Opinnäytetyö (AMK)

Tietojenkäsittelyn koulutusohjelma

2020

Sami Laine

TIETOTURVAN JA LOKIENHALLINNAN TESTIYMPÄRISTÖN SUUNNITTELU JA TOTEUTTAMINEN

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietojenkäsittelyn koulutusohjelma

2020 | 39 sivua

Sami Laine

TIETOTURVAN JA LOKIENHALLINNAN TESTIYMPÄRISTÖN SUUNNITTELU JA TOTEUTTAMINEN

Penetraatiotestaus ja lokienhallinta ovat merkittävä osa modernia tietoturvaa. Penetraatiotestaamista ja lokienhallinnan muutosten testaamista varten tarvitaan kuitenkin suljettu testausympäristö, jonka käytöstä ei aiheudu haittaa yrityksen tuotantoympäristöön. Tällaisessa ympäristössä voidaan muun muassa testata uusia konfiguraatiomuutoksia ennen niiden viemistä tuotantokäyttöön, tai voidaan yrittää toistaa tuotantoympäristössä tapahtuvia ongelmia ja etsiä niihin ratkaisuja. Tämän lisäksi ympäristöä voidaan käyttää oppimistarkoituksiin tai kokonaan uusien sovellusten testaamiseen.

Tämän opinnäytetyön tavoitteena on suunnitella ja luoda toimeksiantajalle suljettu tietoturvan ja lokienhallinnan testiympäristö sekä penetraatiotestaamiseen että lokienhallinnan testaamiseen. Tavoitteena oli myös selvittää virtualisoinnin tuomia hyötyjä testiympäristöjen luomisessa sekä penetraatiotestauksen ja lokienhallinnan merkitystä tietoturvan kannalta.

Virtualisoinnin tuomat hyödyt tällaisen ympäristön luomisessa ovat kiistattomat. Laitteiden konfigurointi on nopeampaa kuin fyysisillä laitteilla, resurssien käyttö tehokasta ja vaadittu fyysinen tila vähäistä. Lisäksi virtualisointi tarjoaa ominaisuuksia, jotka eivät ole mahdollisia fyysisellä laitteistolla, kuten palautuspisteiden luomisen sekä laitteiden kloonaamisen.

Suljettu testiympäristö toteutettiin käyttämällä VMWaren virtualisointiohjelmistoja, jotka olivat jo valmiiksi asennettuna yrityksen virtualisointipalvelimelle. Ympäristössä käytettiin pääasiassa avoimen lähdekoodin sovelluksia. Penetraatiotestaukseen käytettiin Linux-pohjaista Kali-käyttöjärjestelmää, sekä Nmap-, OpenVAS- ja Nessus-työkaluja. Lokienhallinnassa päädyttiin käyttämään NXLogia, Elastic Stackia sekä Graylogia. Tämän lisäksi ympäristöön luotiin domain controller -palvelin ympäristön hallinnoimiseen.

ASIASANAT:

tietoturva, lokienhallinta, virtualisointi, penetraatiotestaus

BACHELOR´S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Business Information Technology

2020 | 39 pages

Sami Laine

DESIGN AND CREATION OF VIRTUAL ENVIRONMENT FOR INFORMATION SECURITY TESTING AND LOG MANAGEMENT

Penetration testing and log management are a significant part of modern information security. Conducting penetration testing and testing changes to log management, however, requires a closed environment where it is not possible to cause problems to the production environment. In an environment like this new configuration changes can be tested before making the changes in the production environment. Attempts can also be made to replicate problems happening in the production environment to try to find solutions. In addition, the test environment can be used for learning purposes, and for testing completely new applications.

The objective of this thesis was to design and create a closed environment for testing information security and log management for the client. Another objective was to find out the pros of using virtualization to create a test environment, and to discuss the significance of penetration testing and log management in information security.

The pros of virtualization in creating an environment like this are undeniable. Device configuration is faster, the use of resources is more efficient than on physical devices, and the physical space taken by the equipment is minor. Virtualization also provides features which are not possible on physical devices, such as creating snapshots and cloning devices.

The closed test environment was implemented using virtualization software by VMWare, which was already installed on the company’s virtualization server. Mainly open source applications were used in the environment. The Linux-based operating system Kali was used for penetration testing, as well as the penetration testing tools Nmap, OpenVAS and Nessus. For log management, NXLog, Elastic Stack and Graylog were chosen. In addition, a domain controller was implemented in the environment.

KEYWORDS:

information security, log management, virtualization, penetration testing

SISÄLTÖ

KUVAT

SANASTO

AD	Microsoftin kehittämä hakemistopalvelu toimialueen käyttäjien ja laitteiden hallinnointiin, Active Directory
DHCP	verkkoprotokolla IP-osoitteiden jakamiseen, Dynamic Host Configuration Protocol
DNS	nimipalvelujärjestelmä IP-osoitteiden nimiksi muuntamiseen, Domain Name System
Domain	toimialue, viittaa joukkoon Windows-koneita, joiden hallinnointi on mahdollista keskitetysti Windows-palvelimen kautta
Domain Controller	palvelin, joka vastaa toimialueen käyttäjien kirjautumisten varmentamisesta
JSON	tekstipohjainen tiedostomuoto datan lähettämiseen ja tallentamiseen, JavaScript Object Notation
SIEM	sovellus tai joukko sovelluksia, jotka keräävät lokidataa useilta ympäristön laitteilta tietoturvapoikkeamien havaitsemiseksi, Security Information and Event Management Software
SQL	IBM:n kehittämä relaatiotietokantojen kyselykieli, Structured Query Language
TCP	tietoliikenneprotokolla, jossa tietokoneiden välille luodaan yhteys ennen varsinaista tiedonsiirtoa, Transmission Control Protocol
UDP	tietoliikenneprotokolla, jossa tietokoneiden välille ei luoda yhteyttä ennen tiedonsiirtoa, User Datagram Protocol

1 JOHDANTO

Palveluiden ja tiedon jatkuvan digitalisoitumisen seurauksena on tietoturvan korkea taso nykyään tärkeää niin pienille kuin suurillekin yrityksille. Erityisen tärkeää se on kuitenkin arkaluontoisia tietoja käsitteleville sosiaali- ja terveysalan toimijoille, joiden on varmistettava potilastietojen salassa pysyminen. Jotta tietoturvan korkea taso voidaan varmistaa, on käytössä olevia työkaluja sekä käytäntöjä jatkuvasti päivitettävä ja testattava. Testaamista ei kuitenkaan usein voida tehdä tuotantokäytössä olevassa verkossa aiheuttamatta käyttökatkoja tai muuta haittaa loppukäyttäjille. Erityisesti sosiaali- ja terveysalalla näistä käyttökatkoista voi olla merkittäviä seurauksia esimerkiksi potilasturvallisuuden kannalta. Tämän vuoksi tarvitaan suljettu ympäristö, jossa testaaminen voidaan suorittaa turvallisesti, ilman käyttäjille aiheutuvaa haittaa.

Tämän opinnäytetyön tavoitteena on tutkia tietoturvatestauksen, lokituksen sekä virtuaalisten ympäristöjen keskeisiä konsepteja ja ominaisuuksia, ja niiden perusteella suunnitella ja rakentaa toimeksiantajalle virtuaalinen tietoturvan ja lokituksen testiympäristö. Ympäristön tarkoituksena on mahdollistaa erilaisten sovellusten ja lokityökalujen testaaminen suljetussa ympäristössä hallitusti, ennen kuin muutokset otetaan käyttöön tuotantoympäristössä. Tämän lisäksi ympäristössä voidaan suorittaa penetraatiotestausta. Opinnäytetyön toimeksiantajana toimii suomalainen julkisomisteinen sosiaali- ja terveysalan ICT-yhtiö.

Opinnäytetyön teoriaosuudessa käsitellään virtualisoinnin tuomia hyötyjä erityisesti testiympäristöjen rakentamisessa. Tämän lisäksi esitellään testiympäristöjen ominaisuuksia sekä syitä penetraatiotestaamiseen ja lokitukseen. Teoriaosuudessa esitellään myös tärkeimmät tietoturvatestaamis- ja lokitustyökalut, joita ympäristössä käytetään, sekä miksi juuri nämä työkalut ympäristöön valittiin. Opinnäytetyön soveltavassa osuudessa kuvataan ympäristön fyysisten ja virtuaalisten laitteiden suunnittelu ja käyttötarkoitukset sekä verkon rakenne. Lopuksi tarkastellaan ympäristön toteuttamista suunnitelman pohjalta ja testataan ympäristön toimintaa.

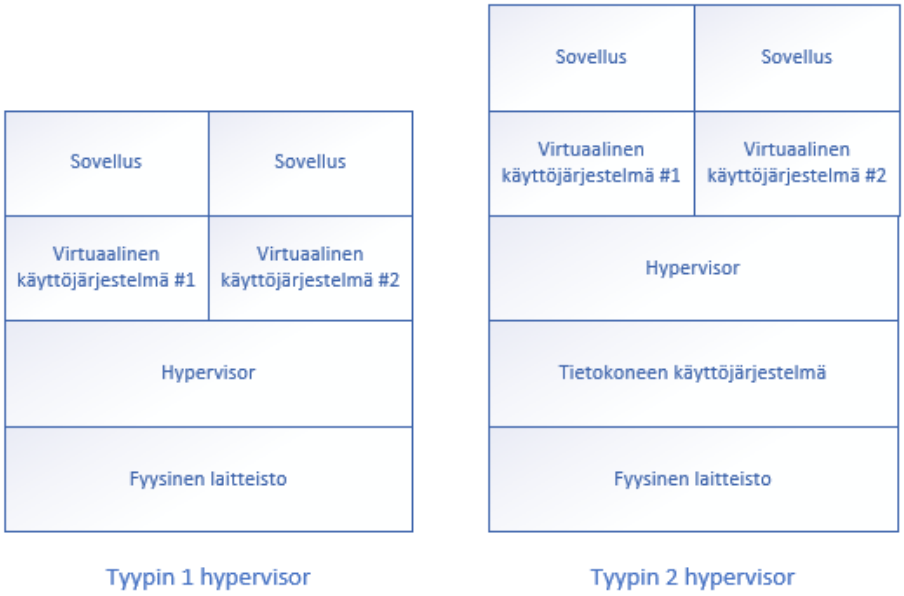
2 VIRTUALISOINTI JA TESTIYMPÄRISTÖT

2.1 Virtualisointi

Perinteisesti yhdelle fyysiselle palvelimelle on asennettu yksi käyttöjärjestelmä, joka isännöi yhtä sovellusta. Tällainen sovellusten isännöinti ei kuitenkaan ole tehokasta, sillä sovelluksen viedessä vain murto-osan laitteiston resursseista, jää laitteiston hyötykäyttö huonoksi. Tämä huono hyötykäyttö on kuitenkin ollut aikaisemmin välttämätön paha, jotta on voitu varmistaa sovellusten tietoturva ja ottaa huomioon mahdolliset suorituskykyyn liittyvät rajoitteet. Ratkaisu tähän ongelmaan on virtualisointi. Virtualisoimalla voidaan yhdelle fyysiselle laitteelle asentaa monta käyttöjärjestelmää, jotka isännöivät kukin vain yhden sovelluksen, kuten perinteisillä palvelimillakin. Koska yhdellä palvelimella isännöidään useampia sovelluksia, on resurssien hyötykäyttö parempaa, ja vaadittu fyysinen tila sekä jäähdytys- ja sähkökustannukset pienempiä. (Fitzhugh 2014, 28.)

Virtualisoinnissa luodaan virtuaalisia tietokoneita, eli virtuaalikoneita, joiden ominaisuudet määritellään erilaisilla tiedostoilla. Nämä tiedostot muodostavat virtuaalikoneen virtuaaliset komponentit, jotka toimivat vastaavasti kuin perinteisen palvelimen fyysiset komponentit. Erona on kuitenkin se, että virtuaalisten komponenttien ominaisuuksien muuttaminen on mahdollista muuttamalla virtuaalikoneiden asetuksia, eikä vaihtamalla palvelimen fyysisiä komponentteja. Virtuaalisten komponenttien ominaisuuksia kuitenkin rajoittavat virtualisointipalvelimelle asennettujen komponenttien ominaisuudet. Tämä tarkoittaa sitä, ettei virtuaalikoneella voi esimerkiksi olla enempää virtuaaliprosessoreja kuin virtualisointipalvelimessa on fyysisiä prosessoreja. (Fitzhugh 2014, 23.)

