization, syyskuu.

ITKST55 - Kyberhyökkäys ja sen torjunta. Opintoesite. 2020. `https://sisu.jyu.fi/student/courseunit/jy-CU-9382-v2/brochure`. Noudettu 10.4.2021.

Jain, Raj, ja Subharthi Paul. 2013. “Network virtualization and software defined networking for cloud computing: a survey”. *IEEE Communications Magazine* 51 (11): 24–31.

Järvinen, Pertti ja Annikki. 2000. *Tutkimustyön metodeista.* Opinpajan kirja.

Kasanen, Eero, Kari Lukka ja Arto Siitonen. 1993. “The constructive approach in management accounting research”. *Journal of management accounting research* 5 (1): 243–264.

Kim, Joonsoo, Youngjae Maeng ja Moonsu Jang. 2019. “Becoming Invisible Hands of National Live-Fire Attack-Defense Cyber Exercise”. Teoksessa *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW),* 77–84. IEEE.

Lesko Jr, Charles J. 2019. “A design case: Assessing the functional needs for a multi-faceted cybersecurity learning space”. *Journal of Cybersecurity Education, Research and Practice* 2019 (1): 6.

Linux-KVM.org. 2021. *List of KVM management tools.* `https://www.linux-kvm.org/page/Management_Tools`. Noudettu 10.4.2021.

Linuxtopia. 2020. *Linux Packet Filtering and iptables. Chapter 11. Iptables targets and jumps. 11.11 NETMAP target.* `https://www.linuxtopia.org/Linux_Firewall_iptables/x4471.html`. Noudettu 10.4.2021.

Macedo, Ricardo, João Paulo, José Pereira ja Alysson Bessani. 2020. “A Survey and Classification of Software-Defined Storage Systems”. *ACM Computing Surveys (CSUR)* 53 (3): 1–38.

Makrodimitris, Georgios, ja Christos Douligeris. 2015. “Towards a Successful Exercise Implementation–A Case Study of Exercise Methodologies”. Teoksessa *International Conference on Human Aspects of Information Security, Privacy, and Trust,* 207–218. Springer.

Manik, Varun Kumar, ja Deepak Arora. 2016. “Performance comparison of commercial VMM: ESXI, XEN, HYPER-V & KVM”. Teoksessa *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom),* 1771–1775. IEEE.

March, Salvatore T, ja Gerald F Smith. 1995. “Design and natural science research on information technology”. *Decision support systems* 15 (4): 251–266.

Netgate. 2019. *The pfSense Book.* Electric Sheep Fencing LLC.

ONF. 2012. “Software-defined networking: The new norm for networks”. *ONF White Paper.*

Ouzounis, Evangelos. 2009. *Good Practice Guide On National Exercises.* ENISA.

Peffers, Ken, Tuure Tuunanen, Marcus A Rothenberger ja Samir Chatterjee. 2007. “A design science research methodology for information systems research”. *Journal of management information systems* 24 (3): 45–77.

Pham, Cuong, Dat Tang, Ken-ichi Chinen ja Razvan Beuran. 2016. “CyRIS: a cyber range instantiation system for facilitating security training”. Teoksessa *Proceedings of the Seventh Symposium on Information and Communication Technology,* 251–258.

Piirainen, Kalle A, ja Rafael A Gonzalez. 2014. “Constructive synergy in design science research: a comparative analysis of design science research and the constructive research approach”. *Liiketaloudellinen Aikakauskirja* 3 (4): 206–234.

Plauth, Max, Lena Feinbube ja Andreas Polze. 2017. “A performance survey of lightweight virtualization techniques”. Teoksessa *European Conference on Service-Oriented and Cloud Computing,* 34–48. Springer.

Popek, Gerald J, ja Robert P Goldberg. 1974. “Formal requirements for virtualizable third generation architectures”. *Communications of the ACM* 17 (7): 412–421.

Portnoy, Matthew. 2016. *Virtualization Essentials.* John Wiley & Sons.

