Niina Rantanen

Päätelaitteiden testauksen raportointijärjestelmä pilvialustalla

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Mobiilisovellukset tutkinto-ohjelma

Insinöörityö

21.5.2017

Metropolia

Tekijä Otsikko Sivumäärä Aika	Niina Rantanen Päätelaitteiden testauksen raportointijärjestelmä pilvialustalla 57 sivua + 3 liitettä 21.5.2017
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintätekniikka
Pääaine	Mobiilisovellukset
Ohjaaja	Yliopettaja Kari Salo

Teleoperaattorille palvelujen ja laitteiden testaus on olennainen osa laadunvalvontaa. Insinöörityön tarkoituksena oli parantaa päätelaitetestien raportointia. Päätelaitetesteissä havainnoidaan päätelaitteiden suorituskykyä, sisäistä toimintaa sekä laitteen ja verkon välisiä asioita. Insinöörityön toimeksiantajana oli Suomessa toimiva teleoperaattori.

Insinöörityön tavoitteena oli verkkoyhteydellä varustettujen päätelaiteiden testitulosten asettaminen automaattisesti käyttäjien saataville niin, että data on ymmärrettävässä graafisessa muodossa ja helposti saatavilla. Ennen raportointijärjestelmän rakentamista testaajat keräsivät testitulokset raporttimuotoon manuaalisesti raakadatasta, testiraporteille ei ollut keskitettyä säilytyspaikkaa ja testituloksien saaminen visuaaliseen muotoon oli työlästä.

Raportointijärjestelmän pääkomponentteina toimivat datan käsittelyä tarjoava lokienhallinta-ohjelmisto sekä julkisen pilven palvelu, joka tarjoaa konesalikapasiteettia. Insinöörityön keskeisenä tavoitteena oli järjestelmän alustana toimivan konesalin suunnittelu ja toteutus pilveen, mikä vaati saumatonta integraatiota sovellusten kesken.

Insinöörityön tuloksena syntyi testitulosten raportointijärjestelmä, joka käyttää skaalautuvaa pilvialustaa. Raportointijärjestelmä tallentaa ja tuottaa käyttäjille automaattisesti visuaalisia testiraportteja keskitetyssä paikassa. Raportointijärjestelmän automaatio saatiin nostettua halutulle tasolle ja testitulokset kulkeutuvat testiajojen jälkeen automaattisesti parsittaviksi ja visuaalisiksi raporteiksi verkkokäyttöliittymään, johon tehtiin kahdet eritasoiset käyttäjätunnukset. Raporteista on selkeästi havaittavissa testiajojen tulokset ja mahdolliset epäkohdat.

Avainsanat	Splunk, eSali, raportointi, IaaS-palvelu, datan visualisointi, ELK

Author	Niina Rantanen
Title	Test log management system of user equipment on a cloud platform
Number of Pages	57 pages + 3 appendices
Date	22 May 2017
Degree	Bachelor of Engineering
Degree Programme	Information and Communication Technology
Major	Mobile Solutions
Instructor	Kari Salo, Principal Lecturer

This engineering thesis was commissioned by the Finnish network operator. For a network operator, it is an essential part of quality assurance to continuously test any user equipment, particularly before releasing any new products and services to the consumers. This study examines the test log management system for equipment such as mobile phones and modems with the main testing areas which are performance, device features and various network functionalities.

The objective of this thesis was to plan, build and implement a centralized test log management system on a scalable cloud platform. Before the implementation of the test log system implementation that is described in this study, a data collection and a results processing were done manually by the test engineers. Test reports had to be compiled manually and there was no common place to store them nor was it easy to create visual graphs or charts making the whole test reporting process cumbersome.

The test log management system implemented in this study is using two major components: a software for data compliance and a public cloud solution. One of the key discussion points of this thesis was the design and implementation a virtual data center that hosted the entire system. The theory part of the study discussed the utilized cloud platform service features and service model. Study also introduced Splunk software architecture and functionalities while compared to a similar open source product ELK (Elasticsearch, Logstash and Kibana).

The study enabled the implementation of the automated log management system that collects and compiles test data from various different sources, creates visual reports on the go and archives them for easy user access.

Keywords	Cloud Computing, Splunk, ELK, eSali, test report automation

Sisällys

Liitteet

Liite 1. Splunk-järjestelmävaatimukset

Liite 2. Palomuurisäännöt: liikenne ulospäin

Liite 3. Näytökuvat Kibanasta

1 Johdanto

Insinöörityön toimeksiantaja on Elisa Oyj. Elisa on tietoliikenne-, ICT- ja online-palveluyritys, jonka asiakkaana on 2,3 miljoonaa kuluttajaa, yritystä ja julkishallinnon organisaatiota. Elisa tuotanto vastaa muun muassa valittujen Elisan palveluiden ja asiakaspäälaitteiden testaamisesta sekä testiautomaation, testausprosessien ja menetelmien kehittämisestä.

Asiakaspäätelaitteilla tarkoitetaan kiinteän verkon ja mobiilipuolen päätelaitteita, joille yhteinen tekijä on verkkoyhteys. Näitä laitteita testataan tuotannon osalta viikoittain. Päätelaitetestauksen tavoitteena on varmistaa hankittavien laitteiden toimivuus ja soveltuvuus Elisan verkkoympäristöön sekä varmistaa laitteen vastaavan ennalta määriteltyjä teknisiä ja toiminnallisia vaatimuksia. Laitetoimittajat vastaavat omista perustesteistä, ennen kuin laite toimitetaan Elisalle.

Elisan päätelaitetesteissä tarkkaillaan laitteen suorituskykyä, sisäistä toimintaa ja laitteen ja verkon välisiä asioita. Näiden testien perusteella päätetään, onko laite hyväksyttävissä ja käyttöönotettavissa. Tällainen testaus on edellytyksenä laadukkaalle tuotteelle ja näkyy suoraan asiakastyytyväisyytenä ja kustannussäästöinä.

Insinöörityön tarkoituksena on päätelaitteiden testitulosten raportoinnin kehitys. Raportointi on tärkeä osa koko testausprosessia. Sen avulla voidaan helpottaa ja nopeuttaa päätöstä laitteiden käyttöönotosta. Insinöörityössä käytettävät päätelaitteet rajautuvat mobiilipuolen päätelaitteisiin eli matkapuhelimiin sekä Tampereella rakenteilla olevan WLAN-laboratorion testattaviin langattoman verkon laitteisiin.

Tavoitteena on rakentaa reaaliaikainen raportointijärjestelmä yhtenäiseen paikkaan, jonne päätelaitetestauksen testitulokset kulkeutuvat automaattisesti testiajojen jälkeen. Testitulokset halutaan näyttää käyttäjälle visuaalisina, selkeinä ja informatiivisina raportteina. Järjestelmän rakentamisen pääkomponentit ovat lokienhallintaan tarkoitettu ohjelmisto Splunk Enterprise ja Elisan oma kaupallinen tuote Elisa eSali, joka mahdollistaa virtuaalisen konesalin rakentamisen. Keskeisin teema insinöörityössä on yhtenäisen virtuaalisen konesalin suunnittelu ja toteutus raportointijärjestelmää varten. Keskiössä virtuaalisalin toteutuksessa on toimivalla tavalla muodostettu verkkoympäristö, joka mahdollistaa eri sovellusten integroinnin.

Järjestelmä tarvitsee siis myös joukon muita sovelluksia toimiakseen automaattisena raportointijärjestelmänä. Nämä sovellukset esitellään insinöörityössä teoriatasolla peilaten omaan tekemiseen. Erityisesti keskitytään kuitenkin Splunk-ohjelman arkkitehtuurin, komponenttien ja konfiguroinnin esittämiseen. Splunkilla on rakennettu jo ennen insinöörityön aloittamista raportointijärjestelmän front-end-puoli, joka liitetään uuteen valmiiseen ympäristöön. Insinöörityössä esitetään front-end-puolen olennaiset rakennusosat ja avataan datan visualisointia teoriatasolla keskittyen visualisointiprosessin lopputulokseen. Lopuksi työssä otetaan Splunkin kanssa vertailuun samankaltainen avoimen lähdekoodin ohjelmakokonaisuus ELK (Elasticsearch, Logstash ja Kibana).

2 Raportointijärjestelmän vaatimusmäärittely

Seuraavissa alaluvuissa käydään läpi insinöörityönä tehtävän raportointijärjestelmän vaatimusmäärittely, jossa haetaan vastausta kysymykseen, mitä raportointijärjestelmältä odotetaan. Tähän sisältyvät järjestelmän toiminnallisuus, tekniset reunaehdot ja laadulliset seikat. Tätä ennen kuitenkin kuvataan raportoinnin nykytila ja määritellään ratkaistavat ongelmat. Nykytilan kuvauksessa keskitytään mobiilipuolen päätelaitteiden raportoinnin kuvaamiseen, sillä koska WLAN-laboratorio on rakenteilla, rakennetaan sen raportointijärjestelmä insinöörityön yhteydessä. Vaatimusmäärittelyssä ei mennä liian yksityiskohtaisiin kuvauksiin, eikä siihen sisällytetä vaatimusmäärittelyä siitä, mitä laitteita testaan ja mitä tuotteista halutaan testata. Raportointijärjestelmästä halutaan antaa vaatimusmäärittelyn kautta yleisluontoinen kuva.

2.1 Nykytila ja ratkaistavat ongelmat

Päätelaitteiden testausympäristöjen moninaisuuden vuoksi myös se vaihtelee, missä muodossa testitulokset näkyvät käyttäjälle. Osassa ympäristöissä testausohjelmat antavat suoritetuista testeistä yleiskuvan valmiina raporttimuodossa, jossa näkyvät aikaleimat sekä minimissään tieto siitä, kuinka moni testeistä on mennyt läpi onnistuneesti. Joskus tarvitaan kuitenkin paljon yksityiskohtaisempaa tietoa ja raportti pitää itse generoida halutunlaiseksi raakadatasta, kuten mobiilipuolen päätelaitetestauksessa. Raportointi tehdään mobiilipuolen päätelaitetestauksessa tällä hetkellä manuaalisesti, mikä on testaajille aikaa vievää.

Toinen tärkeä tekijä testiraporteissa on yhtenäinen säilytyspaikka, josta tulokset ovat helposti tarkasteltavissa. Tämä lisää koko testausprosessin näkyvyyttä yrityksen sisällä. Koska raportit ovat yhtenäisessä paikassa, saadaan myös paremmin mahdollistettua vertailukelpoisuus eri laitteiden välillä. Mobiilipuolen testiraporteille ei ole tällä hetkellä sopivaa yhtenäistä säilytyspaikkaa, joka tarjoaisi asianomaisille testiraporttien tarkastelun, silloin kun he sitä tarvitsevat.