Virtuaalikoneet luodaan käyttämällä hypervisoreja, jotka voidaan jakaa tyyppeihin 1 ja 2. Tyypin 1 hypervisorit muistuttavat tietokoneen normaalia käyttöjärjestelmää siinä mielessä, että ne asennetaan suoraan fyysiselle laitteistolle. Tämän tyyppiset hypervisorit kommunikoivat siis suoraan laitteiston kanssa ilman välikäsiä. Tyypin 2 hypervisorit sen sijaan asennetaan laittestoon jo valmiiksi asennettuun käyttöjärjestelmään. Ne ovat siis kuin sovelluksia, jotka kommunikoivat fyysisen laitteiston kanssa toisen käyttöjärjestelmän kautta. Virtuaalikoneet eivät tiedä hypervisorin eivätkä sen seurauksena muidenkaan virtuaalikoneiden olemassaolosta, mikä ratkaisee aikaisemmin mainitun ongelman sovellusten tietoturvassa. (Gregg 2015, 11.) Kuvassa 1 havainnollistetaan hypervisortyyppien rakenteet.



Kuva 1. Hypervisor-tyypit.

Virtualisointi tarjoaa tehokkaamman fyysisen laitteiston hyväksikäytön lisäksi myös monia muita etuja, joita perinteisillä fyysisillä palvelimilla ei voida toteuttaa. Hypervisorit, kuten VMwaren ESXi, mahdollistavat palautuspisteiden luomisen, virtuaalikoneiden siirtämisen toiselle fyysiselle palvelimelle sekä käyttökatkoista nopean palautumisen. Käyttämällä valmiita pohjia ja kloonausta virtuaalikoneiden luomiseen, voidaan myös säästää tunteja verrattuna perinteisen palvelimen konfigurointiin. (Fitzhugh 2014, 28.)

Vaikka virtualisointi saattaa kuulostaa täydelliseltä ratkaisulta moneen ongelmaan, on virtualisoinnillakin huonot puolensa. Jotta virtualisointi onnistuu ongelmitta, tulee varmistaa kaikkien komponenttien yhteensopivuus virtualisointiohjelmiston kanssa, jonka johdosta vanhaa laitteistoa ei välttämättä voida käyttää uudelleen virtualisointiin siirtyessä. Tämän lisäksi ohjelmistokustannukset voivat olla korkeammat kuin fyysistä laitteistoa käyttäessä, sillä virtuaalikoneiden sovelluslisenssien lisäksi täytyy maksaa myös virtualisointisovelluksesta, joka voi laajoissa ympäristöissä tulla hyvinkin kalliiksi. (Fitzhugh 2014, 28.)

2.2 Testiympäristöt

Uuden sovellusmuutoksen tai kokonaan uuden sovelluksen käyttöön ottaminen suuressa yrityksessä ei ole niin yksinkertaista, että muutos vain otetaan käyttöön sen tullessa saataville. Ennen muutoksen käyttöönottoa täytyy varmistaa, että tämä muutos on yhteensopiva tuotantoympäristön muiden osien kanssa, eikä aiheuta haittoja itse muutettavan sovelluksen tai muiden sovellusten toimintaan. Tämä voidaan varmistaa testaamalla muutoksen toimintaa suljetussa testiympäristössä, jonka toiminnalla ei aiheuteta katkoja tuotantoverkon toimintaan. Jotta testien lopputuloksia voidaan pitää luotettavina, tulee ympäristön olla suunniteltu testaamiseen, eli ympäristön tulee olla ominaisuuksiltaan tuotantoympäristöä vastaava. (Gregg 2015, 2.)

Sovellusmuutosten ja uusien ominaisuuksien testaamisen lisäksi voidaan testiympäristössä testata monia muitakin asioita. Yksi tärkeä käyttökohde on tuotantoympäristön ongelmien simuloiminen testiympäristössä. Jos tuotantoympäristössä oleva ongelma saadaan toistettua testiympäristössä esimerkiksi kloonaamalla ongelmia aiheuttava laite testiympäristöön, on ongelmanratkaisu helpompaa. Testiympäristössä voidaan vaihtaa asetuksia ja ominaisuuksia aiheuttamatta käyttökatkoja oikeaan ympäristöön, ja ratkaisun löytyessä voidaan se ottaa kerralla käyttöön tuotantoympäristössä. (Kusek ym. 2014, 48–50.)

Testiympäristöä suunnitellessa on tärkeää valita oikeat työkalut ympäristön rakentamiseen, alkaen hypervisor-tyypin valinnasta. Tyypin 1 hypervisor tarjoaa enemmän ominaisuuksia ja mahdollistaa muun muassa monimutkaisempien ympäristöjen kehittämisen, sekä testiympäristöön pääsyn etäyhteydellä verkon ulkopuolelta. Tyypin 1 hypervisorit kuitenkin vaativat fyysisen laitteen pyhittämisen tähän käyttötarkoitukseen, eli tyypin 1 hypervisorit sopivat pääasiassa palvelimille rakennettaviin ympäristöihin. Jos ympäristö halutaan isännöidä tietokoneelta, jota käytetään muihinkin käyttötarkoituksiin, on tyypin 2 hypervisor oikea ratkaisu. Tämä vaihtoehto ei tarjoa yhtä monia ominaisuuksia kuin tyypin 1 hypervisorit, mutta saattaa yksinkertaisten ympäristöjen rakentamisessa olla järkevämpi ratkaisu, koska ympäristölle ei tarvita kokonaan omaa palvelinta. (Cardwell 2014, 25–26.) Hypervisorin valinnan lisäksi tärkeää ympäristön suunnittelussa on seurata mahdollisimman tarkasti tuotantokäytössä olevien sovellusten ja laitteiden ominaisuuksia, jotta saadut testitulokset vastaavat tuotantoympäristössä nähtävää lopputulosta (Kusek ym. 2014, 55).

3 TIETOTURVA JA LOKIT

3.1 Tietoturva

Tietoturva voidaan määritellä monella eri tavalla, mutta useimmiten määritelmään kuuluvat seuraavat kolme tietoturvan peruspilaria: luottamuksellisuus, eheys ja käytettävyys. Nämä kolme tietoturvan osa-aluetta tunnetaan ehkä parhaiten nimellä CIA-kolmio, jonka nimitys tulee englannin kielen sanoista Confidentiality, Integrity ja Availability. Confidentiality (suomeksi luottamuksellisuus) tarkoittaa, että vain henkilöt, joilla on oikeus tiettyyn informaatioon tai dataan, pääsevät siihen käsiksi. Luottamuksellisuuden varmistamiseksi vaaditaan niin fyysisiä toimia, kuten ulkopuolisten pääsyn sääntelyä tiloihin, joissa dataa säilötään, kuin myös teknisiä toimia, kuten käyttövaltuushallintaa ja erilaisia datan käyttöön liittyviä sääntöjä ja ohjeistuksia. Kolmion toinen osa integrity eli eheys tarkoittaa datan säilymistä alkuperäisessä muodossaan, ellei sitä muokkaa taho, jolla on oikeus datan muokkaamiseen. Riippumatta siitä, onko data säilöttynä esimerkiksi palvelimelle, tai sen siirtyessä paikasta toiseen, tulee datan pysyä alkuperäisessä muodossaan. Viimeisenä kolmion osana on availability, eli käytettävyys, joka tarkoittaa, että data on saatavilla niille tahoille, joilla on oikeus datan käyttöön. Tätä tietoturvan kohtaa vastaan hyökkääminen on yksi suosituimmista hyökkäysvektoreista, joka usein toteutetaan palvelunestohyökkäyksillä. Saatavuuden varmistamiseksi tulee datasta olla varmuuskopiot, jonka lisäksi tarvitaan suunnitelma datan palauttamiseen varmuuskopioista takaisin käyttöön. (Cabric 2015, 185–186.)

Tietoturvan huomiotta jättämisestä seuraa huomattavia riskejä niin yksilölle kuin yritykselleki, kuten mahdollisuus tietotovuotoon, taloudellisiin tappioihin, identiteettivarkauteen tai yksityisyyden suojan menetykseen. Riskit eivät kuitenkaan lopu hyökkäyksen uhriin kohdistuviin riskeihin. Jos suojaamattoman laitteen liittää internetiin, voidaan laitettaa väärinkäyttää muutenkin kuin varastamalla tietoja laitteelta. Rouskun mukaan mikä tahansa internetiin liitetty laite voi toimia osana tietoverkkorikollisten verkkoa, jolla suoritetaan erilaisia muihin kohdistuvia tietoturvahyökkäyksiä. Omien etujen ajattelun lisäksi organisaatioiden on Suomen ja EU:n lakien ja asetusten mukaisesti pakko huolehtia tietoturvasta. Organisaatioita velvoittaa tietoturvan ylläpitämiseen myös erilaiset sopimukset muiden organisaatioiden, kuten alihankkijoiden kanssa. Näiden sopimusten rikkominen voi johtaa muun muassa erilaisiin sanktiomaksuihin tai organisaatioiden välisten sopimusten purkamiseen. (Rousku 2014, 73–75.)

3.2 Penetraatiotestaus

Yksi keino tietoturvan ylläpitämiseen on penetraatiotestaus. Georgia Weidman (2014, 1) määrittelee penetraatiotestauksen, tai lyhyemmin pentestauksen, prosessiksi, jossa yritetään löytää mahdollisia tietoturva-aukkoja, ja selvittää niiden aiheuttamia riskejä simuloimalla oikeita tietoturvahyökkäyksiä. Toisin kuin normaalissa haavoittuvaisuuksien kartoituksessa, ei penetraatiotestauksessa vain arvioida riskejä, vaan hyökkäys yritetään myös toteuttaa ja näin todentaa mahdolliset riskit. Weidmanin mukaan suuretkin yritykset joutuvat usein muun muassa yksinkertaisten SQL-haavoittuvaisuuksien, sosiaalisen hakkeroinnin sekä heikkojen salasanojen aiheuttamien riskien uhreiksi, eikä syynä tietovuotoon useimmiten olekaan viimeisin nollapäivähaavoittuvaisuus. Nämä yksinkertaiset haavoittuvaisuudet ovat kaikki sellaisia, jotka voidaan löytää penetraatiotestaamisen avulla, ja näin korjata ennen kuin hyökkääjät pääsevät niitä hyödyntämään. (Weidman 2014, 1.)

Penetraatiotestaajan tehtävänä on tutkia, kartoittaa ja testata kohdeyrityksen tietoturvaa käyttämällä vastaavia työkaluja kuin oikea hyökkääjä käyttäisi. Penetraatiotestaajan erottaa hakkerista hyökkäyksen tarkoituksen lisäksi suhde hyökkäyksen kohteeseen. Penetraatiotestaaja on joko yrityksen oma työntekijä tai kyseiseen tehtävään palkattu ulkoisen toimittajan työntekijä, jolla on lupa testaamisen tekemiseen, kun taas hakkeri tekee tietoturvahyökkäyksiä ilman kohteen lupaa. Muita penetraatiotestaajista käytettyjä nimiä ovat eettinen hakkeri ja valkohattuhakkeri. (Oriyano 2016, 2.)

3.3 Lokit ja lokienhallinta

Kyberturvallisuuskeskuksen mukaan loki tarkoittaa ”aikajärjestyksessä kirjattua tallennetta tapahtumista ja niiden aiheuttajista” (Kyberturvallisuuskeskus 2020). Lokeihin voidaan kirjata muun muassa tietokoneen erilaisia tapahtumia, kuten kirjautumisia ja erilaisia ohjelmistomuutoksia tai palomuurin läpi käyvää liikennettä. Tällaisten tietojen tallentamista ja hyödyntämistä kutsutaan lokitukseksi. (Kyberturvallisuuskeskus 2020.)

Jotta loki on hyödyllinen, täytyy siinä olla tarpeeksi tietoja tapahtumasta, joka lokiin on kirjattu. Kyberturvallisuuskeskuksen mukaan riittävän hyvään lokiin tarvitaan vähintäänkin seuraavat tiedot: aikaleima, josta selviää milloin tapahtuma on tapahtunut, itse tapahtuma ja tapahtuman tekijä, käyttöoikeus jolla tapahtuma tehtiin, tapahtuman lähde,

eli mistä tapahtuma tehtiin ja tapahtuman tila, eli onnistuiko tekijä aikeessaan. Lokien tallentamisessa tulee olla tarkkana, sillä lokit saattavat sisältää henkilötietoja, jolloin lokien käsittelyssä tulee ottaa huomioon EU:n tietosuoja-asetus. Pääsääntöisesti tuleekin välttää ylimääräisten henkilötietojen keräämistä lokeihin. (Kyberturvallisuuskeskus 2020.)

Lokeja käytetään moniin erilaisiin käyttötarkoituksiin, kuten ongelmien ratkomiseen, laitteiden ja verkkojen suorituskyvyn optimointiin, sekä käyttäjien toimien, kuten kirjautumisten seuraamiseen. Erilaisten lokeja keräävien laitteiden määrän kasvaessa on myös lokien määrä kasvanut. Tämän lisäksi lokeja tulee useammassa eri muodossa, jonka johdosta tarvitaan oma prosessi lokien hallinnalle. Tähän prosessiin kuuluu lokien luominen, siirtäminen, tallentaminen, analysointi ja lopulta poistaminen. (NIST 2006, 2-1.) Tällaisella lokienhallintaprosessilla yritys voi varmistaa, että lokeja säilötään tarvittava aika, muttei tätä pidempään. Tämän lisäksi prosessiin kuuluvalla säännöllisellä lokien analysoinnilla voidaan löytää tietoturvapoikkeamia, ja informaatiota kuinka estää tällaiset ongelmat tulevaisuudessa. (NIST 2006, 2-7.)