Prat, Nicolas, Isabelle Comyn-Wattiau ja Jacky Akoka. 2014. “Artifact Evaluation in Information Systems Design-Science Research-a Holistic View.” *PACIS* 23:1–16.

Proxmox 6.2 press release. 2020. `https://www.proxmox.com/en/news/press-releases/proxmox-ve-6-2`.

Proxmox Virtual Environment 6.2 - Datasheet. 2020. `https://www.proxmox.com/en/downloads/item/proxmox-ve-admin-guide-for-6-x`.

Proxmox VE Administration Guide. 2019. Release 6.0. Proxmox Server Solutions Gmbh.

Ragsdale, D. J., S. D. Lathrop ja R. C. Dodge. 2003. “A virtual environment for IA education”. Teoksessa *IEEE Systems, Man and Cybernetics SocietyInformation Assurance Workshop, 2003.* 17–23. doi:`10.1109/SMCSIA.2003.1232395`.

Roesch, Martin. 2017. *Daemonlogger 1.2.1 README.* `https://github.com/Cisco-Talos/Daemonlogger`. Noudettu 10.4.2021.

Russo, Enrico, Gabriele Costa ja Alessandro Armando. 2018. “Scenario design and validation for next generation cyber ranges”. Teoksessa *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA),* 1–4. IEEE.

———. 2020. “Building next generation Cyber Ranges with CRACK”. *Computers & Security* 95:101837.

Schreuders, Z Cliffe, Thomas Shaw, Mohammad Shan-A-Khuda, Gajendra Ravichandran, Jason Keighley ja Mihai Ordean. 2017. “Security Scenario Generator (SecGen): A Framework for Generating Randomly Vulnerable Rich-scenario VMs for Learning Computer Security and Hosting {CTF} Events”. Teoksessa *2017 {USENIX} Workshop on Advances in Security Education ({ASE} 17).*

Seker, Ensar, ja Hasan Huseyin Ozbenli. 2018. “The concept of cyber defence exercises (cdx): Planning, execution, evaluation”. Teoksessa *2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security),* 1–9. IEEE.

Stewart, Kyle E, Jeffrey W Humphries ja Todd R Andel. 2009. “Developing a virtualization platform for courses in networking, systems administration and cyber security education”. Teoksessa *Proceedings of the 2009 Spring Simulation Multiconference,* 65. Society for Computer Simulation International.

Thompson, Michael F, ja Cynthia E Irvine. 2018. “Individualizing cybersecurity lab exercises with labtainers”. *IEEE Security & Privacy* 16 (2): 91–95.

Topham, Luke, Kashif Kifayat, Younis A Younis, Qi Shi ja Bob Askwith. 2016. “Cyber security teaching and learning laboratories: A survey”. *Information & Security* 35 (1): 51.

Traficom. 2019. *Kyberharjoitusohje - käsikirja harjoituksen järjestäjälle.* Nide 26/2019. Liikenne-ja viestintävirasto Traficom.

Tunc, Cihan, ja Salim Hariri. 2015. “CLaaS: Cybersecurity Lab as a Service.” *J. Internet Serv. Inf. Secur.* 5 (4): 41–59.

Vext.info. 2018. *Port mirroring IDS data into a Proxmox VM.* `https://vext.info/2018/09/03/cheat-sheet-port-mirroring-ids-data-into-a-proxmox-vm.html`. Noudettu 10.4.2021.

Von Alan, R Hevner, Salvatore T March, Jinsoo Park ja Sudha Ram. 2004. “Design science in information systems research”. *MIS quarterly* 28 (1): 75–105.

Vykopal, Jan, Martin Vizváry, Radek Oslejsek, Pavel Celeda ja Daniel Tovarnak. 2017. “Lessons learned from complex hands-on defence exercises in a cyber range”. Teoksessa *2017 IEEE Frontiers in Education Conference (FIE),* 1–8. IEEE.

Yamin, Muhammad Mudassar, Basel Katt ja Vasileios Gkioulos. 2020. “Cyber ranges and security testbeds: Scenarios, functions, tools and architecture”. *Computers & Security* 88:101636.