Raportointijärjestelmällä on kaksi käyttäjäryhmää: päätelaitteiden testaajat ja hankintapuolen henkilöt. Näille käyttäjäryhmille tarvitaan molemmille omat näkymät testituloksista. Testaajat tarvitsevat yksityiskohtaisempaa tietoa siitä, miten laite on käyttäytynyt testin aikana, kun taas ostopuolen henkilöille riittää yleiskuvaus laitteen suoriutumisesta.

Näille käyttäjäryhmille annetaan oma rooli raportointijärjestelmään, ja näin rajataan oikeuksia tehdä ja nähdä asioita. Testaajilla on tunnukset, joilla pystyy hallinnoimaan järjestelmää, pääsee käsiksi raakadataan ja pystyy tekemään muutoksia testitulosten näkymään ja järjestelmän asetuksiin. Hankintapuolen henkilöiden tunnukset rajataan testiraporttien katseluoikeuteen.

Raportointijärjestelmän käyttötapaukset keskittyvät raporttien tuottamiseen ja katselemiseen testatuista päätelaitteista, jotka jakautuvat mobiilipäätelaitteisiin ja WLAN-laboratoriossa testattaviin langattoman verkon laitteisiin. Raportointijärjestelmän halutaan parsivan (engl. parsing, datan pilkkominen pienempiin palasiin seuraamalla sääntöjä, jotta dataa on helpompi hallita, tulkita ja välittää eteenpäin) tulokset ajetuista testeistä ja esittävän ne selkeästi kuvaajien ja lukujen avulla niin, että epäkohdat, ei-toimivat ominaisuudet ja laitteen heikko suorituskyky ovat helposti havaittavissa. Tämän jälkeen on helpompi tehdä päätös siitä, onko laite käyttöönotettavissa.

2.2 Järjestelmän infrastruktuuri

2.2.1 Ympäristö

Raportointijärjestelmän ympäristö rakennetaan julkiseen pilveen, sillä se tarjoaa joukon muunneltavia resursseja, kuten tietoliikenneverkot, palvelimet, tallennustilan, sovellukset ja erilaiset palvelut, joita voidaan kätevästi asentaa ja myös helposti poistaa

käytöstä. Toimintaympäristöksi haluttiin myös paikka, johon on helppo pääsy erilaisilla päätelaitteilla milloin tahansa ja lähes kaikkialta.

Pilvipalvelutoimittajia on paljon, mutta ympäristön tarjoajaksi valittiin yrityksen oma kaupallinen tuote Elisa eSali, sillä sen käyttöönotto onnistui vaivattomasti. Elisa eSali tarjoaa mahdollisuuden rakentaa oma virtuaalinen konesali. Taulukossa 1 esitetään virtuaalisen konesalin kapasiteettivaatimukset.

Taulukko 1. eSalin minimikapasiteettivaatimus virtuaalisen konesalin perustamiselle [1].

Konesalin ominaisuudet	**Minimikapasiteetti**
Keskusmuisti	1 GB
Levytila	10 GB
Suoritin	1 kpl

2.2.2 Ydinsovellus

Järjestelmän ydinsovelluksen tehtävä on mahdollistaa testituloksien automaattinen analysointi ja esitys. Tähän valittiin kaupallinen ohjelmisto nimeltä Splunk Enterprise. Tuotteeseen päädyttiin sen monipuolisuuden ja näyttävien visualisointiominaisuuksien vuoksi. Talon sisällä oli myös muita tämän ohjelman käyttäjiä, joilta saatiin kuulla ohjelman laajoista käyttömahdollisuuksista.

Splunk pähkinänkuoressa

Splunk on lokienhallintatyökalu, joka tallentaa, indeksoi ja analysoi konetietoa, jota se voi vastaanottaa lukemattomista lähteistä, kuten verkkolaitteilta, palvelimilta, sensoreilta, ohjelmilta ja verkkosivuilta. Tämän jälkeen työkalun avulla voidaan luoda raportteja, käyttöliittymiä ja erilaisia graafisia esityksiä datasta. [2.]

Splunk lukee raakadataa, jota monet järjestelmät kirjoittavat jatkuvasti. Raakadata jaetaan määriteltyihin tapahtumiin, joiden pohjalta Splunkin hakujärjestelmä toimii. Data käy Splunkin sisällä läpi useita eri vaiheita muuntuessaan alkuperäisesti lähteestä lopulliseen tapahtumamuotoon, jota pystytään hakemaan ja analysoimaan. [2.]

Data kirjataan kiintolevylle valittuun indeksiin, joka toimii datan varastona. Paketoitu ja indeksoitu data on noin 10 % alkuperäisen datan koosta. Lopulta tietoa hallitaan Splunkin tietohaun avulla, jossa hakukyselyt kirjoitetaan Splunkin omalla SPL-kielellä (engl. search processing language). Nämä haut tallennetaan raporteiksi ja liitetään dashboard-näkymään paneeleina, johon käyttäjät pääsevät käsiksi käyttöliittymän kautta. [2.]

Järjestelmävaatimukset

Splunk tukee useita eri Windows- ja Unix-pohjaisia käyttöjärjestelmiä. Arkkitehtuurituki vaihtelee kuitenkin eri käyttöjärjestelmien välillä. Tuetut järjestelmät ovat tarkemmin listattuna liitteessä 1. Vaatimuksena kaikissa käyttöjärjestelmissä on 64-bittinen versio.

Splunkin käyttöliittymä toimii seuraavissa selaimissa:

- Firefox (uusin)
- Internet Explorer 11
- Safari (uusin)
- Chrome (uusin). [3.]

2.3 Koventaminen

Koventaminen halutaan ottaa Splunkissa huomioon rajaamalla käyttäjätunnukset kahteen eri ryhmään. Järjestelmän ylläpitäjän tunnuksille annetaan oikeus tehdä muutoksia järjestelmään, ja muut tunnukset luodaan katselua varten.

Alustana toimivalla virtuaalisalilla on samat tietoturvauhat kuin fyysiselläkin palvelimella. Turhien ohjelmien ja palveluiden poisto halutaan ottaa huomioon virtuaalisalin virtuaalikoneiden luonnissa. Splunk-ohjelmaa suorittava virtuaalikone otetaan käyttöön palvelimena ilman työpöytää, eikä palvelimelle asenneta mitään muuta verkkopalvelua Splunkin lisäksi. Kattava palomuuri on myös vaatimuksena virtuaaliympäristön toiminnalle.

Koventamista on mietittävä myös käyttöoikeuksien näkökulmasta: kuka saa tehdä mitä ja missä [4]. Pääkäyttäjänä ei saa pystyä kirjautumaan Splunkin palvelimelle SSH:n (engl. Secure Shell, etäkäyttöohjelmisto, jolla voidaan ottaa salattuja yhteyksiä järjestelmästä toiseen) kautta. Sudo-komentoa käytetään täyttämään root-tason

komennot. Sudoa käyttämällä lisätään järjestelmän turvallisuutta ilman, että jaetaan root-salasanaa käyttäjille. Splunk-palvelimelle luodaan aluksi yksi käyttäjätunnus, jota käytetään kehitystyöhön.

Tiedonsiirtomenetelmänä käytetään FTP:n (engl. File Transfer Protocol, Tiedostojen siirtämiseen suunniteltu protokolla) rinnalla myös SFTP:tä (engl. SSH File Transfer Protocol), jossa koko sessio on salattu, mikä tarkoittaa, että yhtään salasanaa ei lähde ilman salausta [4]. FTP-käyttöön luodaan myös omat käyttäjätunnukset, joilla pääsy rajoitetaan vain tiettyyn käytettyyn hakemistoon.

2.4 Suorituskyky

Järjestelmän suorituskyky riippuu yksittäisten komponenttien suorituskyvystä. Splunk asettaa omat minimivaatimukset toimivalla suorituskyvylle, samoin kuin eSali. Toimivan suorituskyvyn mittareina toimivat vastausaika, saatavuus, välityskyky ja resurssien käyttöaste.

Splunkin suorituskykyvaatimuksiin vaikuttaa se, miten paljon dataa indeksoidaan päivätasolla ja miten monta käyttäjää ohjelmalla on. Raportointijärjestelmän indeksoima data on reilusti alle 1 gigatavua päivätasolla, ja käyttäjätunnuksia on kaksi, joten yksi Splunk-installaatio, joka yhdistää eri komponentit, on riittävä takaamaan hyvän suorituskyvyn [5]. Lisää Splunk-komponenttien toiminnasta on luvussa 5.3.2. Taulukossa 2 esitetään Splunkin minimivaatimukset virtuaalikoneelle.

Taulukko 2. Splunkin minimivaatimukset virtuaalikoneelle [6].

Virtuaalikoneen ominaisuudet	**Minimivaatimus**
Käyttöjärjestelmä	Linux, kernel versio 2.6 tai myöhempi
Suoritin	12 vCPU
Keskusmuisti	12 GB

eSalin hyvään suorituskykyyn vaikuttaa muun muassa WMware-tools -ohjelmiston asennus ja ajan tasalla pitäminen. Tämä ohjelmisto sisältää virtuaalikoneen toiminnan kannalta tärkeitä ajuri- ja ohjelmistopäivityksiä, jotka parantavat suorituskykyä [7].

Asennuksista ja päivittämisestä on itse huolehdittava, mutta palveluntarjoaja muistuttaa automaattisilla viesteillä, jos niissä on puutteita. eSalin suorituskykyyn vaikuttaa myös riittävä resurssien varaaminen. Erityisesti muistin (RAM) ja suorittimien (CPU) määrä vaikuttaa merkittävästi.

3 Raportointijärjestelmän datalähteet

Splunk-ohjelman käyttöön voidaan kuljettaa dataa kolmella eri tavalla: manuaalisesti lataamalla, monitoroimalla tiettyä kohdetta tai käyttämällä Splunkin omaa Forwarder-komponenttia (ns. aineistovälittäjä), joka toimii datalähteen ja Splunk-palvelimen rajapintana [8]. Forwarderin toimintaan perehdytään paremmin luvussa 5.3.2. Raportointijärjestelmän vaatimus toimia automaattisena testitulosten esittäjänä poistaa manuaalisen latauksen käytettävistä vaihtoehdoista.