Lokienhallintaprosessia helpottaakseen voidaan käyttää muun muassa syslog-pohjaista keskitettyä lokienhallintajärjestelmää, jos kaikki lokeja keräävät laitteet tukevat syslog-formaattia. Tällaisessa järjestelmässä kaikki ympäristön laitteet lähettävät lokit samassa formaatissa syslog-palvelimelle, jossa lokien analysointi voidaan suorittaa. (NIST 2006, 3-5.) Lokien keskitetty kerääminen voidaan totetuttaa myös SIEM-järjestelmällä, joka mahdollistaa erilaisissa formaateissa olevien lokien normalisoinnin haluttuun muotoon. Tämä normalisointi voidaan toteuttaa joko palvelimella tai lokeja luovalla laitteella käyttäen SIEM-järjestelmän agenttisovellusta. (NIST 2006, 3-9.)

4 TESTIYMPÄRISTÖN TESTAUS- JA LOKIENHALLINTATYÖKALUT

4.1 Kali Linux

Kali Linux on penetraatiotestaukseen ja tietoturva-auditointiin tarkoitettu Debianiin pohjautuva Linux-jakelu. Kali on täysin ilmainen, ja sen lähdekoodi on avoin kaikille nähtäväksi tai omaan käyttöön muokattavaksi. Se julkaistiin maaliskuussa 2013 BackTrack Linuxin täysin uudistettuna versiona. Kalia kehittää, rahoittaa ja ylläpitää Offensive Security. (Kali 2020d.)

Kali on tarkoitettu penetraatiotestaajille ja tietoturva-ammattilaisille, eikä sitä suositella muuhun käyttöön. Vaikka Kali on laajasti muokattavissa, ei ulkoisten sovellusten asentaminen aina onnistu helposti, tai joissain tapauksissa ollenkaan. Kaliin onkin tietoturvasyistä lisätty vain minimaalinen määrä tietolähteitä, jotka ovat varmasti luotettavia. Kalin käyttäjien tulee hallita Linuxin käyttö yleisesti ja ymmärtää mahdolliset seuraamukset, joita Kalin työkalujen väärinkäytöstä seuraa. (Kali 2020c.)

Kaliin sisältyy valmiiksi yli 600 työkalua (Kali 2020d), jotka voidaan jakaa 13 kategoriaan:

- Information Gathering
- Vulnerability Analysis
- Wireless Attacks
- Web Applications
- Exploitation Tools
- Stress Testing
- Forensics Tools
- Sniffing & Spoofing
- Password Attacks
- Maintaining Access
- Reverse Engineering
- Reporting Tools
- Hardware Hacking.

Osalla Kalin työkaluista on useampi käyttötarkoitus, ja näin ollen ne myös kuuluvat useampaan kategoriaan. (Kali 2020a.)

4.2 OpenVAS

OpenVAS on avoimen lähdekoodin haavoittuvaisuusskanneri, jota Greenbone Networks on kehittänyt vuodesta 2009 asti. Sovellusta on kehitetty vuodesta 2005 lähtien eri nimillä haavoittuvaisuusskanneri Nessuksen lopetettua toimintansa avoimen lähdekoodin alla. Sen ominaisuuksiin kuuluvat todennettu ja todentamaton testaaminen, useat korkean ja matalan tason internet- ja teollisuusprotokollat sekä tehokas sisäinen ohjelmointikieli. Skanneriin kuuluu päivittäin päivittyvä Greenbone Community Feed -syöte, joka sisältää yli 50 000 haavoittuvaisuustestiä. (Greenbone n.d.a.)

Yhdistettynä muiden avoimen lähdekoodin moduulien kanssa muodostaa OpenVAS Greenbone Vulnerability Management -ratkaisun (GVM), joka on osa Greenbonen suurempaa kaupallista ohjelmistoperhettä nimeltään Greenbone Security Manager tai lyhyemmin GSM. GSM mahdollistaa entistä laajemman skannauksen yrityskäyttöön, ja useita muita lisäominaisuuksia. (Greenbone n.d.a.)

4.3 Nessus

Nessus on yksi laajimmin käytetyistä haavoittuvaisuuksien kartoittamiseen käytettävistä työkaluista yli 30 000 yrityksen käyttäessä sitä maailmanlaajuisesti. Nessuksesta on olemassa kolme eri versiota: Essentials, Professional ja Tenable.io. Essentials on opetus- ja yksityiskäyttöön tarkoitettu ilmainen versio, jolla voidaan skannata korkeintaan 16 IP-osoitetta kerrallaan. Professional on konsulteille, penetraatiotestaajille ja tietoturvan ammatinharjoittajille tarkoitettu maksullinen versio, jossa skannattavia IP-osoitteita ei olla rajoitettu. Viimeinen versio Tenable.io on laajan yrityskäyttöön tarkoitettu versio työkalusta, joka antaa mahdollisuuden ottaa käyttöön useita Nessus-skannereita yhden sijaan. (Tenable 2020.)

Nessus julkaistiin ensimmäistä kertaa Renaud Deraisonin toimesta vuonna 1998, ja se on vuosien varrella kehittynyt yksinkertaisesta haavoittuvaisuusskannerista monipuolisemmaksi arviointi- ja auditointityökaluksi. Parhaiten Nessus on kuitenkin edelleen tunnettu sen haavoittuvaisuusskannerin helppokäyttöisyydestä. Nessuksen haavoittuvaisuusskanneri kattaa verkkolaitteet, virtuaaliympäristöt, käyttöjärjestelmät, tietokannat sekä verkkosovellukset. (Himanshu 2013, 28.)

4.4 Nmap

Nmap, jonka nimi tulee englannin kielen sanoista Network Mapper, on ilmainen verkkoskannaukseen ja tietoturva-auditointiin tarkoitettu avoimen lähdekoodin työkalu. Nmap käyttää IP-paketteja selvittääkseen verkossa olevien laitteiden useita ominaisuuksia, kuten mitä palveluita laitteilla on käynnissä tai mitä käyttöjärjestelmiä laitteet käyttävät. Nmap on suunniteltu suurten verkkojen nopeaan skannaamiseen, mutta soveltuu yksittäistenkin laitteiden skannaamiseen. Nmap tukee kaikkia suuria käyttöjärjestelmiä, ja viralliset asennuspaketit ovat saatavilla Linuxille, Windowsille sekä Mac OS X:lle. (Nmap n.d.a.)

Kali Linuxin mukana tulee esiasennettuna Nmap, sekä sen sisartyökalut Nping, Ndiff, ja Ncat. Nping on verkkopakettien luomiseen, lähettämiseen ja vastausten analysoimiseen tarkoitettu työkalu, kun taas Ndiff-ohjelman tarkoitus on vertailla Nmap-skannausten tuloksia. Nmapin viimeinen sisartyökalu, eli Ncat, on datan siirtoon, uudelleenohjaukseen, sekä virheenkorjaukseen tarkoitettu työkalu. (Kali, 2020b.)

4.5 NXLog

NXLog on lokien keskittämiseen tarkoitettu ohjelma, jonka kehittäminen aloitettiin 2009 suljetun lähdekoodin ohjelmana. Sovelluksesta on olemassa maksullinen Enterprise Edition -versio ja ilmainen Community Edition -versio, jonka lähdekoodi avattiin julkiseksi 2011. NXLog voi prosessoida tapahtumalokeja tuhansista eri lähteistä, ja se tukee Linuxia, Windowsia ja Androidia. NXLog voi vastaanottaa lokeja useiden eri tiedonsiirtotekniikoiden, kuten TCP:n tai UDP:n kautta, sekä TLS- tai SSL-salattuna. NXLog voi myös vastaanottaa lokeja useassa eri formaatissa, kuten Syslog, Windows Event Log tai suoraan JSON-tiedostoina. NXLogista voidaan asentaa konfiguraatiosta riippuen joko palvelin- tai päätelaiteversio tai yhdistelmä näistä kahdesta. (NXLog 2020.)

NXLog on suunniteltu hyödyntämään moderneja moniytimisiä prosessoreja, joka takaa korkean suorituskyvyn. Ohjelmiston arkkitehtuuri on skaalautuva ja modulaarinen, jonka seurauksena sovellusta on mahdollista mukauttaa pitkälläkin aikavälillä. NXLog ei kuitenkaan ole kokonaisvaltainen lokienhallintajärjestelmä, vaan se tulee yhdistää muihin järjestelmiin datan koostamiseksi, analysoimiseksi ja tallettamiseksi. (NXLog 2020.)

4.6 Elastic Stack

Elastic Stack, joka tunnetaan myös nimellä ELK-stack, on joukko Elastic NV:n kehittämiä lokienhallintatyökaluja. Elastic Stackin avulla voidaan luotettavasti ja turvallisesti ottaa dataa mistä tahansa lähteestä missä tahansa formaatissa, jonka jälkeen dataa voidaan käsitellä ja visualisoida halutusti reaaliajassa. Elastic Stackiin kuuluu lokien analysointiin tarkoitettu Elasticsearch, lokidatan visualisointiin luotu Kibana, sekä lokien keräämiseen tarkoitettu Logstash-sovellus. Tämän lisäksi lokidatan lähettämiseen voidaan käyttää Elasticin Beats-sovelluksia. (Elastic 2020d.)

Beats-sovellukset ovat avoimen lähdekoodin sovelluksia, jotka asennetaan palvelimille tai työasemille lähettämään dataa Elastic stackille. Erilaisten lokien lähettämiseen on erilaisia Beats-sovelluksia, kuten verkkoliikenteen seuraamiseen tarkoitettu Packetbeat, tai Windowsin tapahtumalokien seuraamiseen käytettävä Winlogbeat. Jos omaan käyttötarkoitukseen sopivaa Beats-sovellusta ei löydy, on myös mahdollista luoda oma community beat -sovellus. (Elastic 2020a.)

Logstash on avoimen lähdekoodin datankeräysmoottori, jolla voidaan yhdistää dataa useista eri lähteistä, kuten Beats-sovelluksista, ja normalisoida se haluttuun muotoon. Logstash tukee yli 200:aa lisäosaa, ja tarjoaa mahdollisuuden omien lisäosien luomiseen, jonka johdosta Logstash tukeekin kaikenlaisia lokityyppejä. Datan normalisoinnin jälkeen data voidaan siirtää helposti analysoitavaksi Elasticsearchiin, tai moniin muihin arkistointi-, analyysi- tai monitorointityökaluihin. (Elastic 2020c.)

Elasticsearch on Elastic Stackin haku- ja analyysimoottori, joka mahdollistaa Logstashin ja Beat-sovellusten keräämän datan analysoinnin, tehokkaan tallentamisen ja indeksoinnin tavalla, joka mahdollistaa nopeat haut. Elasticsearch tarjoaa lähes reaaliaikaisen haun ja analyysin riippumatta datatyypistä. Elasticsearchin hajautettu luonne mahdollistaa myös skaalautumisen datamäärän kasvaessa. (Elastic 2020e.)

Elastic Stackin viimeinen osa, Kibana, on avoimen lähdekoodin sovellus Elastic Stackin visuaaliseen hallinnoimiseen. Kibanalla voidaan luoda erilaisia visualisointeja ja raportointinäkymiä Elasticsearchin datasta, jonka lisäksi Kibanan kautta voidaan myös hoitaa Elastic Stackin hallinnollisia tehtäviä, kuten muokata käyttäjäoikeuksia tai hallinnoida tietoturva-asetuksia. (Elastic 2020b.)

4.7 Graylog

Graylog on pääasiassa keskitetty lokienhallinta-alusta, joka pystyy vastaanottamaan, käsittelemään ja tallettamaan jopa useita teratavuja dataa päivittäin. Graylogin web-käyttöliittymä mahdollistaa halutun informaation etsimisen lokeista, ja sen lajittelun erilaisiin taulukoihin ja kaavioihin. Graylogista on tällä hetkellä kolme versiota: ilmainen avoimen lähdekoodin versio, ilmainen enterprise-versio, jossa on enterprise-version mukana tulevat lisäominaisuudet, mutta datan lajittelu on rajoitettu viiteen gigatavuun päivässä, sekä varsinainen enterprise-versio, jonka hinta perustuu päivittäiseen käsiteltävän datan määrään. (Graylog 2020.)