Data voi siis olla valmiina samassa paikassa Splunk-palvelimen kanssa tai se voidaan kuljettaa eri paikasta ohjelman käyttöön. Esittelen seuraavaksi raportointijärjestelmän käyttämät datalähteet ja sen, miten ja missä muodossa data kulkeutuu indeksoitavaksi.

3.1 viSer-testausohjelma

SmartViser on ranskalainen startup-yritys, jonka tuotetta viSeria käytetään mobiililaitteiden testaamiseen. viSer on puhelimeen asennettava ohjelma, joka suorittaa automaattisesti testejä liittyen puhelimen ja verkon välisiin asioihin ja testaa laitteen sisäistä toimintaa ja suorituskykyä erilaisin käyttäjän toiminnoin. [9.]

viSer kirjoittaa omaa lokia suoritetuista testeistä CSV-, XML- ja JSON-muodossa ja tallentaa ne oletuksena puhelimen muistiin. Näistä tiedostoista käytetään kattavaa CSV (engl. Comma Seperated Value) -tiedostoa, joka sisältää kaikki tallennetut tiedot testin kulusta. Tiedoston jokainen rivi sisältää aikaleiman ja statuksen siitä, minkälainen tallennus on kyseessä. Tallennus voi olla perustietoa laitteesta, järjestelmän ja ympäristön käyttäytymisestä tai tietyn käyttäjätilanteen käyttötapaus, joka antaa Pass- tai Fail-tuloksen.

CSV-tiedoston nimeämisen suhteen noudatetaan kaavaa <Aikaleima>_<Laitteen nimi>. Se oli helposti valittavissa viSerin käyttöliittymän asetuksista, mikä mahdollistaa automaattisen testitulosten nimeämisen. Nimeämisellä on tärkeä rooli testitulosten esittämisessä, sillä tulostiedoston kulkeutuessa Splunkiin indeksoitavaksi parsitaan nimestä tietoa säännöllisen lausekkeen avulla.

Testitulosten siirtäminen puhelimesta Splunk-palvelimelle onnistuisi hyvin Splunkin tarjoamilla laajennussovelluksilla. Testattavat puhelimet kuitenkin vaihtuvat tiuhaan, ja tämä vaatisi jokaisen testattavan puhelimen konfiguroinnin datan lähettäjäksi. Helpompi vaihtoehto on käyttää viSerin tarjoamaa tulosten FTP-lähetystä. viSer tarjoaa mahdollisuuden lähettää tulokset automaattisesti eteenpäin FTP-tiedostonsiirtomenetelmällä käyttäen TCP-protokollaa (engl. Transmission Control Protocol, Protokollan avulla tietokoneet voivat lähettää toisilleen tavujonoja luotettavasti). FTP-tiedostosiirtoa varten viSerin asetuksiin tarvitaan kohdepalvelimen osoite, kohdekansio sekä käyttäjätunnus ja salasana.

Spunk-palvelimen käytössä olevaan virtuaalikoneeseen asennetaan FTP-palvelin, johon testitulokset lähetetään ZIP-paketissa. Tämän FTP-palvelimen konfiguroinnista on enemmän tietoa luvussa 5.3. Splunk määritellään monitoroimaan hakemistoa, jonne tiedostopaketit ohjataan. Aina uuden tiedoston saapuessa Splunk lataa sen valittuun indeksiin eli tiedostomuotoiseen tietokantaan.

viSerin lähettämä ZIP-paketti sisältää kaikki viSerin tuottamat tulostiedostot. Raportointijärjestelmä käyttää vain CSV-tiedostoa, jolloin muut tiedostot tulevat turhaan monitoroituun hakemistoon. Tähän ei kuitenkaan pysty viSerin puolelta vaikuttamaan. Splunk tarjoaa mahdollisuuden poimia halutut tiedostot monitoroidusta paikasta käyttämällä säännöllistä lauseketta käyttävää whitelist- ja blacklist-toimintoa. Tämä ei kuitenkaan päde pakattujen tiedoston sisältöön, sillä kaikki pakatut tiedostot monitoroidaan yhtenäisenä kokonaisuutena ennen purkamista ja indeksointia. Datan määrä on kuitenkin niin vähäistä, että koko ZIP-paketti voidaan huoletta lähettää indeksoitavaksi ilman pelkoa järjestelmän turhasta kuormittumisesta.

3.2 Testilaboratorio

Elisan Helsingin Talin toimistotilojen kellariin on rakennettu testilaboratorio (ns. EMP-koppi). Laboratorioon on rakennettu minikoossa koko Elisan radioverkko niin, että kaikkiin verkkoihin pystyy siirtymään kopista käsin. Laboratorio on rakennettu tekniikan ehdoilla: ulkoseinät ovat päällystetty teräksellä, joka estää radioaaltojen liikkumisen kopin ja ulkomaailman välillä, sisäseinillä on paksu vaahtomuovikerros, joka imee itseensä harhailevat radioaallot.

Laboratoriossa testataan hallitusti niitä matkapuhelinverkon olosuhteita, joihin käyttäjä liikkuessaan joutuu. Laboratoriossa on myös vaimennin, jolla voi simuloida radioverkon eri tilanvaihtoja. viSerin ja vaimentimen käyttö on haluttu kopissa yhdistää ja automatisoida, joten koppiin on rakennettu kokonainen automaatioympäristö, joka käyttää lukuisia työkaluja ja komponentteja. Tällä testausympäristöllä halutaan testata laitteen solunvaihdot verkossa, signaalitaso, datanopeudet sekä datayhteyksien toimiminen.

Laboratorion testiympäristössä suoritetaan samaa viSer-ohjelmaa puhelimessa ja testiympäristö koostuu useasta integroidusta komponentista. Tuloslähteenä käytetään samaa viSerin tuottamaa CSV-tiedostoa. Testiympäristön testeissä ollaan myös kiinnostuneita vaimentimen käyttämästä signaalitasosta testin aikana. Testiympäristö luo tätä varten myös toisen CSV-tiedoston nimeltä ”stand_alone.csv”, jota haluttiin hyödyntää raportoinnissa.

Testiympäristössä testitulokset tallentuvat testilaboratorion tietokoneen paikalliseen hakemistoon. Testitulokset tallentuvat aina yhtenäiseen paikkaan, joten voidaan hyödyntää Splunk Forwarder-komponenttia siirtämään data Splunk-palvelimelle. Testilaboratorion pöytäkoneeseen asennetaan Splunk Forwarder, joka kuljettaa tiedostot TCP:tä käyttäen Splunkin omalla kuljetusprotokollalla.

3.3 WLAN-laboratorio

WLAN-laboratorio on Elisan Tampereelle rakentama testiympäristö, jossa tutkitaan langattoman verkon laitteita. Ympäristössä on tarkoitus tutkia, kuinka ne toimivat

kodinomaisessa, häiriöttömissä ympäristöissä sekä häiriöllisissä ympäristöissä. Pääpaino on asioissa, jotka jokainen kuluttaja voi huomata laitetta käyttäessä. Tarkoitus on asettaa laitteita myös eri ominaisuuksien osalta järjestykseen.

Testien tuloksen tulevat JSON- (JavaScript Object Notation) ja CSV-muodossa. Lokit sisältävät muun muassa tuloksia iperf-ohjelmalla suoritetuista verkon kuormitustesteistä, joista nähdään vasteaikaa, pakettihäviötä (engl. packet loss), huojuntaa (engl. jitter) ja siirtonopeutta (engl. throughput). Näissä testimittauksissa käytetään yhteydellistä TCP-liikennettä ja yhteydetöntä UDP-liikennettä eri WLAN-kanavilla molemmista suunnista.

Lokien nimeämiseen piti kiinnittää tarkkaa huomiota, sillä järjestelmä poimii nimestä yksityiskohtaista tietoa testatusta laitteesta, suoritetun testisetin nimen ja aikaleiman, päiväkohtaisen ajon juoksevan järjestysnumeron sekä tietoa testissä käytetyistä parametreista. Esimerkki yhden ajetun testintuloksen nimestä:

Zyxel__VMG3925__Elisa6__2016-11-08__00_General_Robot__003__iperf__raspi1-desktop__iperf__tcp__downlink__2GHz.json.

Lokit siirtyvät Splunkin käyttöön Forwarderilla, joka on asennettu koneeseen, johon testitulokset kerätään. Testitulokset tallentuvat paikallisesti sille määriteltyyn polkuun, josta Splunk Forwarder kuljettaa ne eteenpäin indeksoitavaksi ja parsittavaksi.

4 Elisan eSali-pilvipalvelu

Elisa eSali on suomalainen palveluntarjoaja, joka tarjoaa konesalikapasiteettia skaalautuvasta julkisesta ja yksityisestä pilvestä. Tuote käyttää palvelun pohjana VMware vCloud-tekniikkaa. Palvelun hallintaan käytetään eCloud-hallintaportaalia, jossa luodaan virtuaalipalvelimet ja verkot. Virtuaalipalvelimia hallitaan VMwaren vCloud Directorilla sekä API-rajapinnan kautta. [1.]

4.1 Pilvipalvelun palvelu- ja käyttöönottomalli

Pilvipalvelut määritellään yksinkertaisesti palveluntarjoajan dynaamisesti verkon välityksellä tarjoamista IT-resursseista, kuten ohjelmistoista, laitteistoista tai palveluista.