Graylog voi vastaanottaa melkein minkä tyyppistä dataa tahansa useiden eri tiedonsiirtoprotokollien, kuten TCP:n ja UDP:n kautta. Kun data on vastaanotettu, voidaan se organisoida Graylogin REST API:n kautta tai käyttämällä järjestelmän web-käyttöliittymää. Streams-ominaisuuden avulla voidaan erottaa tietyntyyppiset lokit, kuten virheilmoitukset muusta datasta, ja prosessoida ne halutulla tavalla. Lokeja voidaan muokata tai ne voidaan poistaa kokonaan lokien liikkuessa Graylogin läpi käyttämällä järjestelmän putkitusominaisuuksia. Tämän lisäksi lokiviesteihin voidaan myös lisätä kenttiä ja arvoja hakutaulukkojen avulla. Kun data on saatu järjestelmään, Graylogin Enhanced Search, Search Workflow ja Dashboards -ominaisuudet mahdollistavat kohteiden haun datasta, useiden hakujen yhdistämisen yhteen toimintoon ja datan visualisoinnin useissa erilaisissa raportointinäkymissä, jotka voidaan muokata halutunlaisiksi. (Graylog 2020.)

Graylog tarjoaa myös kattavat työkalut datan siirtämiseen toisiin järjestelmiin. Graylogista on mahdollista lähettää tilastoja säännöllisin aikavälein suoraan sähköpostiin, hakea niitä käyttämällä Graylogin APIa, tai lähettää dataa sellaisenaan eteenpäin toisiin järjestelmiin. Tämän lisäksi Graylogissa on useita tietoturvallisuutta ja saatavuutta parantavia ominaisuuksia, joka on kriittistä, sillä järjestelmä sisältää usein arkaluontoista dataa. Graylog kerää omia lokeja sen käyttäjien toiminnoista, mahdollistaa lokien arkistoinnin toisiin sijainteihin sekä sisältää roolipohjaisen käyttäjäoikeushallinnan. Graylog voidaan myös suunnitella skaalautuvaksi aina muutamista gigatavuista useaan teratavuun päivässä. Jos nämä ominaisuudet eivät riitä, voidaan Graylogia laajentaa myös Graylog-yhteisön avoimen lähdekoodin lisäosien avulla. (Graylog 2020.)

5 TESTIYMPÄRISTÖN SUUNNITTELU

5.1 Fyysinen ympäristö ja tietoverkot

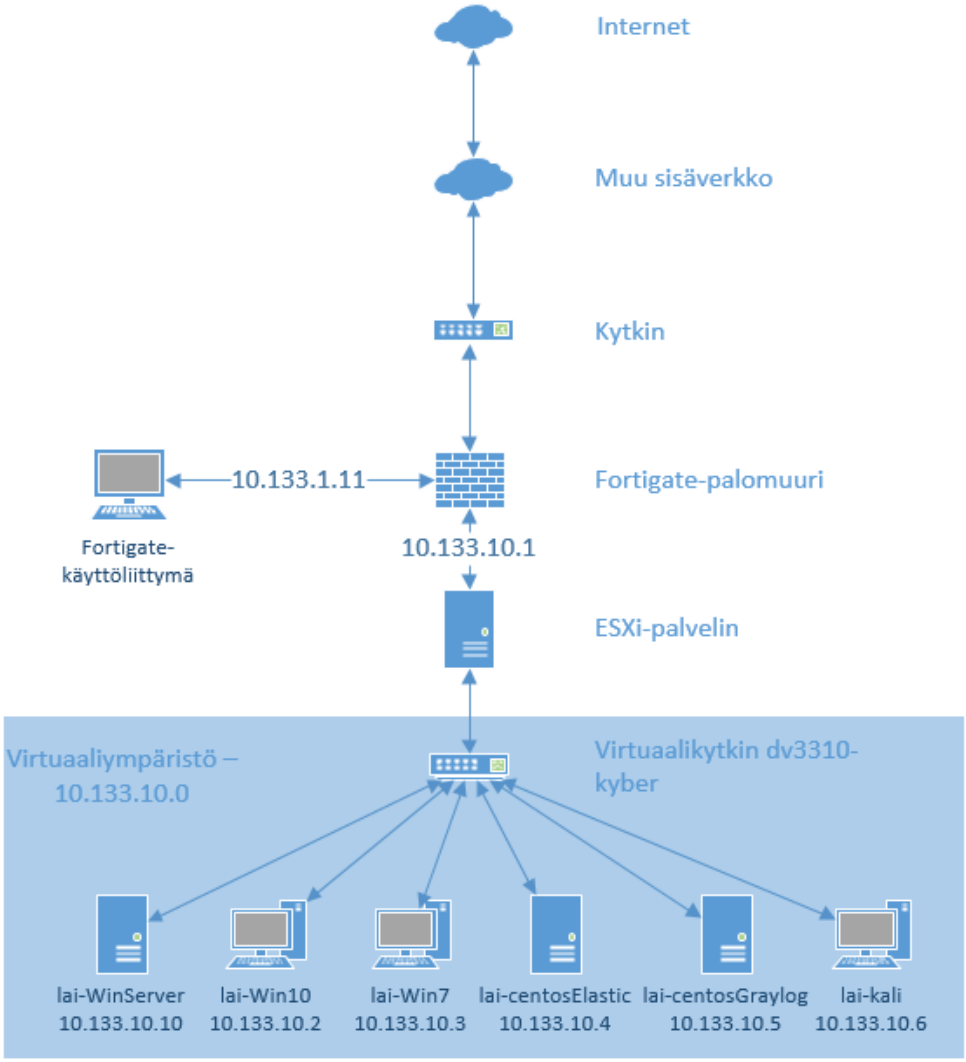
Testiympäristö luodaan käyttämällä VMwaren vSphere-virtualisointialustaa, joka muodostuu ESXi-hypervisorista, sekä vSpheren hallinnointiin tarkoitettusta vCenteriä, jotka on jo valmiiksi asennettu omalle palvelimelleen. ESXi-ohjelmisto on tyypin 1 hypervisor, eli se kommunikoi suoraan palvelimen fyysisen laitteiston kanssa ilman, että välissä olisi toista käyttöjärjestelmää. vSpherellä voidaan helposti luoda useita virtuaalikoneita palvelimelle ja yhdistää ne samaan verkkoon käyttämällä VMwaren virtualisointiratkaisuja. Virtualisoimalla voidaan ympäristö luoda yhdelle palvelimelle nopeasti, ja käyttämällä VMWaren työkaluja on palautuspisteiden luominen ja niihin palaaminen helppoa.

Fortigate-palomuuria käytetään testiympäristön eristämiseen muusta yrityksen verkosta, ja sen avulla voidaan kontrolloida suljetun verkon ja muun verkon välistä tietoliikennettä. Tämä on tärkeää, ettei ympäristöä käytettäessä aiheuteta haittoja muun verkon toimintaan. Palomuuriin on konfiguroitu kolme virtuaalilähiverkkoa eli VLANia, joista yksi on kytkimen ja palomuurin välinen VLAN 3397, toinen palomuurin ja virtuaaliympäristön välinen VLAN 3310, ja kolmas palomuurin hallintaan tarkoitettu VLAN 3301. Palomuuri on yhdistetty vuorostaan Huawein kytkimeen, jonka kautta ympäristö on yhdistetty muuhun yrityksen sisäverkkoon.

5.2 Virtuaalinen ympäristö

Virtuaaliympäristöön luodaan kolme palvelinta (Windows Server ja kaksi CentOS-palvelinta) ja kolme työasemaa (Kali, Windows 7 ja Windows 10). Kali-työasemaan sekä CentOS- ja Windows Server -palvelimille annetaan neljä virtuaalista prosessoria ja 16 Gt muistia. Windows-työasemille annetaan vähemmän resursseja, kaksi virtuaalista prosessoria ja 8 Gt muistia. Tämä johtuu sitä, ettei Windows-työasemilla tulla tekemään mitään, mikä vaatisi huomattavia määriä resursseja, vaan työasemat tulevat toimimaan vain penetraatiotestaamisen kohteina, sekä lokituksen testialustoina. Lisäksi kaikille laitteille annetaan 70 Gt virtuaalinen kiintolevy. Koneiden ollessa virtuaalisia, voidaan näitä ominaisuuksia muuttaa myös jälkikäteen. Tarvittaessa voidaan palvelimelle tulevaisuudessa lisätä työasemia ja palvelimia, jos niitä testaamiseen tarvitaan. Tarkemmat tiedot

sekä fyysisen että virtuaalisen ympäristön laitteista ja muusta verkosta todetaan kuvassa 2.



Kuva 2. Testiympäristön verkkokaavio. Sinisellä pohjalla virtuaaliympäristö.

Virtuaalisten laitteiden hallinta on mahdollista vCenterin kautta selaimella tai VMwaren VMRC-sovelluksen kautta. Kaikilla päätelaitteilla on graafinen käyttöliittymä CentOS-palvelimia lukuun ottamatta, joista asennetaan CentOS-käyttöjärjestelmän minimalistinen versio. Näiden palvelinten käyttö tapahtuu vain komentorivin kautta.

Jo aikaisemmin esitellyt työkalut, joita ympäristössä käytetään, valittiin muutaman eri ominaisuuden perusteella. Osa työkaluista on sellaisia, jotka ovat jo tällä hetkellä tuotantokäytössä yrityksessä, kuten lokienhallintasovellus Graylog, sekä haavoittuvaisuusskanneri Nessus. Osaa työkaluista taas ei yrityksellä ole lainkaan käytössä, vaan

ympäristöä halutaan käyttää näiden työkalujen testaamiseen. Näiden työkalujen valinnassa vaikutti muun muassa avoin lähdekoodi, aikaisemmat kokemukset ja kiinnostus työkalujen käyttöön, sekä yhteensopivuus muiden työkalujen kanssa. Esimerkkejä näistä työkaluista ovat penetraatiotestaukseen suunniteltu Kali-käyttöjärjestelmä, sekä Elastic Stack -nimellä tunnettu kokoelma lokienhallintatyökaluja.

5.2.1 Kali Linux -työasema

Kali Linux tullaan asentamaan käyttöjärjestelmäksi yhdelle testiympäristön päätelaitteista, joka tulee toimimaan penetraatiotestauksen kulmakivenä. Tätä työasemaa tullaan käyttämään penetraatiotestauksen suorittamiseen, Windows-työasemien toimiessa pääasiallisina kohteina. Kalin sisältäessä suuren määrän penetraatiotestaustyökaluja jo valmiiksi, on se paras käyttöjärjestelmä tähän käyttötarkoitukseen. Penetraatiotestaamista varten Kaliin asennetaan erikseen Nessus ja OpenVAS -ohjelmistot, joiden lisäksi käytössä on useita esiasennettuja sovelluksia, kuten Nmap, Burp Suite ja Metasploit.

5.2.2 Windows-työasemat

Ympäristöön luodaan kaksi normaalia työasemaa vastaavaa tietokonetta, joista toisessa on vanhempi Windows 7 -käyttöjärjestelmä, ja toisessa uudempi Windows 10 -työasema. Näitä työasemia käytetään sekä penetraatiotestauksen kohteina, että lokituksen testaamiseen. Vaikka suurin osa yrityksen ja sen asiakkaiden työasemista onkin jo päivitetty tai tullaan tulevaisuudessa päivittämään uudempaan Windows 10 -käyttöjärjestelmään, on silti Windows 7 -työasemia vielä jonkin verran käytössä tehtävissä, joissa uudemmat käyttöjärjestelmät eivät ole yhteensopivia vanhojen sovellusten kanssa. Tämän vuoksi myös Windows 7 -työasemien tietoturvan ja lokituksen toiminnan testaaminen on yhä tärkeää. Lokidatan lähettämiseksi CentOS-palvelimille asennetaan työasemille NXLog-sovellus.