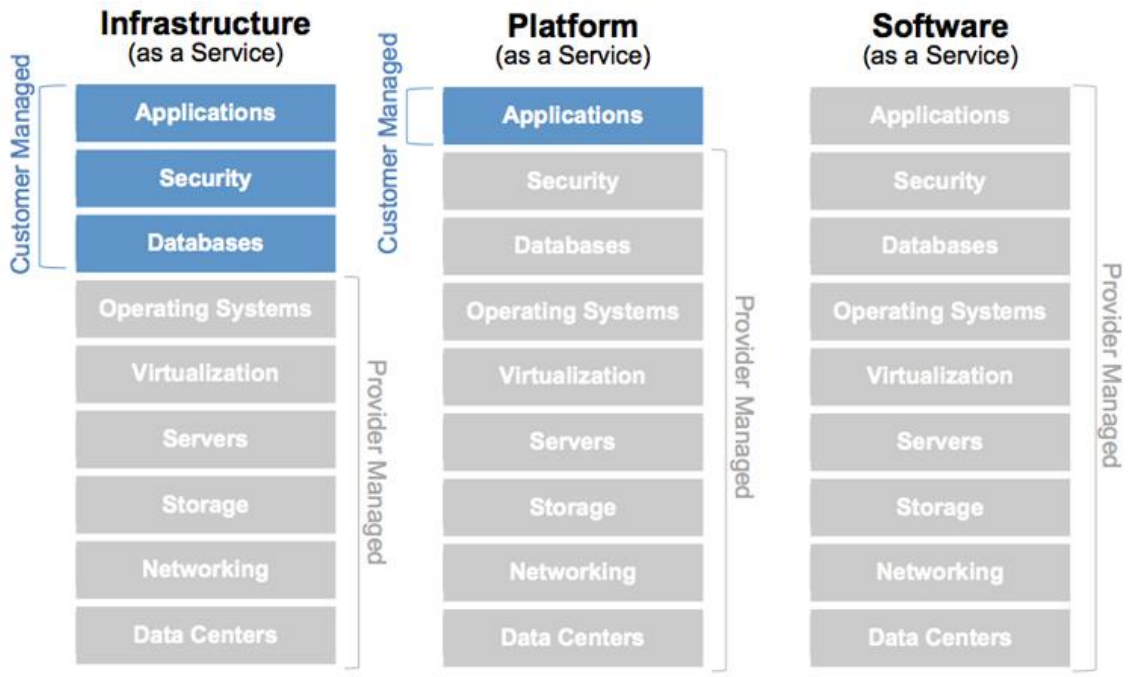
Käyttäjä tarvitsee siis minimissään toimivan verkkoyhteyden ja selaimen palvelujen käyttöön. [10, s.16.]

Pilvipalveluilla ei kuitenkaan tarkoiteta aina ulkoisen palveluntarjoajan tuottamia palveluja vaan myös yrityksen omaa pilvitoimintaa, jossa yritys on itse tuottaja ja käyttäjä. Tällöin puhutaan yksityisestä pilvestä. Pilvipalvelut jaotellaan siis käyttöönottomallin mukaan, joka kuvaa, kuka palvelun tarjoaa ja kenelle se on suunnattu. Nämä käyttöönottomallit jaetaan kaiken kaikkiaan kolmeen luokkaan: yksityiseen pilveen, julkiseen pilveen ja hybridipilveen [10, s. 32]. Elisa eSali tarjoaa palvelua julkiseen pilveen, mikä tarkoittaa laskentakapasiteetin tarjoamista julkisen internetin välityksellä. Elisa eSali mahdollistaa myös julkisen pilven liittämisen yrityksen omaan yksityiseen pilveen eli hybridi-vaihtoehdon [11].

Toinen jaottelu tehdään palvelumallin mukaan, jolla määritellään, mitä palveluarkkitehtuurin kerrosta palveluntarjoaja tarjoaa. Pilvipalvelut jaetaan yleisesti kolmeen palvelumalliin:

- IaaS – infrastruktuuri palveluna
- PaaS – sovellusalusta palveluna
- SaaS – sovellukset palveluna [10, s. 22].

Tämä jako tehdään käyttäjän ja palvelutarjoajan vastuualueiden mukaan, joista on havainnollistus kuvassa 1. Kuten kuvasta näkyy, käyttäjän vastuu kasvaa siirryttäessä SaaS-mallista kohti IaaS-mallia.



Kuva 1. Vastuunjako SaaS-, PaaS- ja IaaS-malleissa [12].

Elisa eSali -palvelu vastaa täysin IaaS-palvelutasoa. IaaS tarkoittaa käytännössä palveluntarjoajalta vuokrattavaa palvelintilaa, joka sijaitsee palveluntarjoajan konesaleissa. Palveluntarjoaja ylläpitää konesalia, josta se lohkoo etukäteen räätälöityjä ja hinnoiteltuja osioita asiakkaan käyttöön.

IaaS-mallissa palveluntuottajan vastuu ulottuu vain alustoihin, joita käytetään kapasiteetin tuottamiseen. Palvelun käyttäjän vastuulle jäävät siis kaikki palvelimet, konfiguraatiot ja hallinnointi [10, s. 25]. Tämä vastuujako mahdollistaa palvelun helpon räätälöinnin mieleiseksi. Elisa eSalia hallinnoidaan täysin web-pohjaisen graafisen käyttöliittymän kautta, jonka avulla on mahdollista tilata ja toteuttaa infrastruktuurimuutoksia helposti [1]. Tämä liikkumavapaus ja käyttäjän kontrolli on eduksi raportointijärjestelmälle ajatellen tulevaisuuden laajennusmahdollisuuksia. Edellä mainittujen IaaS:n tunnuspiirteiden mukana tulee kuitenkin myös suurempi vastuu järjestelmän toimimisesta.

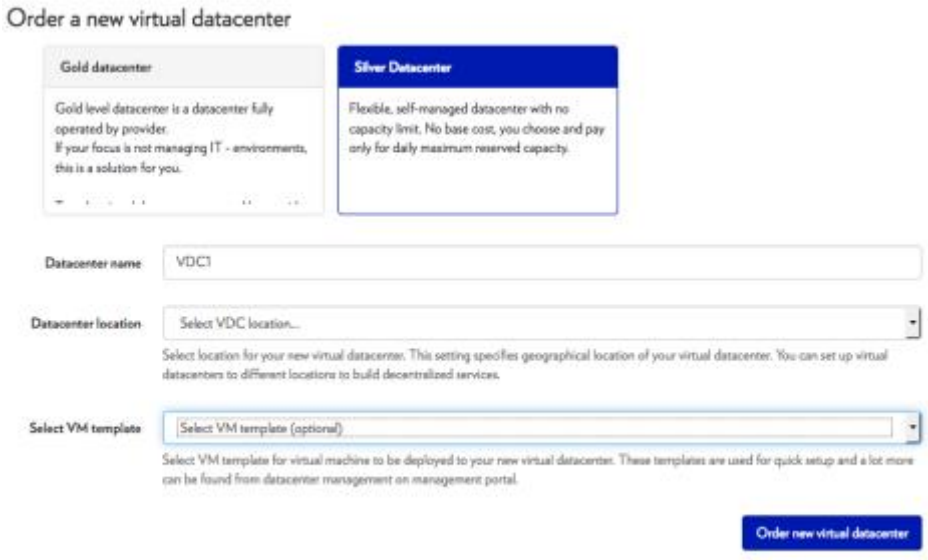
4.3 Pilviympäristön rakentaminen

4.3.1 Käyttöönotto

eSali-tuote tilattiin Elisa Appelsiiniilta, verkkosivulta www.eSali.fi. Rekisteröitymisen jälkeen pääkäyttäjä sai tunnukset palveluun ja sen käyttö aloitettiin 30 päivän demolla, jolla oli mahdollista tutustua palveluun. Demoajan umpeuduttua siirryttiin eSalin tarjoamaan hopeatason palveluun luomaan varsinaista konesalikokonaisuutta tuotantokapasiteetilla.

Palvelun itsehallinta tapahtuu hallintaportaalin admin.ecloud.fi kautta. Palvelussa käytetään vahvaa tunnistautumista, mikä tarkoittaa mobiilivarmenteen käyttöä: joka kerta kirjautumisen yhteydessä on kirjoitettava puhelimeen saapuva kertakäyttöinen salasana varmenteeksi.

Varsinaisen konesalin tilaus onnistuu helposti hallintaportaalin etusivulta: Order new virtual datacenter. Tilauslomakkeessa nimetään konesali, valitaan konesalin fyysinen osoite ja määritellään kapasiteetti, joka käsittää suorittimen tehon, muistin ja levytilan määrän (kuva 2). Tämän jälkeen suoritetaan tilaus ja odotetaan hetki, kunnes konesali on valmis käytettäväksi.



Kuva 2. eSalin hallintaportaalin tarjoama virtuaalisen konesalin tilauslomake.

4.3.2 Kapasiteetti

Perustettaessa virtuaalista konesalia tarvitaan saliin tarpeeksi kapasiteettia, joka voidaan jakaa virtuaalipalvelimien kesken. Virtuaalikoneelle allokoidut resurssit näkyvät ja käyttäytyvät koneessa kuten fyysisissäkin tietokoneissa.

Oman ympäristön kokoonpanoon valittiin prosessitehoa 3 000 MHz, keskusmuistikapasiteettia 16 384 MB ja levypalveluja yhteensä 70 GB jaettuna kahteen eri levyjärjestelmätasoon (engl. Tier). Valittavana on kaiken kaikkiaan kolme eri levyjärjestelmätasoa, joiden suorituskyky vaihtelee siten, että taso 1 on tehokkain. Toimittaja takaa näille levyjärjestelmille tietyn suorituskyvyn ja asettaa enimmäisrajat levyjärjestelmän käytölle. [1.] Kuvassa 3 on esitetty käytössä olevan kapasiteetin käyttöastemittari.

.

Kuva 3. Kapasiteetin käyttömittari näyttää, kuinka paljon kapasiteettia on käytössä. Mittari näyttää käyttöasteen virtuaalisalikohtaisesti.

Kapasiteettia on helppo muuttaa lennosta ja räätälöidä siitä tilanteeseen sopiva hallintaportaalin kautta. Huomioitavaa kuitenkin on, että kovalevyn kokoa ei voi enää muuttaa pienemmäksi luomisen jälkeen.

4.3.3 Verkkoympäristö

Ennen virtuaalikoneiden luomista on hyvä aloittaa verkkoelementtien tilauksella, jotta luodut koneet voidaan suoraan liittää oikeaan verkkoon. Virtuaalisalin sisäiseen liikenteeseen luotiin Lähiverkko (LAN). Tämä mahdollistaa turvallisemman tiedon kulkemisen virtuaalikoneiden välillä. Verkkoalueelle luotiin myös julkinen IP-osoite ulkoista liikennettä varten. Ulkoinen liikenne määriteltiin kulkemaan palomuuripalvelimen kautta.

eCloud-portaalista saatiin tilattua LAN-verkko polusta Network > LAN. Tilauslomakkeeseen täytettiin kentät Gateway, Netmask ja IP-range. Tämän jälkeen

tilattiin konesalin käyttöön myös julkinen IP-osoite. Julkisen IP-osoitteen tilaus onnistui eCloud-portaalin polusta Network > IP. IP-osoite valittiin annetusta listasta, minkä jälkeen se allokoitiin käyttöön. Ennen kuin julkinen IP-osoite voitiin ottaa varsinaisesti käyttöön, se täytyi sitoa ulkoiseen verkkoon ja muuttaa statukseksi ”enabled”. Tämä onnistui polusta Network > IP > Edit. Kuvassa 4 on esitetty konesalin käyttöön luodut verkkoyhteydet.