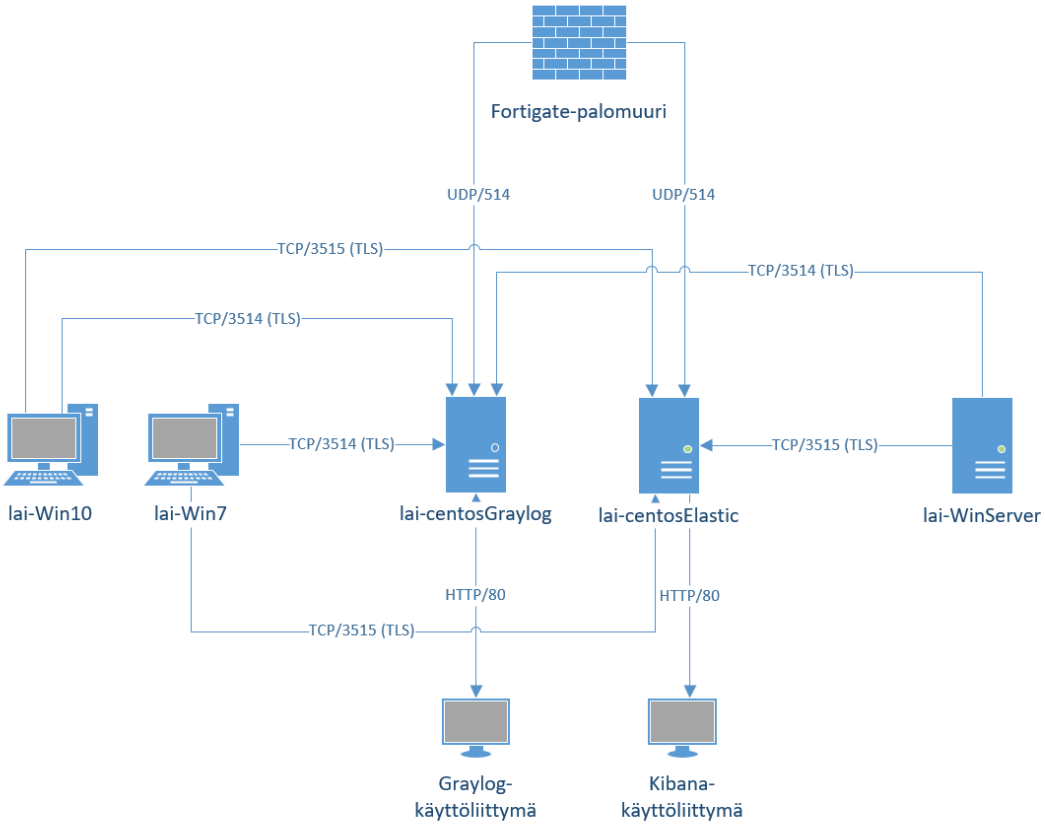
5.2.3 Windows Server 2019 -palvelin

Testiympäristöön luodaan Windows Server 2019 -palvelin Domain Controller -palvelimeksi. Tämä palvelin isännöi AD-, DNS- ja DHCP-palveluja. Näillä palveluilla ylläpidetään ympäristön käyttöoikeuksia, verkkotunnusten muuntamista IP-osoitteiksi, sekä

tarvittaessa IP-osoitteiden automaattinen asettaminen ympäristön muille työasemille. Tämän lisäksi palvelimelle asennetaan NXLog-sovellus lähettämään lokeja molemmille CentOS-palvelimille, joilla varsinainen lokien käsittely tehdään.

5.2.4 CentOS 8 -palvelimet

Windows Server -palvelimen lisäksi ympäristöön luodaan kaksi Centos-palvelinta, joita käytetään pääasiassa lokien keskittämiseen ja lokituksen testaamiseen. Toiselle palvelimista asennetaan Graylog ja toiselle Elastic Stackin osat Elasticsearch, Kibana ja Logstash. Kummatkin palvelimet keräävät lokitietoja kaikilta ympäristön työasemilta Kali-tietokonetta lukuun ottamatta, sekä Fortigaten palomuurilta. Näin voidaan testata kahta eri lokienhallintajärjestelmää yhtäaikaisesti. Graylogin ja Elastic Stackin hallinta tapahtuu palvelimen komentorivin lisäksi käyttämällä kummankin järjestelmän tarjoamaa web-käyttöliittymää. Kuvasta 3 voidaan tarkastella, kuinka lokit kulkevat ympäristön laitteilta lokipalvelimille.

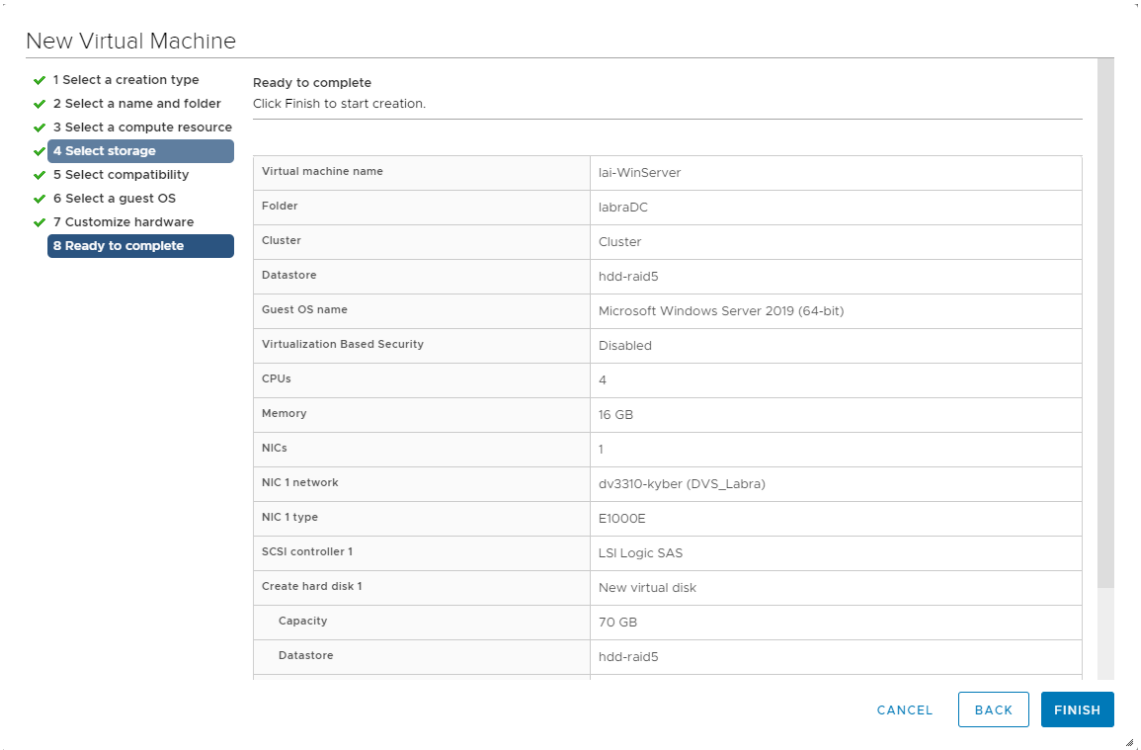


Kuva 3. Tiedonsiirtoprotokollat ja portit, joita lokitukseen käytetään.

6 TESTIYMPÄRISTÖN TOTEUTUS

6.1 Virtuaalikoneiden asennus

Ympäristön toteutus aloitettiin lataamalla kaikkien käyttöjärjestelmien ISO-asennustiedostot palvelimelle, jonne testiympäristö toteutetaan. Kaikki päätelaitteet luotiin kokonaan uusina virtuaalikoneina lukuun ottamatta toista CentOS-palvelinta, jonka luomisessa pystyttiin hyödyntämään yhtä virtualisoinnin suurimmista eduista, eli kloonaamista. Kaikki työasemat liitettiin jo valmiiksi luotuun verkkoon dv3310-kyber, joka on yhteydessä muuhun verkkoon Fortigate-palomuuriin kautta. Windows-pohjaisille laitteille asetettiin SCSI-ohjaimeksi LSI Logic SAS, kun taas Linux-pohjaisilla laitteilla käytettiin VMWare Paravirtual -ohjainta, joka oli ainoa käyttöjärjestelmien vaatima eroavaisuus virtuaalikoneiden konfiguroinneissa. Työasemien kovalevyt luotiin käyttämällä VMWaren thin provisioning -ominaisuutta, jolloin työasemat käyttävät vain niin paljon tallennustilaa kuin tarvitaan, eivätkä koko levytilaa, joka virtuaalikoneelle on varattu. Kuvassa 4 ovat näkyvissä Windows Server -virtuaalikoneelle asetetut ominaisuudet.



Kuva 4. Windows Server -virtuaalikoneen asetukset.

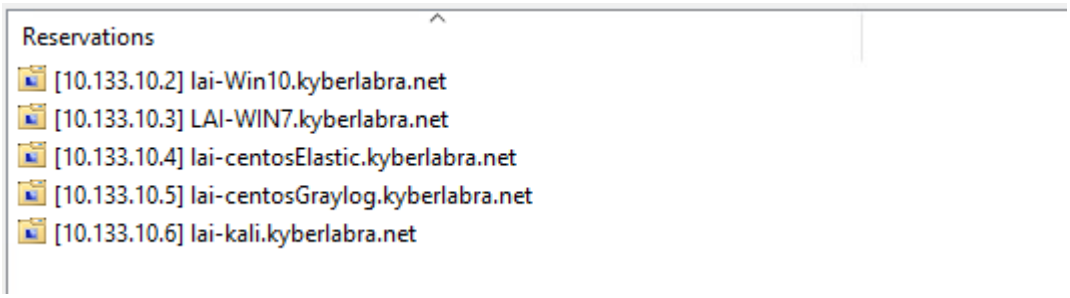
Virtuaalikoneiden asetusten ja ominaisuuksien määrityksen jälkeen oli seuraavana itse käyttöjärjestelmien asentaminen. Tämä prosessi on jokaisen käyttöjärjestelmän osalta samanlainen kuin käyttöjärjestelmän asennus fyysiselle päätelaitteellekin. Käyttöjärjestelmien asennukset eivät myöskään eroa huomattavasti toisistaan: kaikkien asennukset suoritetaan graafisen käyttöliittymän kautta, niissä valitaan käytettävä kieli, näppäimistöasettelu, aika-asetukset, verkkoasetukset ja levy jolle käyttöjärjestelmä asennetaan. Tämän lisäksi jokaisen asennuksen yhteydessä luodaan työasemalle vähintään yksi käyttäjätili. Kun työasemien asennukset olivat valmiit, hyödynnettiin jälleen yhtä virtualisoinnin etua eli palautuspisteitä. Kaikista päätelaitteista tehtiin palautuspiste välittömästi asennusten valmistuttua, jotta tarvittaessa pystytään palaamaan alkuun, jos jotain myöhemmin menee pieleen.

6.2 Active Directory, DHCP ja DNS

Työasemien palvelimelle luonnin ja käyttöjärjestelmien asentamisen jälkeen luotiin Windows Serverin ominaisuuksia käyttäen domain, johon Windows-laitteet liitetään. Tämä mahdollistaa muun muassa yhden käyttäjätunnuksen käyttämisen kaikilla Windows-laitteilla, sekä työasemien keskitetyn hallinnoinnin. Tämän lisäksi Windows Server -palvelimelle asennettiinn DHCP- ja DNS-palvelimet, joita käytetään IP-osoitteiden jakamiseen, ja työasemien verkkotunnusten muuntamiseen IP-osoitteiksi. DNS-palvelin yhdistettiin DHCP-palvelimeen siten, että aina DHCP-palvelimen jakaessa IP-osoitteen, välittää se myös tiedon IP-osoitteesta sekä sen saajan nimestä DNS-palvelimelle. Ennen näiden palveluiden asennusta Windows Server -palvelimelle asetettiin staattinen IP-osoite 10.133.10.10, ja oletusyhdyskäytäväksi Fortigate-palomuurin staattinen IP-osoite 10.133.10.1.

Active Directoryn asentamiseen käytettiin Windows Serverin tarjoamaa ohjeistettua asennusta, ja domainin nimeksi asetettiin kyberlabra.net. Tämän jälkeen DHCP-palvelimelle luotiin uusi IP-ulottuvuus, jonka alueelta DHCP-palvelin jakaa IP-osoitteita. Ulottuvuuden nimeksi annettiin Kyberlabra, ja sen alueeksi IP-osoitteet väliltä 10.133.10.1–10.133.10.254. Tästä ulottuvuudesta kuitenkin jätettiin pois osoitteet 10.133.10.1 ja 10.133.10.10, sillä ne ovat varattuja palomuurille ja Windows Serverille, sekä osoitteet, jotka ovat suurempia kuin 10.133.10.20. Laitteiden oletusyhdyskäytäväksi DHCP asettaa palomuurin IP-osoitteen, ja aliverkon peitteeksi 255.255.255.0. Tämän jälkeen varmistettiin, että kaikki laitteet saavat IP-osoitteen, jonka jälkeen jokaiselle laitteelle luotiin

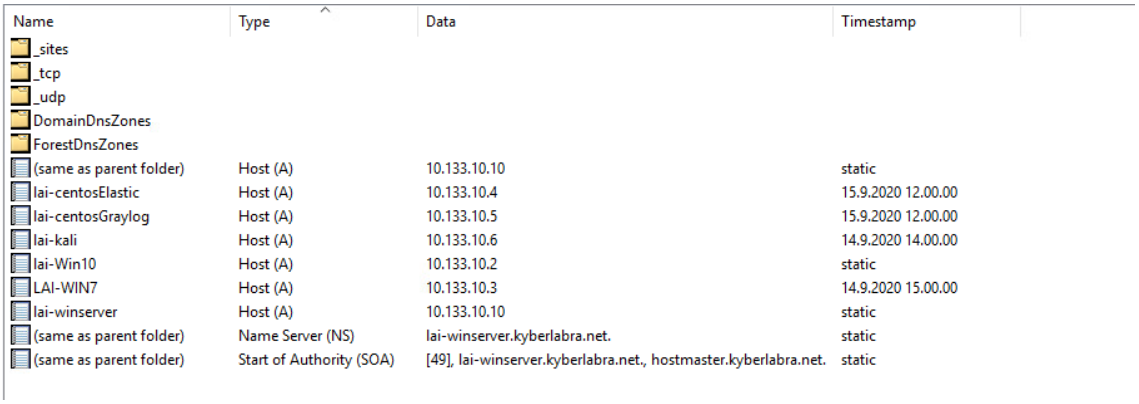
oma IP-osoitevaraus DHCP:hen, kuten voidaan kuvasta 5. Näin laitteiden IP-osoitteet pysyvät aina samana ilman, että jokaiselle työasemalle täytyy manuaalisesti asettaa verkkoasetuksia.



Kuva 5. IP-osoitevaraukset.

Varatut IP-osoitteet kirjattiin yrityksen käyttämään Device42-järjestelmään, jolla muun muassa pidetään kirjaa käytössä olevista IP-osoitteista. Windows-päätelaitteet liitettiin luotuun domainiin kyberlabra.net, ja Active Directoryn avulla luotiin käyttäjätunnukset kaikille ympäristöä käyttäville henkilöille. Koska kyseessä on testiympäristö ja ympäristön Windows-laitteiden määrä on pieni, ei Active Directoryn kaikkia hienouksia, kuten ryhmäkäytäntöjen keskitettyä jakamista, ollut tarpeellista ottaa käyttöön.

Lopulta konfiguroitiin DNS-palvelin. DNS-palvelimelle luotiin Forward Lookup Zone joka kääntävät verkkonimet IP-osoitteiksi, ja Reverse Lookup Zone, joka kääntää IP-osoitteet verkkonimiksi. Tämän jälkeen pyydettiin jokaisella laitteella DHCP-palvelimelta päivitetty IP-osoite, jonka seurauksena DHCP-palvelimen jakaessa osoitteen, siirtyi myös tieto osoitteesta ja työaseman nimestä DNS-palvelimelle, kuten voidaan nähdä kuvasta 6.



Name	Type	Data	Timestamp
_sites			
_tcp			
_udp			
DomainDnsZones			
ForestDnsZones			
(same as parent folder)	Host (A)	10.133.10.10	static
lai-centosElastic	Host (A)	10.133.10.4	15.9.2020 12.00.00
lai-centosGraylog	Host (A)	10.133.10.5	15.9.2020 12.00.00
lai-kali	Host (A)	10.133.10.6	14.9.2020 14.00.00
lai-Win10	Host (A)	10.133.10.2	static
LAI-WIN7	Host (A)	10.133.10.3	14.9.2020 15.00.00
lai-winserver	Host (A)	10.133.10.10	static
(same as parent folder)	Name Server (NS)	lai-winserver.kyberlabra.net.	static
(same as parent folder)	Start of Authority (SOA)	[49], lai-winserver.kyberlabra.net., hostmaster.kyberlabra.net.	static

Kuva 6. DNS-palvelimen Forward Lookup Zone.

6.3 NXLogin asennus Windows-laitteille

Ensimmäisenä lokisovelluksista asennettiin Windows-laitteille asennettava NXLog. NXLogista asennettiin avoimen lähdekoodin Community Edition -versio, jota käytetään Event Log -lokien lähettämiseen Windows-laitteilta CentOS-palvelimille. Lokien lähettämiseen kummallekin palvelimelle käytetään TLS-salattua TCP-yhteyttä. Elastic Stackille lähettäessä käytetään lokimuotona JSON-formaattia, kun taas Graylogille lähettäessä Graylogin omaa GELF-formaattia, joka toimii kuten normaali JSON-formaatti sillä erotuksella, että GELF-formaatissa on muutamia ylimääräisiä kenttiä. Kuvassa 7 on kuvattu, kuinka NXLog käsittelee ja lähettää lokit eteenpäin lokipalvelimille.

```
<Extension json>
    Module      xm_json
</Extension>

<Extension gelf>
    Module      xm_gelf
</Extension>

<Input eventlog>
    Module      im_msvistalog
</Input>

<Output logstash>
    Module              om_ssl
    Host                10.133.10.4
    Port                3515
    CAFile              C:\Program Files (x86)\nxlog\cert\ca.crt
    CertFile            C:\Program Files (x86)\nxlog\cert\client.crt
    CertKeyFile         C:\Program Files (x86)\nxlog\cert\client.key
    AllowUntrusted      TRUE
    Exec                to_json();
</Output>

<Output graylog>
    Module              om_ssl
    Host                10.133.10.5
    Port                3514
    CAFile              C:\Program Files (x86)\nxlog\cert2\ca.crt
    CertFile            C:\Program Files (x86)\nxlog\cert2\client.crt
    CertKeyFile         C:\Program Files (x86)\nxlog\cert2\client.key
    AllowUntrusted      TRUE
    OutputType          GELF_TCP
</Output>

<Route graylog_route>
    Path        eventlog => graylog
</Route>

<Route elastic>
    Path        eventlog => logstash
</Route>
```

Kuva 7. NXLogin konfiguraatio.

NXLogiin lisättiin JSON- ja GELF-lisäosat xm_json ja xm_gelf, sekä syötteeksi im_msvistalog, jolla voidaan kerätä Windowsin Event Log -lokeja moderneilta Windows-laitteilta (Windows Vista tai uudempi). Tämän jälkeen määritettiin lokien lähdöt, eli minne NXLog

lähettää sen keräämät lokit. Määriteltiin kaksi lähtöä, yksi palvelimelle jolle Elastic Stack tullaan asentamaan, ja toinen palvelimelle jolle Graylog tullaan asentamaan. Kummallekin yhteydelle määriteltiin palvelimien IP-osoitteet, portti, sertifikaatit, sekä lokiformaattien vaatimat asetukset. Lopuksi määriteltiin reitit, joita lokit kulkevat, eli mikä syöte kulkee mihinkin ulostuloon. Tässä tapauksessa sama syöte eventlog määriteltiin kulkemaan kumpaankin lähtöön.

6.4 Elastic Stackin asennus

Lokeja käsittelevien sovellusten asennus aloitettiin Elastic Stackin, eli Elasticsearchin, Kibanan ja Logstashin asennuksesta Centos-palvelimelle lai-centosElastic. Ennen asennusta lisättiin Elasticsearchin avain ja repositorio palvelimelle, sekä asennettiin uusin versio avoimen lähdekoodin Java-sovelluksesta OpenJDK. Tämän jälkeen Elastic Stackin asennus aloitettiin Elasticsearchista, jonka asetukset jätettiin tässä kohtaa oletusasetuksiin.

Kibanan portiksi määritettiin oletusportti 5601, ja se yhdistettiin samalla palvelimella toimivaan Elasticsearchiin käyttämällä osoitetta http://127.0.0.1:9200. Kibanaa varten asennettiin palvelimelle myös Nginx-sovellus, jota käytetään Kibanan edessä toimivana reverse proxyna. Näin saadaan ohjattua osoitteeseen 10.133.10.4 tuleva liikenne porttiin 5601 eli Kibanan porttiin. Tämän seurauksena käyttäjien ei tarvitse muistaa porttinumeroa, vaan he voivat syöttää vain palvelimen IP-osoitteen selaimeen ja päästä Kibanaan. Nginx myös mahdollistaa asettamaan palvelimelle oman verkko-osoitteen, jolloin käyttäjien ei tarvitse käyttää IP-osoitetta Kibanaan päästäkseen. Myös tietoturvan parantaminen Nginxin avulla on mahdollista: Kibanaan voidaan asettaa käyttäjätunnus ja salasana, ja liikenne voidaan salata https-tekniikalla.

Viimeisenä osana Elastic Stackia asennettiin Logstash, joka vastaanottaa palvelimelle lähetettyä lokidataa ja syöttää sen eteenpäin Elasticsearchille. Liikenteen salaamiseen käytetään TLS-salausta. Vaikka ympäristö ei olekaan ulkopuolisille avoin, on silti testaamisen kannalta toivottavaa, että ympäristön ominaisuudet vastaisivat ominaisuuksia, jotka oikealla lokijärjestelmällä olisi. Logstashin konfigurointi on toteutettu kahdessa osassa: toisessa osassa otetaan vastaan NXLogin lähettämiä Event Log -lokeja Windows-laitteilta, kun taas toisessa osassa otetaan vastaan palomuurin toimittamia lokeja syslog-muodossa. Kuvasta 8 voidaan nähdä, kuinka Logstash vastaanottaa NXLogin toimittamat lokit portin 3515 kautta TCP-yhteydellä, ja lähettää ne edelleen

Elasticsearchille. Koska Elastic Stack osaa käsitellä JSON-muodossa saapuvat Event Log-lokit ilman kenttien erittelyä, on konfigurointi hyvin yksinkertainen. Konfiguroinnissa määritetään portti jota Logstash kuuntelee, sertifikaatit joita se käyttää lähettäjän turvallisuuden varmentamiseen, aikaformaatti johon saapuvat lokit täsmätään, sekä Elasticsearchin portti ja indeksi, johon lokit lähetetään edelleen.

```
input {
        tcp {
                codec => json_lines { charset => CP1252 }
                port => "3515"
                ssl_enable => true
                ssl_certificate_authorities => ["/etc/pki/tls/certs/ca.crt"]
                ssl_cert => "/etc/pki/tls/certs/server.crt"
                ssl_key => "/etc/pki/tls/private/server.key"
                tags => [ "tcpjson" ]
                type => "nxlog"
        }
}
filter {
        date {
                locale => "en"
                timezone => "Europe/Helsinki"
                match => [ "EventTime", "YYYY-MM-dd HH:mm:ss" ]
        }
}
output {
        if [type] == "nxlog" {
                elasticsearch {
                        hosts => [ "localhost:9200" ]
                        index => "nxlog-%{+YYY.MM.dd}"
                }
                stdout { codec => rubydebug }
        }
}
```

Kuva 8. Logstashin konfigurointi saapuville Event Log -lokeille.

Palomuurilta syslog-muodossa saapuvat lokit vaativat hieman enemmän lokien käsittelyä. Tämä johtuu siitä, ettei Elasticsearch osaa käsitellä kaikkia syslogin kenttiä automaattisesti oikein. Syslogin kautta tuleva aika täytyy muuttaa oikeaan formaattiin, joidenkin arvojen tyyppi tulee muuttaa tekstistä numeroksi, sekä osa arvoista tulee poistaa kokonaan. Toisin kuin NXLogin lähettämissä lokeissa, ei palomuurin lokeissa käytetä TCP-yhteyttä vaan UDP-yhteyttä, jolla lokit vastaanotetaan portissa 1514. Saapuvien lokien käsittelyssä keskitytään ”message” -kentän paloitteluun. Konfiguroinnissa ohjeistetaan käyttämällä Grok-, KV- ja mutate-ominaisuuksia kuinka ”message” -kentän arvot on eroteltu, mitä arvoja lokeihin lisätään, mitä nimetään uudelleen, minkä kenttien arvoja muutetaan tekstistä numeroiksi ja mitä arvoja poistetaan kokonaan. Kuten Event Log -lokien tapauksessa, täsmätään aikaleimat oikeaan formaattiin ja aikavyöhykkeeseen. Kun lokit on saatu haluttuun muotoon, viedään ne edelleen Elasticsearchin porttiin 9200.

Jotta kaikki lokiliikenne saatiin toimimaan, tehtiin CentOS:n omalle palomuurille muutoksia. Koska syslogin oletusportti on 514, joka on suojattu portti, ohjattiin porttiin 514 saapuva liikenne ohjataan porttiin 1514, josta Logstash ottaa datan vastaan. Näin voidaan

saapuva syslog-data ottaa vastaan normaalissa syslog-portissa, ilman että Logstash tarvitsee root-oikeuksia. Palomuurilta saapuvia lokeja varten avattiin portti 514 UDP-liikenteelle, ja NXLogin lähettämiä lokeja varten avattiin portti 3515 TCP-liikenteelle. Lisäksi hyväksyttiin http-liikenne, jotta Kibanaan pääsy on mahdollista.