Name	Status	Gateway Address 1 ▲	Network Mask	Primary DNS	Secondary DNS
ElisaOyj13-LAN-PTVLAN	✓	10.0.0.1	255.255.255.0	62.142.21.193	62.142.21.194
SDE2T1Z5-ElisaOyj13-Palvelutestaus-VDC-1-Internet	✓	[redacted]	255.255.252.0	62.142.21.193	62.142.21.194

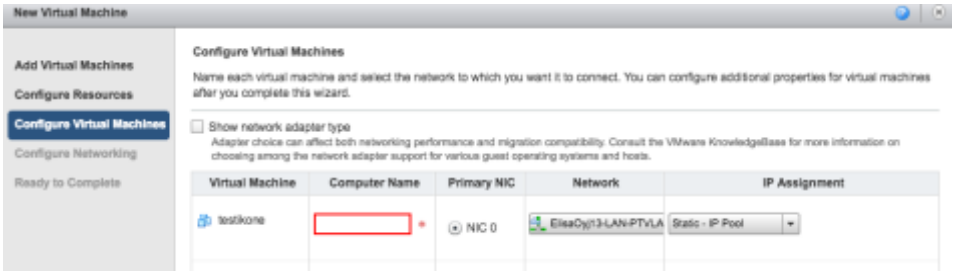
Kuva 4. Virtuaalisalin käytetyt verkot listattuna vcloud-portaalissa.

Luoduista verkoista jaettiin aliverkot luoduille virtuaalikoneille. Jakaminen tehtiin valitsemalla verkkokortin (engl. NIC, Network Interface Controller) käyttämä IP-osoitteiden jakamismalli.

IP-osoitteen jakamismallit:

- *DHCP* – Palvelu vaatii, että palveluun on asennettu DHCP-palvelin jakamaan osoitteet.
- *Staattinen IP-Pooli* – Palvelu käyttää jakamiseen IP-poolia, joka on määritelty palvelussa konesalille.
- *Staattinen IP-Manuaalinen* – Käyttäjä voi käsin määritellä halutun osoitteen koneelle.

Vaihtoehdoista valittiin Staattinen IP-Pooli, joka mahdollisti lähiverkon (LAN) jakamisen konesalin sisäiseen liikenteeseen ja palomuuripalvelimessa myös ulkoiseen verkkoon. Kuvassa 5 näkyy, miten luodaan virtuaalikone ”testikone”, jonka verkkoyhteydet konfiguroidaan toimiviksi. Network-pudotusvalikko listaa kaikki käytössä olevat verkot, josta valitaan sopiva. Seuraavassa pudotusvalikossa valitaan IP-osoitteiden jakamisen malli.

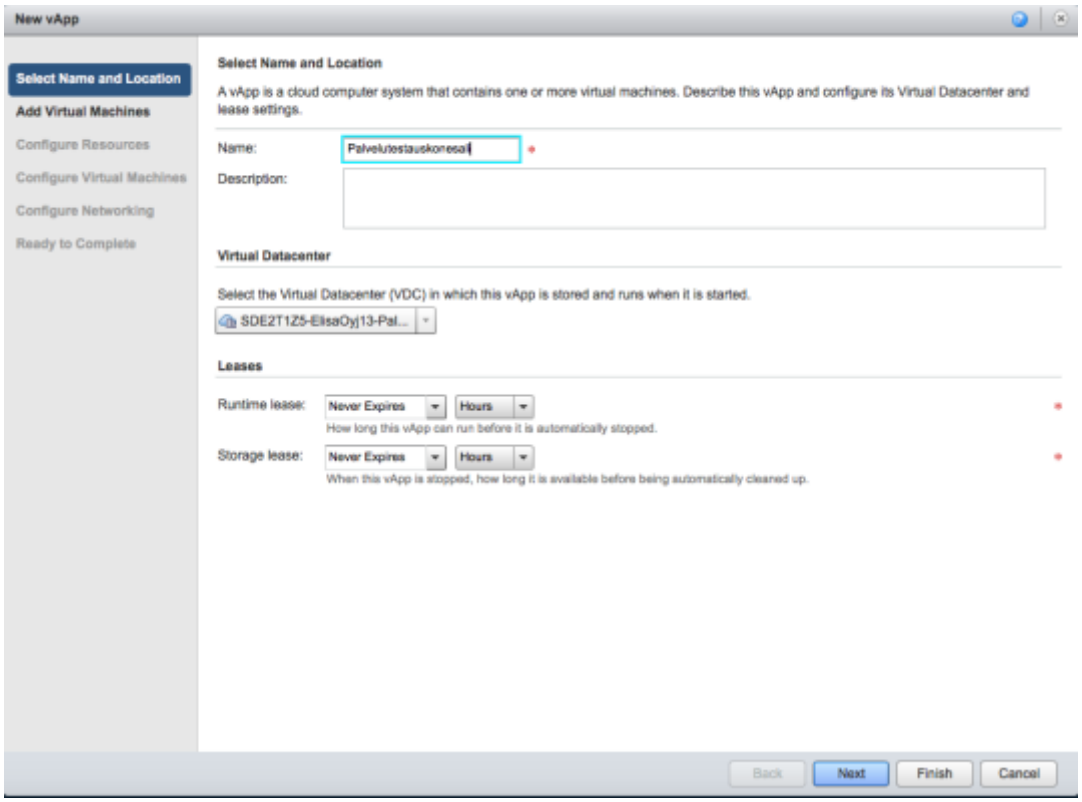


Kuva 5. Uuden virtuaalikoneen käyttöönotossa konfiguroidaan verkkoyhteydet kuntoon.

4.3.4 vApp ja virtuaalikoneet

Konesalin tilauksen ja verkkoyhteyksien luomisen jälkeen luotiin vApp-virtuaalisalikokonaisuus. vApp on virtuaalikoneiden ja verkkorakenteiden kokonaisuus, joka voi sisältää useita virtuaalikoneita. vCloud tarjoaa valmiita vApp-alustoja, joita voi halutessaan ottaa käyttöön julkisesta tuotevalikosta. On myös mahdollistaa rakentaa oma vApp, jonka voi tarpeen vaatiessa kloonata tai tallentaa omaksi alustaksi palveluun yrityksen sisäiseen käyttöön.

Järjestelmään luotiin oma vApp, joka nimettiin palvelutestauskonesaliksi. vAppin asennuksen yhteydessä on mahdollista valita asennettavat virtuaalikoneet julkisesta tuotevalikosta, valita, mitä varastointimallia käytetään, sekä konfiguroida verkkoyhteydet (kuva 6). Nämä toiminnot on kuitenkin mahdollista suorittaa myös jälkikäteen jo luotuun vAppiin.



Kuva 6. Uuden vAppin luominen.

Uusia virtuaalikoneita luodaan helposti vCloud-portaalista avaamalla oma vApp ja klikkaamalla ”New Virtual Machine”. Luodut virtuaalikoneet toimivat hypervisorin päällä, ja niille asennetaan oma käyttöjärjestelmä tavallisen tietokoneen tapaan. Hypervisor-ohjelma sallii useamman käyttöjärjestelmän käyttää samaa laitteistoa kontrolloiden prosessorin ja resurssien käyttöä virtuaalikoneiden välillä.

Kaikkien luotujen virtuaalikoneiden käyttöjärjestelmäksi valittiin julkisesta tuotevalikosta Linux-käyttöjärjestelmä. Jokaiselle järjestelmään tarvittavalle sovelluskokonaisuudelle luotiin oma virtuaalikone, jolloin sovellukset eivät häiritse toisiaan edes vikatilanteissa. Insinöörityössä kuvaillaan kuitenkin vain raportointijärjestelmään liittyviä palveluja konesalissa.

Virtuaalikone: Endian Firewall

Endian Firewall on avoimen lähdekoodin Linux-pohjainen jakelu, joka mahdollistaa reitityksen ja palomuurin. Endian Firewall tarvitaan suojaamaan liikenne ulkoverkosta sisäverkkoon. vCloud-portaalista luotiin uusi virtuaalikone, ja julkisista alustoista valittiin

pohja: Endian Firewall Community v2.5.2 template 1.0, joka on tarkoitettu Endian Firewall -palvelimelle.

Kuvan 7 tilanteessa konfiguroitiin Endian Firewall -palvelimen verkkoasetukset. Konfiguroinnissa valittiin käytettävät verkot ja IP-osoitteiden jakomalli. Verkkokorttien suhteen valittiin NIC #0 /eth0-verkoksi lähiverkko (LAN). Tämä on oletusportti muille virtuaalikoneilla samassa lähiverkossa. NIC #1 / eth1 käytetään julkiseen verkkoon. Ensisijaiseksi verkon liittymän kontrolleriksi valittiin julkinen verkko.

Kuva 7. Endian Firewall -palvelimen verkkoasetusten konfigurointi

Taulukko 3 kuvaa Endian Firewall -virtuaalikoneen kapasiteetin. Valittu kapasiteetti oli oletuksena kyseisellä koneella, eikä sitä ollut syytä muuttaa.

Taulukko 3. Endian Firewall -virtuaalikoneen kapasiteetti.

Virtuaalikoneen ominaisuudet	**Kapasiteetti**
Levytila	10 GB (Disk 0)
Keskusmuisti	512 MB
Virtuaalinen suoritin	1 kpl

Virtuaalikone: Endian Firewall -hallinta

Endian Firewall -palvelinta varten tarvittiin konfigurointipalvelin, jonka avulla hallitaan palomuuria web-käyttöliittymän kautta. Tämän virtuaalikoneen alustaksi valittiin eSalin tuotevalikosta Linux Ubuntu Desktop 14.04.4 ja käyttöjärjestelmäksi Ubuntu (32-bit). Virtuaalikoneeseen asennettiin työpöytäympäristö, niin että sillä pääsee kontrolloimaan palomuuriohjelmaa. Taulukko 4 kuvaa Endian Firewall hallinta -virtuaalikoneen kapasiteetin. Valittu kapasiteetti oli oletuksena kyseisellä koneella, eikä sitä ollut syytä muuttaa.

Taulukko 4. Endian Firewall hallinta -virtuaalikoneen kapasiteetti.