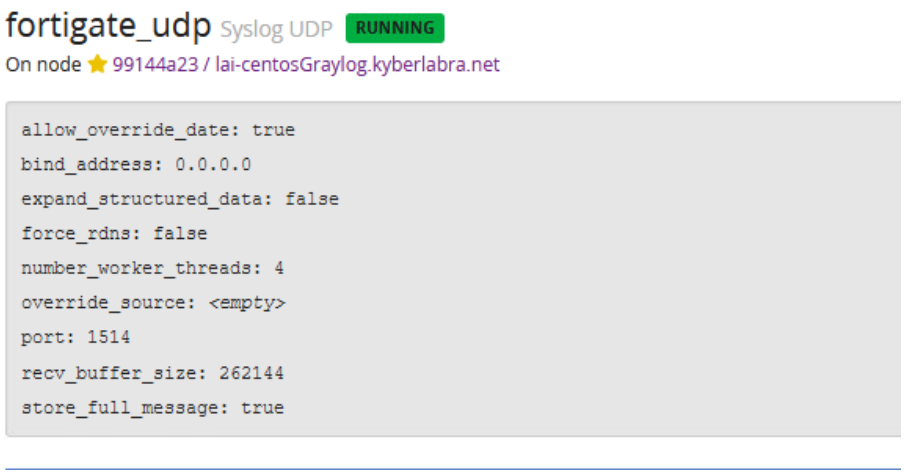
6.5 Graylogin asennus

Toiselle CentOS-palvelimista lai-centosGraylog asennettiin palvelimen nimen mukaisesti Graylog. Graylog toimii toisena testattavista lokienhallintajärjestelmistä, ja vastaanottaa dataa samoista lähteistä kuin Elastic Stack. Kuten Elastic Stack, tarvitsee myös Graylog Java-sovellus OpenJDK:n asennuksen. Toisena yhteisenä tekijänä toimii Elasticsearch, jota myös Graylog käyttää hakumoottorina. Lokien tallettamiseen käytetään tietokantasovellus MongoDB:n Community Edition -versiota. Jotta MongoDB saatiin toimimaan halutusti, piti asennuksen yhteydessä muuttaa SELinuxin asetuksia. SELinux on CentOS-käyttöjärjestelmään kuuluva Linux-ytimen laajennus, jonka tarkoituksena on mahdollistaa entistä tarkempi käyttöoikeuksien hallinta. SELinuxin asetusten seurauksena MongoDB:llä ei ole oikeuksia kaikkiin sen vaatimiin tiedostosijainteihin, vaan sille piti erikseen luvittaa pääsy kansioon /sys/fs/cgroup, jotta MongoDB pystyy seuraamaan palvelimen muistinkäyttöä.

MongoDB:n jälkeen asennettiin Elasticsearch, kuten edellisellekin palvelimelle, ja viimeisenä Graylog. Sekä Elasticsearchin, että Graylogin asetukset jätettiin tässä vaiheessa pitkälti oletusarvoihinsa, lukuun ottamatta salasanaa jota käytetään salasanojen salaamiseen, ilman jota Graylog ei suostu käynnistymään, sekä admin-käyttäjän salasanaa, joka vaaditaan web-käyttöliittymään kirjautumiseen. Tämän lisäksi asetettiin oikea aikavyöhyke, sekä web-käyttöliittymän osoitteeksi palvelimen IP-osoite, jotta käyttöliittymään päästään käsiksi myös muilta verkon laitteilta.

Toisin kuin Elastic Stackissä, jossa lokien konfigurointi jouduttiin tekemään erinäisten asetustiedojen kautta komentorivillä, onnistuu Graylogin konfigurointi asennuksen jälkeen pitkälti graafisen web-käyttöliittymän kautta, joka tarjosi hyvin käyttäjäystävällisen kokemuksen. Kuten Elastic Stackissa, luotiin sekä NXLogin lähettämille lokeille, että palomuurin lokeille oma lokikanava, jonka kautta lokit otetaan vastaan. Tämän tekeminen oli erittäin helppoa Graylogin käyttöliittymän kautta, jossa oli valmiit kentät, osassa tapauksista valmiine vaihtoehtoineen, jolloin asetusten syntaksia ei tarvinnut tuntea, kuten Logstashin kanssa. NXLogin lähettämät lokit otetaan vastaan portissa 3514 TLS-

yhteydellä GELF-muodossa, kun taas Fortigaten lokit otetaan vastaan portissa 1514 UDP-muodossa, kuten Elastic Stackissakin. Kuvassa 9 on kuvattu, kuinka palomuurilta saapuneet lokit otetaan vastaan.



Kuva 9. Asetukset, joiden mukaisesti palomuurilta saapuvat lokit käsitellään.

Koska Graylogissakin käytetään Elasticsearchia lokien indeksointiin ja lokeista datan etsimiseen, osaa myös Graylog käsitellä Event Log -lokit ilman erillisiä asetuksia. Palomuurilta saapuvien syslog-lokien käsittely vaatii kuitenkin hieman käsittelyä, jotta aikavyöhykkeet täsmäävät. Graylog automaattisesti luulee palomuurilta saapuvien lokien olevan UTC-aikavyöhykkeen mukaisia, vaikka todellisuudessa aikavyöhyke on jo palomuurilta korjattu oikeaksi. Tämän seurauksena myös Graylog yrittää korjata saapuvien lokien kellonajan, jolloin lokien aikaleimat ovat kolme tuntia pielessä.

Tämän ongelman korjaaminen ei ollut aivan yhtä helppoa kuin Graylogin yksinkertaisempien asetusten muuttaminen. Ongelma saatiin ratkaistua käyttämällä Graylogin putkitustoimintoa. Luotiin putki ”Fortigate UDP”, joka asetettiin käsittelemään kaikki saapuvat lokit. Käsittelyn ensimmäiselle tasolle (Stage 0) luotiin sääntö ”Fortigate timestamp”, jossa erotellaan kaikki lokit, joissa on kenttä ”tz”. Kyseinen kenttä on tässä ympäristössä käsiteltävistä lokeista uniikki kenttä palomuurin lokeille, joten se soveltui hyvin erottavaksi tekijäksi. Näistä lokeista otetaan vielä tarkemmin käsittelyyn kenttä timestamp, johon korjataan lokien oikea aika korvaamalla vanha aika korjatulla.

Lopuksi, kuten lai-centosElastic -palvelimella, täytyi Graylog-palvelimelle tehdä muutoksia palomuuriin. Tehdyt muutokset olivat hyvin samanlaisia kuin Elastic-palvelimelle

tehdyt muutokset. Hyväksyttiin http-liikenne, avattiin portti 514 UDP-liikenteelle ja ohjattiin siihen saapuva liikenne porttiin 1514 sekä avattiin portit 3514 ja 9000 TCP-liikenteelle.

6.6 Penetraatiotestaussovellusten asennus

Koska penetraatiotestauksen alustana käytetään kyseiseen käyttötarkoitukseen suunniteltua Kali-käyttöjärjestelmää, oli iso osa halutuista penetraatiotestaussovelluksista jo valmiiksi asennettuna käyttöjärjestelmän mukana. Muun muassa verkkosivujen ja -sovellusten testaamiseen tarkoitetut Owasp Zap ja Burp Suite, hyökkäyskoodien etsimiseen ja suorittamiseen luotu Metasploit Framework sekä verkkoskannaamiseen käytettävä Nmap ja sen lisäosat löytyvät kaikki jo esiasennettuina. Testiympäristöön halutuista työkaluista vain OpenVAS ja Nessus vaativat erillisen asennuksen.

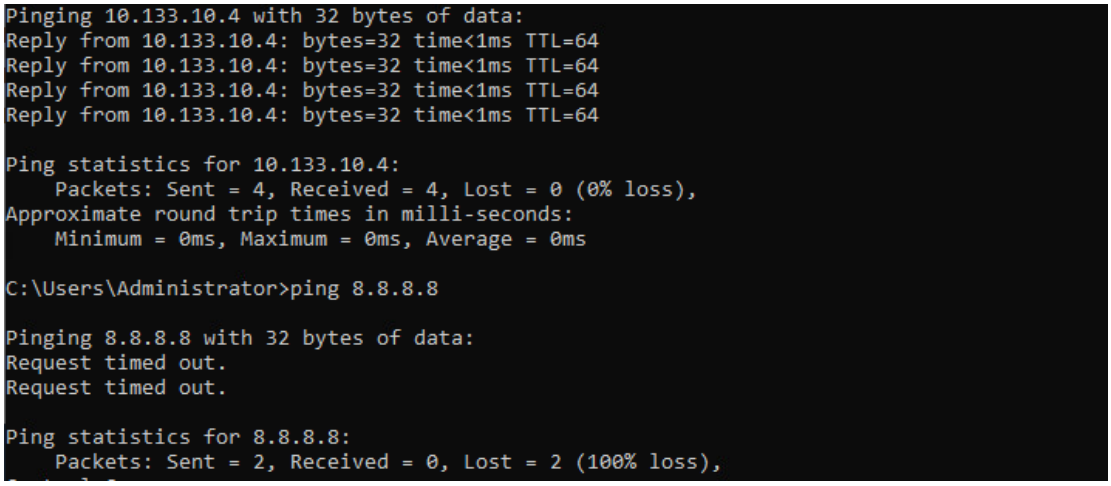
OpenVAS-sovelluksen asennus onnistui helposti komentorivin kautta. Pienenä erikoisuutena oli komentorivillä käytettävät komennot, joissa ei käytetä enää vanhaa OpenVAS-nimeä, vaan uutta sovelluksen kehittäjän mukaan nimettyä GVM-lyhennettä. Sovelluksen asentaminen vaati kuitenkin vain yhden komennon, ja konfigurointi toisen, joten asennus oli erittäin helppo toteuttaa. Konfiguroinnin kesto oli kuitenkin hyvin pitkä, OpenVASin ladatessa huomattavan määrän tiedostoja sen aikana. Konfiguroinnin valmistuttua antoi OpenVAS komentoriville automaattisesti luodun salasanan, jota voidaan käyttää Web-käyttöliittymään kirjautumiseen. Jäljellä oli enää OpenVASin käynnistäminen, ja sen avaaminen tietokoneen paikallisen IP-osoitteen 127.0.0.1 ja portin 9392 kautta.

Nessuksen asennus erosi kaikista muista ympäristön Linux-laitteille asennetuista sovelluksista siinä, että Nessuksen lataus tuli tehdä selaimen kautta, eikä suoraan komentoriviltä. Kun asennustiedosto oli ladattu, onnistui Nessuksen asentaminen kuitenkin normaalisti komentorivin kautta. Kun asennus oli valmis, asetettiin Nessus aukeamaan automaattisesti virtuaalikoneen käynnistyessä, jonka jälkeen suunnattiin tietokoneen paikalliseen IP-osoitteeseen 127.0.0.1, tällä kertaa porttiin 8834. Tämän jälkeen valittiin asennettavaksi versioksi ilmaisversio Nessus Essentials. Koska kyseessä ei ole tuotantoympäristö, ja Nessusta käytetään vain testikäyttöön, on tämä versio käyttötarkoitukseen sopivin. Erona muihin ympäristön sovelluksiin Nessus vaatii myös rekisteröitymisen, jonka jälkeen Nessuksen aktivointikoodi toimitetaan sähköpostitse. Kun

aktivointikoodi oli syötetty ja paikallinen tili Nessukseen kirjautumiseen oli luotu, viimeisteli Nessus asennuksen lataamalla viimeisetkin lisäosat, jonka jälkeen käyttö voitiin aloittaa.

7 YMPÄRISTÖN TESTAUS

Kun ympäristö oli käyttövalmis, oli aika testata, että kaikki toimii. Testiympäristössä on kaikkein tärkeintä, että tietoliikenne toimii halutusti, ja että palomuuri estää liikenteen ulkoverkkoon. Yhteyksien toiminta varmistettiin lähettämällä ICMP-paketteja Windows Server -palvelimelta muille ympäristön tietokoneille, sekä ulkoiseen osoitteeseen 8.8.8.8, joka on Googlen palvelin. Tämä vaati ICMP-pakettien hyväksymisen Windows-laitteiden omilta palomuureilta. Kuten kuvasta 10 voidaan nähdä, onnistui ICMP-pakettien lähettäminen tämän jälkeen ping-komennolla lai-centosElastic -palvelimelle, mutta palomuuri esti liikenteen verkon ulkopuolella sijaitsevalle Googlen palvelimelle.



Kuva 10. Palomuuri estää liikenteen ulkoverkkoon.

Kun verkko todettiin toimivaksi, päästiin lokien testaamiseen. Kibanaan luotiin kaksi index patternia, toinen tunnistamaan forti-alkuiset indeksit, ja toinen nxlog-alkuiset. Näin saadaan jaettua palomuurilta ja Windows-laitteilta saapuvat lokit kahteen eri näkymään. Graylogin puolella lokit jakautuvat automaattisesti omiin näkymiinsä lähteen perusteella, jonka johdosta konfiguraatio oli yksinkertaista. Kummallekin palvelimelle saatiin näkyviin sekä Windows-laitteilta saapuvat Event Log -lokit, että palomuurin syslog-lokit. Kuvassa 11 voidaan nähdä esimerkki lai-WinServer -palvelimelta saapuneesta lokista, jossa ilmoitetaan DNS-palvelimen poistaneen tallennetun nimen IP-osoitteesta 10.133.10.10 tulleen dynaamisen päivityksen takia.

t	EventReceivedTime	2020-11-09 11:51:18
t	EventTime	2020-11-09 11:51:16
t	EventType	INFO
t	Hostname	lai-WinServer.kyberlabra.net
#	Keywords	4,611,686,018,460,941,312
t	Message	A resource record of type 1, name lai-winserver and RDATA 0x0A850A0A was deleted from scope Default of zone kyberlabra.net via dynamic update from IP Address 10.133.10.10.
t	NAME	lai-winserver

Kuva 11. Ote DNS-palvelimen muutoksen aiheuttaneesta lokista Kibanassa.