Virtuaalikoneen ominaisuudet	**Kapasiteetti**
Levytila	11 GB (Disk 0, Disk 1)
Keskusmuisti	1 GB
Virtuaalinen suoritin	1 kpl

Virtuaalikone: Raportointipalvelin

Raporttipalvelin luotiin suorittamaan Splunk Enterprise -ohjelmaa ja vastaanottamaan testidataa. Virtuaalikoneen alustaksi valittiin eSalin tuotevalikosta Linux Ubuntu Server 14.04.4, käyttöjärjestelmänä Ubuntu (64-bit). Virtuaalikone asennettiin ilman graafista käyttöliittymää, jolloin virtuaalikonetta hallitaan komentotulkista. Tällä valinnalla haluttiin vähentää turhia resurssitarpeita.

Komentotulkkiesimerkissä 1 on raportointipalvelimen peruskonfiguraatiot. Rivillä 1 ja 2 järjestelmään luotiin yksi uusi käyttäjä. Rivillä 3 käyttäjälle asennettiin kotihakemisto /home-hakemiston alle. Rivillä 4 käytettiin visudo-ohjelmaa muokkaamaan etc/sudoers-asennustiedostoa. Tässä tiedostossa määritellään, mitä kukakin saa tehdä ja millä oikeuksilla sudo-ohjelmaa käyttäen [21]. Käyttäjälle annettiin oikeudet käyttää sudoa kaikissa komennoissa salasanaa vastaan lisäämällä tiedostoon rivin 5 komento. Myöhemmin järjestelmään lisättiin myös toinen käyttäjä samoin oikeuksin.

```
1. # useradd username
2. # sudo passwd username
3. # mkdir /home/usename
4. # sudo visudo
5. # username ALL=(ALL:ALL) ALL
```

Komentotulkkiesimerkki 1. Raportointipalvelimen peruskonfiguraatiot.

Taulukko 5 kuvaa Raportointipalvelin-virtuaalikoneen kapasiteetin. Keskusmuistia oli lisättävä alkuasetusten jälkeen, sillä Splunkin toiminta oli toisinaan pysähtynyt liian pienen keskusmuistin takia.

Taulukko 5. Raportointipalvelin-virtuaalikoneen kapasiteetti

Virtuaalikoneen ominaisuudet	Kapasiteetti
Levytila	30 GB (Disk 0, Disk 1)
Keskusmuisti	12 GB
Virtuaalinen suoritin	2 kpl

4.3.5 Varmuuskopiointi

Elisa eSalissa on tarjolla snapshot-toimintoon perustuva varmuuskopiontiratkaisu. Virtuaalikoneisiin ei siis tarvinnut asentaa erikseen varmuuskopiontiohjelmia, vaan varmuuskopioiden ajastus ja palautus onnistuvat hyvin hallintapaneelin kautta. eSalin varmuuskopiointi toimii kokonaisena varmuuskopiointina, mikä tarkoittaa, että käyttäjä pystyy palauttamaan varmuuskopion uutena virtuaalikoneena ja manuaalisesti kopioimaan tiedostoja uudesta alkuperäiseen virtuaalikoneeseen. Varmuuskopiointiasetuksissa määritellään varmistuskierto ja hallitaan palautuksia. [1.]

5 Raportointijärjestelmän rakentaminen eSaliin

Seuraavissa alaluvuissa käydään läpi, miten raportointijärjestelmä rakennettiin toimimaan virtuaalisessa konesalissa. Ensin avataan teoriaa datan visualisoinnista, ja sen kautta annetaan kuvaa siitä, millainen käyttäjille avautuva raportointijärjestelmän käyttöliittymä takaisi parhaan käytettävyyden. Tämän jälkeen käydään läpi raportointijärjestelmän rakennusvaiheet: integrointisovelluksien ja pääsovelluksena toimivan Splunk-ohjelman asennukset. Viimeisenä tarkastellaan loppukäyttäjälle avautuvaa käyttöliittymää.

5.1 Datan visualisointi

Visualisointia on käytetty jo vuosisatoja edistämään monimutkaisten asioiden ymmärtämistä, ja onkin ilmeistä, että uusi tieto omaksutaan nopeimmin visuaalisesti. Kuvat myös muistetaan paremmin kuin pelkkä teksti. Visualisoinnin keinot ovat moninaiset, ja ne ovat alkaneet jo luolamaalauksien tekemisellä vuosisatoja sitten.

Nykyisin tunnetut tilastografiikan peruskuviotyypit esiteltiin kansantaloustieteilijä William Playfairin *Commercial and Political Atlas* -teoksessa vuonna 1786. [13.]

Visuaalisilla esityksillä voi olla monia tarkoitusperiä. Yleisin lienee kuitenkin tietoa välittävä grafiikka, joka pyrkii havainnollistamaan katsojalle jotain tiettyä asiaa. Raportointijärjestelmän graafisessa esityksessä halutaan esittää testitulokset informatiivisesti ja selkeästi niin, että se on tulkitsijalle merkityksellistä ja siitä voidaan tehdä helposti johtopäätöksiä testatun laitteen käyttöönotosta. Tärkein osa-alue tässä on kuitenkin poikkeamien tunnistaminen testituloksista, mikä sisältää mahdollisen esityksen siitä, mistä poikkeama on johtunut. Tämä graafinen esitys esitetään osittain interaktiivisilla työpöydillä (engl. dashboard), joihin on kerätty olennaiset mittarit ja raportit. Työpöytä on ikään kuin kojelauta, josta on mahdollista nopealla vilkaisulla tehdä johtopäätöksiä testien kokonaistilanteesta.

Seuraavissa alaluvuissa perehdytään siihen, millä eri tavoin päästään haluttuun lopputuloksen datan visualisoinnissa. Luvuissa ei mennä itse visualisointiprosessiin vaan keskitytään ainoastaan prosessin lopputulokseen. Aihetta avataan teorialla ihmisaivojen havainnointikyvystä, minkä jälkeen avataan käytetyt kuvaajat ja vuorovaikutukselliset elementit.

5.1.1 Havainnointi

Ennen varsinaiseen visuaaliseen esitykseen menemistä on hyvä tutustua siihen, miten ihmisen havaintomekanismi toimii, sillä ihmisaivot yksikertaisesti havainnoivat visuaalisia asioita, kun taas esimerkiksi numeroita luetaan. Havainnoinnin tunteminen auttaa datan visualisoinnin suunnitteluvaiheessa ohjaamaan lukijan katsetta kuvissa ja hallitsemaan näistä kuvista tehtyjä tulkintoja. [14.]

Havaintopsykologi Stephen Kosslyn esittää ihmisen havaintoprosessin kolmen väittämän avulla. Ensimmäisen väittämän mukaan ihmismieli ei ole kamera. Tällä tarkoitetaan sitä, miten ihminen havainnoi merkityksellisiä asioita sivuuttaessaan muut. Mieli on valikoiva eikä toimi kameran tavoin [15]. Kun tämä väittämä yhdistetään datan visualisointiin, on suunnittelijan mietittävä, mitä asioita korostetaan ja millä tavalla korostetaan ja vastaavasti myös sitä, miten välttyä tahattomalta korostamiselta. Esimerkiksi voimakkaat värit tulkitaan usein merkityksellisemmiksi kuin vaaleat värit.

Värien käytössä on kuitenkin otettava monia asioita huomioon: esimerkiksi jopa 8 % miehistä kärsii punavihersokeudesta [14]. Värien käytössä on myös käytettävä malttia, ettei lopputulos ole liian sekava.

Toinen väittämä esittää mielen arvioivan kirjaa sen kannen perusteella. Tämä tarkoittaa sitä, kuinka visuaalinen tulkintamekanismi ja muistijärjestelmä toimivat yhdessä. Kun ihminen näkee esimerkiksi katkoviivan, hän olettaa sen muodostavan samanlaisen kuvion kuin yhtenäinen viiva [15]. Ihmisaivot mieltävät kuviojoukkoja yhtenäiseksi kuvioksi tämän mekanismin avulla, joka tunnetaan myös jatkuvuuden lakina hahmopsykologian terminologiassa. Datan visualisoinnissa on pyrittävä tämä väittämä huomioiden yhdenmukaisuuteen datan esittämisessä. Vertailua tehdessä kannattaa suosia lineaarisuutta, jolloin silmän on helpompi hahmottaa kokonaisuus. Visualisoinnissa on myös huomioitava mahdollisesti puuttuva data, joka voi tehdä kuvaajasta katkonaisen, ja mieli kuitenkin hahmottaa kuvaajan yhtenäiseksi yksinkertaisimmalla tavalla. Tämä voi antaa mahdollisesti vääristyneen kuvan datasta.

Kolmas väittämä esittää, miten halua on, mutta rahkeet eivät riitä. Tällä viitataan mielen rajalliseen kapasiteettiin muistaa asioita. Visualisoinnilla on mahdollista kuitenkin yhdistää pienempiä yksikköjä isommaksi kokonaisuudeksi, ja sen avulla voidaan muistaa suurempia tietomääriä samanaikaisesti. [15.]

5.1.2 Kuvaajat

Seuraavaksi esitellään kuvaajat, joita käytetään raportointijärjestelmän työpöydissä. Kuvaajien valinnoilla pystytään vaikuttamaan siihen, mitä asioita halutaan korostaa tuloksista. Kaikki kuvaajat eivät sovellu kaiken tyyppiselle datalle, ja valintaan vaikuttavat monet tekijät: mikä on datan mittayksikkö, kuinka moniulotteista data on, mikä on muuttujien asteikko, kuinka monta elementtiä kuvioon on saatava ja kuinka paljon tilaa on käytettävissä. [15.]

Oikeanlaisten kuvaajien valinnan jälkeen on pidettävä mielessä se, miten kuvaaja antaa parhaiten totuudenmukaisen kuvan datan luonteesta. Datan luonnetta voi helposti muuttaa mahdollisin rajauksin ja näkökulmavalinnoin. Edward Tufte esittelee teoksessaan *The Visual Display of Quantitative Information* termin valhekerroin (engl. Lie-factor), jolla hän esittää vääristymän dataan perustuvan visualisoinnin ja varsinaisten

datan arvojen välillä luvulla, joka sadaan matemaattisella kaavalla. Yksinkertaisesti esitettynä: jos kahden asian välinen ero on datassa 50 prosenttia, on sen myös oltava sama visualisoinnissa. [15.]