Kun lokiliikenteen liikkuminen sekä Graylogiin että Elastick Stackiin todettiin toimivaksi, voitiin siirtyä testaamaan Kaliin asennettuja penetraatiotestaustyökaluja. Kalin työkalujen testaaminen aloitettiin Nmapilla, jolla tehtiin yksinkertainen skannaus kohdistuen Windows Server -palvelimeen. Nmap palautti useita tietoja palvelimesta, kuten avoimet portit sekä näitä käyttävät palvelut. Skannauksesta jäi kuitenkin jälki lokitietoihin (kuva 12).

2020-11-21 10:31:12 +02:00
The DNS server received a bad TCP-based DNS message from 10.133.

2020-11-21 10:30:46 +02:00
The DNS server received a bad TCP-based DNS message from 10.133.

2020-11-21 10:30:39 +02:00
The DNS server received a bad TCP-based DNS message from 10.133.

2020-11-21 10:30:27 +02:00
The DNS server received a bad TCP-based DNS message from 10.133.

Kuva 12. Nmap-skannauksen aiheuttamia lokeja Graylogissa.

Nmap käyttää verkon skannaamiseen muun muassa DNS-viestejä, jonka johdosta kuvassa 12 nähtävät lokit voivat äkkiseltään näyttää normaaleilta, mutta useampi yhdeltä laitteelta tuleva virheellinen DNS-viesti lyhyen ajan sisään voi viitata laitteen skannaukseen.

Nmap-skannauksen jälkeen testattiin Nessuksen sekä OpenVAS:in skannaukset. Näissä skannauksissa käytettiin kohteena lai-Win10 -tietokonetta. Kummastakin skannauksesta jäi lokitietoihin jälkiä, muun muassa epäonnistuneesta yrityksestä liittää

olematon verkkolevy. Jälleen kerran kyseinen ilmoitus voi vaikuttaa täysin normaalilta, mutta vastaavia ilmoituksia tullessa yhdeltä laitteelta useita vain muutamassa sekunnissa, voi syynä olla haavoittuvaisuuksien etsiminen kohteesta.

Nessus ei ilmoittanut löytäneensä yhtään haavoittuvaisuutta, jonka se olisi luokitellut info-tasoa korkeammalle. OpenVAS sen sijaan luokitteli yhden uhan medium-tasolle. Kyseinen uhka on ”DCE/RPC and MSRPC Services Enumeration Reporting”, jonka avulla hyökkääjä voi saada lisätietoja kohteesta. Nessus löysi myös tämän saman uhan, mutta ei luokitellut sen uhkaa info-tasoa korkeammalle. Kuvassa 13 voidaan nähdä osa OpenVASin löytämistä haavoittuvaisuuksista ja tiedoista, jotka se sai selville kohdelaitteesta.

Vulnerability		Severity ▼	QoD	Host IP
DCE/RPC and MSRPC Services Enumeration Reporting	⇆	5.0 (Medium)	80 %	10.133.10.2
DCE/RPC and MSRPC Services Enumeration	⇆	0.0 (Log)	80 %	10.133.10.2
SMB/CIFS Server Detection	?	0.0 (Log)	80 %	10.133.10.2
Microsoft Remote Desktop Protocol Detection	?	0.0 (Log)	80 %	10.133.10.2
SSL/TLS: Collect and Report Certificate Details	?	0.0 (Log)	98 %	10.133.10.2
SSL/TLS: Hostname discovery from server certificate	?	0.0 (Log)	98 %	10.133.10.2
OS Detection Consolidation and Reporting	?	0.0 (Log)	80 %	10.133.10.2
SMB Remote Version Detection	?	0.0 (Log)	80 %	10.133.10.2
Traceroute	?	0.0 (Log)	80 %	10.133.10.2
SSL/TLS: Certificate - Self-Signed Certificate Detection	?	0.0 (Log)	98 %	10.133.10.2

(Applied filter: apply_overrides=0 min_qod=70 rows=10 first=1 sort-reverse=severity)

Kuva 13. OpenVAS-skannauksen tulokset.

Ympäristön suunnittelusta johtuen, ei palomuurille kertynyt skannauksista juurikaan lokeja. Koska penetraatiotestaukseen käytettävä Kali-tietokone on samassa verkossa kuin kohdelaitteet, ei palomuuri tarkastele ja tarvittaessa estä liikennettä samalla tavalla kuin skannaavan tietokoneen ollessa verkon ulkopuolella. Jatkon kannalta olisikin hyvä siirtää penetraatiotestaukseen käytettävä laite omaan verkkoonsa, jotta voitaisiin tutkia tarkemmin penetraatiotestauksesta syntyviä lokeja, sekä palomuurin toimintaa.

8 POHDINTA

Opinnäytetyössä tutkittiin virtualisoinnin tuomia etuja erityisesti testilaboratorioiden kehittämisen kannalta sekä penetraatiotestauksen ja lokienhallinnan merkitystä osana tietoturvaa. Päätavoitteena oli suunnitella ja luoda suljettu virtuaalinen ympäristö, jossa voidaan testata erilaisia penetraatiotestaamiseen käytettäviä sovelluksia ja lokienhallintatyökaluja.

Virtualisoinnin tuoma hyöty testiympäristön luomisessa on merkittävä. Laitteiden asentaminen ja konfigurointi oli virtuaalisessa ympäristössä huomattavasti helpompaa ja nopeampaa kuin fyysisillä laitteilla. Virtualisointi mahdollistaa myös monia ominaisuuksia, kuten palautuspiteiden luomisen, jotka eivät ole edes mahdollisia fyysisillä laitteilla. Myös mahdollisuus konfiguroida laitteet etäyhteyden avulla oli suuri etu verrattuna fyysisiin laitteisiin. Virtualisointipalvelimen jo ollessa olemassa eivät laitteet vaatineet palomuurin asentamisen lisäksi lainkaan fyysistä määrittelyä.

Lokienhallinta sekä penetraatiotestaus ovat tärkeä osa tietoturvaa, ja näihin käytettävien työkalujen testaaminen suljetussa ympäristössä on jo pelkästään oppimisen kannalta tärkeää. Tämän lisäksi testiympäristö tarjoaa myös monia muita etuja, kuten mahdollisuuden uusien muutosten testaamisen ennen tuotantokäyttöä. Testiympäristössä on myös mahdollista yrittää toistaa tuotannossa tapahtuvia ongelmatilanteita, ja etsiä näihin ratkaisuja ilman, että loppukäyttäjille aiheutuu haittaa. Valitut penetraatiotestaukseen sekä lokienhallintaan käytetyt työkalut osoittautuivat toimiviksi, vaikkakin erityisesti lokienhallintatyökaluista oli välillä hankalaa löytää lähdemateriaalia ongelmatilanteissa.

Lokienhallinnan kannalta on sekä oikeanlaisten lokien kerääminen että näiden lokien totuudenmukaisuus kriittistä. Jotta lokien tarkastelusta jälkikäteen on hyötyä, tulee vähintäänkin tietää, mitä tapahtui, kenen toimesta ja milloin. Tämän saavuttamiseksi pitää lokit paitsi kerätä oikeista lähteistä, myös käsitellä ja tallentaa oikeassa muodossa, jotta niiden käsittely on ylipäätänsä mahdollista jälkikäteen. Lokien aikaleimojen täsmääminen on asia, jonka kanssa tässäkin opinnäytetyössä oli haasteita. Ympäristössä, jossa on paljon eri laitteita ja muuttujia, tulee olla tarkkana, että kaikilta laitteilta kerätyt lokit myös tallennetaan oikeassa muodossa ja oikealla aikaleimalla.

Koska luotu testausympäristö on käytännössä vain pohja tulevalle testaukselle, on ympäristöllä monia eri kehitysmahdollisuuksia tarpeesta riippuen. Penetraatiotestauksen

kannalta olisi oleellista siirtää Kali-työasema omaan verkkoonsa, jotta penetraatiotestausta voidaan kontrolloida myös palomuurin kautta. Tämän myötä myös penetraatiotestauksen vaikutusta lokeihin voitaisiin tarkastella paremmin. Lokituksen kannalta kehittämistä voisi jatkaa ylimääräisten lokien karsimisella, ja lisäämällä uusia hyödyllisiä lokilähteitä, kuten Windows Server -palvelimen DHCP-lokit. Lokien visualisointia voitaisiin myös kehittää, jos siihen nähdään tarvetta.

LÄHTEET

Cabric, M. 2015. Corporate Security Management: Challenges, Risks and Strategies. Amsterdam: Elsevier Science & Technology.

Cardwell, K. 2014. Building Virtual Pentesting Labs for Advanced Penetration Testing. Birmingham: Packt Publishing, Limited.

Elastic 2020a. Beats overview. Viitattu 30.6.2020 https://www.elastic.co/guide/en/beats/libbeat/current/beats-reference.html.

Elastic 2020b. Kibana — your window into the Elastic Stack. Viitattu 30.6.2020 https://www.elastic.co/guide/en/kibana/7.8/introduction.html.

Elastic 2020c. Logstash Introduction. Viitattu 30.6.2020 https://www.elastic.co/guide/en/logstash/7.8/introduction.html.

Elastic 2020d. The Elastic Stack. Viitattu 30.6.2020 https://www.elastic.co/elastic-stack.

Elastic 2020e. What is ElasticSearch?. Viitattu 30.6.2020 https://www.elastic.co/guide/en/elasticsearch/reference/7.8/elasticsearch-intro.html.

Fitzhugh, R. 2014. vSphere Virtual Machine Management. Birmingham, Packt Publishing, Limited.

Graylog 2020. Graylog enterprise features. Viitattu 15.7.2020 https://www.graylog.org/features.

Gregg, M. 2015. The Network Security Test Lab : A Step-By-Step Guide. Hoboken: John Wiley & Sons, Incorporated.

Greenbone n.d.a. OpenVAS – Open Vulnerability Assessment Scanner. Viitattu 28.7.2020 https://www.openvas.org/#about.

Kali 2020a. Kali Linux Tools Listing. Viitattu 28.7.2020 https://tools.kali.org/tools-listing.

Kali 2020b. Nmap Package Description. Viitattu 8.7.2020 https://tools.kali.org/information-gathering/nmap.

Kali 2020c. Should I Use Kali Linux?. Viitattu 28.7.2020 https://www.kali.org/docs/introduction/should-i-use-kali-linux/.

Kali 2020d. What is Kali Linux?. Viitattu 28.7.2020 https://www.kali.org/docs/introduction/what-is-kali-linux/.

Kumar, H. 2013. Learning Nessus for Penetration Testing. Birmingham: Packt Publishing, Limited.

Kyberturvallisuuskeskus 2020. Näin keräät ja käytät lokitietoja. Viitattu 8.11.2020 https://www.kyberturvallisuuskeskus.fi/fi/ajankohtaista/ohjeet-ja-oppaat/nain-keraat-ja-kaytat-lokitietoja.

Kyberturvallisuuskeskus 2020. Tietoturva. Viitattu 8.11.2020 https://www.kyberturvallisuuskeskus.fi/fi/toimintamme/saantely-ja-valvonta/tietoturva.

NIST. 2006. Guide to Computer Security Log Management, NIST Special Publication 800-92. Viirattu 14.11.2020 https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-92.pdf.

Nmap n.d.a. Introduction. Viitattu 8.7.2020 https://nmap.org/.

NXLog 2020. About NXLog. Viitattu 12.5.2020 https://nxlog.co/documentation/nxlog-user-guide/about-nxlog.html.

Oriyano, S. 2016. Penetration Testing Essentials. Hoboken: John Wiley & Sons, Incorporated.

Rousku, K. 2014. Kyberturvaopas – Tietoturvaa kotona ja työpaikalla. Helsinki: Talentum Media Oy.

Tenable 2020. The Nessus Family. Viitattu 19.5.2020 https://www.tenable.com/products/nessus.

Kusek, C.; Leibowitz, M. & Spies, R. 2014. VMware vSphere Performance: Designing CPU, Memory, Storage, and Networking for Performance-Intensive Workloads. Hoboken: John Wiley & Sons, Incorporated.

Weidman, G. 2014. Penetration Testing : A Hands-On Introduction to Hacking. San Francisco: No Starch Press, Incorporated.