Ympyrädiagrammi

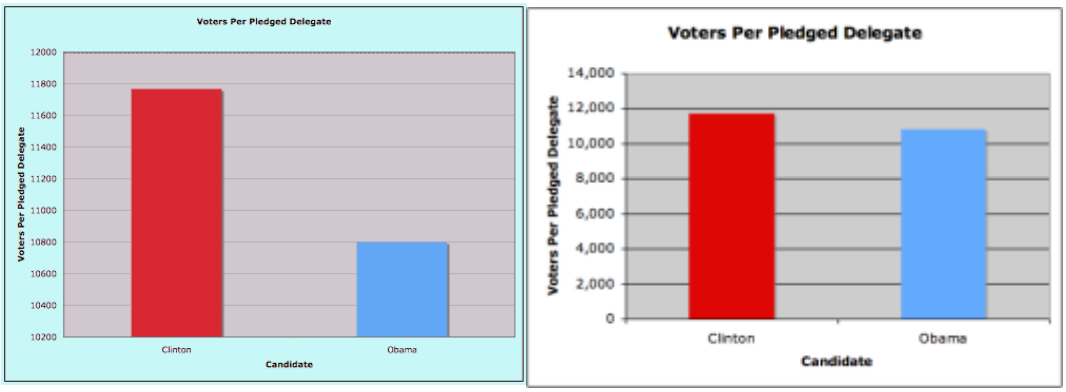
Ympyrädiagrammi on yksi yleisimmistä kuvaajista, ja se on jaettu osien kokoja kuvaaviin sektoreihin. Ympyrädiagrammi on esteettisesti miellyttävä diagrammi, mutta se ei nauti suurta arvostusta ammattimaisessa tienon visualisoinnissa. Tämä siksi, että sen käyttömahdollisuudet ovat hyvin rajalliset eikä sen katsota soveltuvan kovin hyvin tarkempaan tiedon vertailuun. Vertailu on ongelmallista, sillä katsojat saattavat verrata eri kohteita sektoreista. Osa ihmisistä vertaa kaaren pituutta, osa jänteen pituutta ja osa sektorin keskuskulmaa. Oikea vertailukohde on kuitenkin sektorin pinta-ala. [16.]

Prosenttiosuudet on helpointa näyttää ympyrädiagrammin avulla, ja se onkin ainoa, mihin se soveltuu hyvin. Ympyrädiagrammissa kannattaa pitää mielessä rajoituksena se, että enempää kuin seitsemän eri arvoa ei tule näyttää samanaikaisesti, muuten kuvaaja ei enää palvele tarkoitustaan esittää tietoa selkeästi. [16.]

Pylväsdiagrammi

Pylväsdiagrammi koostuu palkeista, joiden korkeus kuvaa arvon suuruutta. Vertailuarvoihin käytetään yleensä pylväsdiagrammia. Se on helppo tapa erottaa, mitkä ovat suurimmat ja pienimmät arvot. [16.]

Pylväiden tasalevys on tärkeää, ja määräasteikko on aloitettava aina nollasta, sillä muuten voi tulkinta vääristyä ja hankaloitua. Kuvassa 8 huomataan, miten erinäköisiä kuvaajia saadaan, kun nollapisteen arvoa muutetaan. Data on täysin totuudenmukaista molemmissa kuvissa, mutta ensimmäinen kuva antaa katsojalle vääristyneen käsityksen, kun nollapiste on jotain muuta kuin nolla. [15.]



Kuva 8. Pylväiden välisen suhteen vääristymä pylväsdiagrammissa, kun nollapisteen arvoa vaihdellaan [17].

Kun tehdään vertailua pylväsdiagrammilla, on syytä välttää myös käyttämästä kolmiulotteisuutta, perspektiivisyyttä tai symboleja pylväiden tilalla. Niiden käyttö hankaloittaa vertailtavuutta ja saattaa antaa vääristynyttä kuvaa datasta. Yksinkertainen kuvaaja on yleensä toimivin. [15.]

Viivadiagrammi

Viivadiagrammi on sarja informaatiopisteitä, jotka on yhdistetty kuvaajaan viivana. Tätä kuvaajaa käytetään yleensä kuvaamaan trendiä jonakin tiettynä ajanjaksona. Viivakuvio kuvaa parhaiten muutosta tai kehitystä tai niiden puutetta. Se korostaa kehityssuuntaa ja vaihtelua sen suhteen. Jos kuviossa on useampi viiva, nähdään myös kehityssuuntien erot. [36, s. 78.]

Viivadigrammissa x-akselille sijoitetaan yleensä aika ja y-akselille määrä. Kuvioon ei kannata mahduttaa liian montaa viivaa, sillä siitä tulee helposti vaikealukuinen. Tärkeää on myös valita oikea tarkkuusaste ja määritellä oikein vaaka- ja pystyakselien asteikkojen suhde eli aspektisuhde. Muutokset aspektisuhteessa aiheuttavat voimakkaimmin vääristymien mahdollisuuden. Sama kehitys voidaan saada näyttämään aspektisuhdetta muuttamalla joko lievältä tai jyrkältä. [18.]

5.1.3 Vuorovaikutuksellisuus

Vuorovaikutuksellisuus on tärkeä tapa lisätä tiedon merkityksellistämistä. Vuorovaikutuksen ansiosta datan esitykseen saadaan upotettua useita ulottuvuuksia,

jolloin tarkasteltavaa datajoukkoa voidaan esimerkiksi rajata tai samaa datajoukkoa voidaan tarkastella useasta näkökulmasta.

Käyttöliittymäsuunnittelussa on joukko erilaisia helpottavia hallintaelementtejä, joita kutsutaan kontrolleiksi. Kontrollit voidaan jakaa neljään eri pääryhmään, jotka ovat käskykontrollit, valintakontrollit, lisäyskontrollit ja esityskontrollit. [19.]

Käskykontrollit sisältävät toimintakäskyn ja kehotavat käyttäjää suorittamaan jonkin toiminnon. Napit, hyperlinkit sekä työkalupainikeet lukeutuvat kaikki käskykontrolleiksi [19]. Splunk-työpöydällä käytetään käskykontrolleista lähinnä hyperlinkkejä, joiden avulla suunnistetaan näkymästä toiseen. Hyperlinkkejä käytetään tekstielementeissä.

Valintakontrollit tarjoavat käyttäjälle erilaisia vaihtoehtoja, joista valita sopivin. Valinnan vahvistaminen tarvitsee kuitenkin rinnalle käskykontrollin. Yleisiä valintakontrolleja ovat pudotuslistat, valintapainikkeet ja radionapit. [19.]

Radionappeja käytetään yleensä pieniin listoihin. Kaikki vaihtoehdot ovat käyttäjälle näkyviä, ja niistä voi valita vain yhden kerrallaan. Toisin kuin valintapainikkeissa, voi valintoja tehdä niin monia, kuin on vaihtoehtojakin. Radionappi on yleisesti ympyränmallinen ja valintapainike neliö. Pudotuslistaan taas on helppo sisällyttää suuria määrä vaihtoehtoja, jotka pysyvät piilossa, ellei niitä selata. Kun vaihtoehtojen määrä kasvaa suureksi, on hyvä helpottaa oikean vaihtoehdon valitsemista hakukentän avulla. [20.]

Lisäyskontrollit tarjoavat käyttäjälle mahdollisuuden tiedon lisäämisen itse. Yleisin lisäyskontrolli on tekstikenttä, jota käyttäjä voi muokata. Lisäyskontrollit jaetaan rajoitettuihin ja rajoittamattomiin. Rajoitetut kontrollit asettavat reunaehdot käyttäjän omalle valinnalle. Tästä hyvänä esimerkkinä ovat voimakkuudensäädöt. Rajoittamaton kontrolli voi olla esimerkiksi palautelomake, johon käyttäjä voi vapaasti kirjoittaa kommenttinsa. [19.]

Esityskontrolleja käytetään datan visuaalisen esittämisen hallintaan. Yksinkertaisin esimerkki tästä lienee vierityspalkki, jonka avulla käyttäjä pääsee liikkumaan sivustolla. Toinen mielenkiintoinen esimerkki on vetolaatikko (engl. drawer), joka kätkee taakseen tietoa, jota voi hallita vaihtamalla kytkimen tilaa. Vetolaatikon taakse voi siis piilottaa

tietoa, jonka ei tarvitse olla koko ajan esillä. Näin saa käyttöliittymästä selkeämmän ja tilaa jää enemmän. Muita esityskontrolleja ovat ruudukot (engl. grid), apuviivat (engl. guidelines) ja viivoittimet (rulers). [19.]

5.2 Integraation mahdollistavat sovellukset

5.2.1 Palomuuri

Palomuuri tarvitaan kontrolloimaan verkkoliikennettä lähiverkon (LAN) ja julkisen tietoverkon (internet) välillä. Sille annetaan joukko sääntöjä, jotka määritelevät, millaista liikennettä halutaan päästää läpi. Säännöt sisältävät erilaisia elementtejä, kuten lähde- ja kohdeosoitteet, protokollat ja porttien numerot. Seuraavaksi käydään läpi konesalin palomuurin peruskonfiguraatio ja palomuurisääntöjen luominen ulospäin menevälle ja sisäänpäin tulevalle liikenteelle. [22.]

Palomuurin konfigurointiin päästiin käsiksi avaamalla aikaisemmin luotu palomuurinhallinta-virtuaalikone. Tähän hallintakoneeseen on käyttäjätunnukset ja salasanat eSalin dokumentoinnissa. Palomuurihallinta-virtuaalikoneelta avattiin internetselain ja suunnistettiin Endian Firewall -hallintapalveluun, joka on osoitteesta https://10.0.0.1:10443/. Käyttäjätunnukset ja salasanat palveluun ovat myös eSalin dokumentaatiosta. Kirjautumisen jälkeen palomuuriin konfiguroitiin julkinen (RED) ja sisäinen (GREEN) rajapinta valitsemalla hallintapalvelusta: System > network configuration. Konfiguraatiota suorittaessa ei tarvinnut muuttaa kuin oikea julkinen IP-osoite.

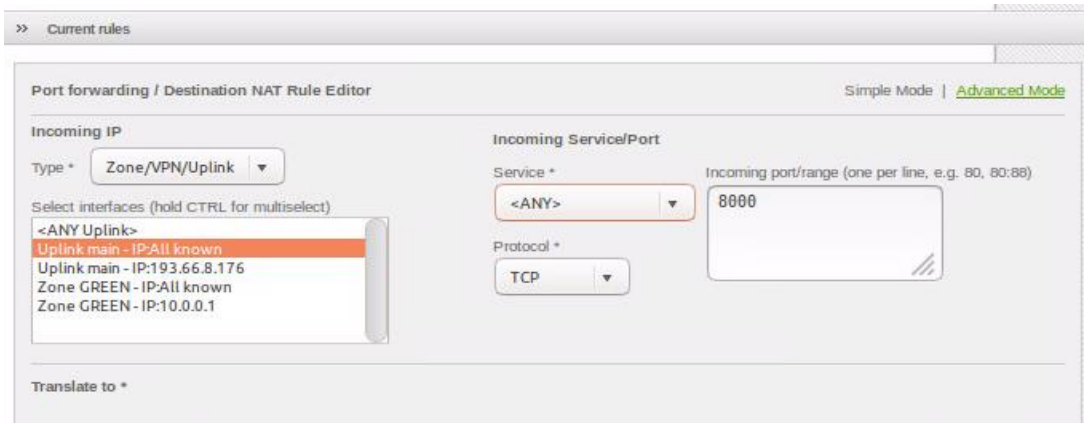
GREEN -> RED -palomuurisäännöt ovat sääntöjä ulospäinlähtevälle liikenteelle. Palomuuriin on määritelty oletuksena joukko porttisääntöjä, joita voi halutessaan muokata, lisätä tai poistaa. Oletussääntöjen joukkoon lisättiin uusi sääntö Splunkin lisenssipalvelimelle, joka käyttää porttia 8089. Liitteessä 2 on näyttökuva ulospäin menevistä palomuurisäännöistä.

RED -> GREEN -palomuurisäännöt ovat sääntöjä, joilla määritellään sisäänpäin tulevan liikenteen parametrit. Sääntöjä käytetään, kun halutaan julkaista palvelu julkisella

osoitteella. Julkaisu tehdään käyttämällä porttien uudelleenohjausta (engl. port forwarding). Sääntöjen hallinta on hallintapaneelista: Firewall > Port forwarding / NAT.

Porttien uudelleenohjauksessa luodaan DNAT-sääntöjä (engl. Destination network address translation), ja näin muuttavat verkkoon tultaessa ulkoverkon IP-osoitteen oikean sisäverkon laitteen osoitteeksi, jota kautta saadaan yhteys palveluun. Halutessaan voisi sääntöjä myös kohdistaa tiettyyn ulkopuolelta tulevaan ryhmään: verkko, zone, VPN tai käyttäjä.

Tarvittavat säännöt luotiin kaikille käytettäville palveluille. Luoduilla säännöillä saatiin julkinen liikenne ohjattua oikeaan sisäverkon koneeseen. Kuvassa 9 luotiin sääntöä Splunkin käyttämälle käyttöliittymälle, joka käyttää porttia 8000. Taulukossa 6 on listattu kaikki DNAT-säännöt, joita tarvitaan raportointipalvelimessa.



Kuva 9. DNAT-säännön luominen Splunkin käyttöliittymälle.

Taulukko 6. Virtuaalisalin raportointipalvelimen käyttämät palvelut ja portit.

Portti	**Palvelu**	**Käyttötarkoitus**	**Protokolla**
8000	Splunk Web	Web-käyttöliittymä	HTTP/HTTPS
8080	Splunk Forwarder	Datan kuljettaminen	TCP
22	SSH	Etäkäyttö kehityskoneelta	TCP
21	FTP	Datan liikkuminen viSerista	TCP

Sisäverkon sisällä tapahtuvaan liikenteeseen on myös omat säännöt kohdassa Inter-Zone traffic. Sännöissä on määritelty, että kaikki sisäinen liikenne on sallittua.

5.2.2 SSH-ohjelmisto

SSH on etäkäyttöohjelmisto, jolla voidaan ottaa salattuja yhteyksiä järjestelmästä toiseen. Useimmiten SSH-asiakasohjelmaa käytetään komentorivin kautta, mutta mahdollista on myös suorittaa graafista ohjelmaa SSH:n kautta. Tällöin tosin pullonkaulaksi muodostuu hidas yhteys. [23.]

Virtuaalikoneiden etäkäyttö on mahdollista VMware Remote Console -lisäosalla suoraan verkkoselaimella vCloud directorin kautta. Tätä tapaa suositellaan kuitenkin ainoastaan virheiden korjauksiin ja ensi asennuksiin. Muussa tapauksessa tarjoaa SSH:n kautta muodostettu yhteys huomattavasti paremman suorituskyvyn ja käytettävyyden. [1.]

SSH-yhteyttä päätettiin käyttää etäyhteyden muodostamiseen, jotta saatiin suojattu ja tunneloitu TCP-yhteys koneiden välille. Tällä tavalla on helppoa omalta kehityskoneelta admin-tason tunnuksilla hoitaa virtuaalikoneiden konfigurointi kuntoon. SSH-palvelin oli valmiina virtuaalikoneessa, mutta yhteydet olivat oletuksena estettynä. Yhteydet piti hyväksyttää /etc/hosts.allow **-**tiedostossa, jossa määritellään pääsy tiettyihin tcpd-ohjelman ohjaimiin [1]. Vaikka tcpd ei hallinnoikaan SSH-palvelua, lukee SSH-palvelin silti tämän tiedoston [21]. Tiedosto avattiin pääkäyttäjänä vi-editorilla käyttäen komentoa sudo vi /etc/hosts.allow ja tiedoston sshd-riviin lisättiin ”ALL”.

SSH-yhteydet sallittiin siis kaikilta asiakaskoneilta, mutta yhteys kiellettiin muodostamasta pääkäyttäjän Root-tunnuksin muokkaamalla OpenSSH:n asennustiedostoa /etc/ssh/sshd_config. Tiedostossa sijaitsevat kaikki tulevien yhteyksien asetukset. Asennustiedosto avattiin vi-editorilla, käyttäen komentoa vi /etc/ssh/sshd_config ja lisättiin uusi rivi PermitRootLogin no.

5.2.3 FTP-palvelin

FTP (File Transfer Protocol) on tiedostojen siirtämiseen suunniteltu protokolla. Se on komentoriviohjelma, joka vaatii TCP/IP-protokollaa toimiakseen. FTP-palvelinta tarvitaan aikaisemmin kuvatun viSerin testituloksien siirtymiseen puhelimista

raportointipalvelimelle. Raportointipalvelimelle asennetiin OpenSSH-palvelin, joka sisältää mahdollisuuden SFTP:llä tapahtuvaan tiedostojen siirtoon. SFTP vastaa täysin FTP:tä, mutta liikenne on salattua SSH-pääteyhteyden tapaan [24]. Raportointipalvelimelle haluttiin mahdollisuus käyttää molempia protokollia, SFTP:tä ja FTP:tä. Asennus tehtiin omalta kehityskoneelta SSH:n kautta.

Komentotulkkiesimerkissä 2 rivillä 1 OpenSSH asennettiin apt-pakettienhallintajärjestelmällä. Rivillä 2 luotiin uusi ryhmä ftp_admin FTP-käyttäjiä varten. Rivillä 3 luotiin uusi käyttäjä ftp_admin ryhmään ftpaccess. Rivillä 4 vaihdettiin omistajuus kotihakemistolle. Rivillä 5 ja 6 luotiin kotihakemiston sisälle viser-hakemisto, johon ftp_admin-käyttäjälle annettiin oikeudet.

```
1. # sudo apt-get install openssh-server
2. # sudo groupadd ftpaccess
3. # sudo usermod -m ftp_admn -g ftpaccess -s /user/sbin/ndologin
4. # sudo chown root /home/ftp_admin
5. # sudo mkdir/home/ftp_admin/viser
6. # sudo chown ftp_admin: ftpaccess /home/ftp_admin/viser
```

Komentotulkkiesimerkki 2. FTP-palvelimen konfigurointi.

Splunk monitoroi mainittua viser-hakemistoa ja lataa tiedostoja määriteltyjen sääntöjen mukaan, kun uusia tiedostoja saapuu. Tiedostot siirtyvät hakemistoon automaattisesti viSerin toimesta testiajon jälkeen.

5.3 Splunk-ohjelmisto

Testitulosten käsittelyyn päätettiin käyttää Splunk Enterpriseä, sillä se vakuutti monipuolisuudellaan ja näyttävillä raportointiominaisuuksilla. Ennen insinöörityötä oli Splunk asennettu työpöytäkoneelle ja front-end-puoli rakennettu valmiiksi käyttäen muun muassa yksinkertaista XML-lähdekoodia. Insinöörityön tehtäväksi jäi asentaa ja konfiguroida Splunk toimimaan uuteen ympäristöön. Valmiiksi tehty XML-lähdekoodi oli helppo liittää uuteen instanssiin.

Seuraavaksi kuvaillaan Splunkin taustaa, ohjelman arkkitehtuuria, työssä käytettyjä komponentteja ja konfigurointiosuutta. Front-end-puolen toimintoihin luodaan katsaus luvussa 5.3.5. Se koetetaan kuitenkin pitää suhteellisen kevyellä tasolla, sillä sen läpikäyminen syvällisesti vaatisi oman insinöörityön. Sen avulla kuitenkin havainnollistuu, mitä Splunkilla pystyy tekemään.

Splunk on amerikkalainen monikansallinen yhtiö, jonka pääkonttori sijaitsee San Franciscossa, Kaliforniassa. Yhtiön ydinliiketoimintaan kuuluu konetiedon etsimiseen, valvontaan ja analyysiin keskittyvän ohjelmiston kehitys. Yhtiö perustettiin vuonna 2003, ja nykyään sillä on yli 700 työntekijää ympäri maailmaa. Nykyään Splunkin asiakaskunta on kohtuullisen laaja: yli 4 800 asiakasta yli 85 maassa. Forten lisäksi partnerikuntaan kuuluu suuria nimiä, kuten Cisco, Vmware ja Blue Coat. Nimi Splunk tulee englanninkielisestä sanasta splunk, jolla viitataan luolien tutkimiseen. [25.]

5.3.1 Asennus

Käytössä oli aluksi ilmainen Splunk Enterprise -lisenssi, johon tarjottiin 60 päivän kokeilujakso. Tämän jälkeen oli mahdollista siirtyä käyttämään ilmaista lisenssiä, jossa on rajalliset toiminnot ja datan päiväkohtainen indeksointiraja 500 megatavua, tai vaihtaa maksullisen Enterprise-lisenssiin [26]. Enterprise-lisenssiin päädyttiin, sillä talon sisällä lisenssi oli jo käytössä, ja se oli mahdollista jakaa raportointijärjestelmän käyttöön.

Splunk asennettiin eSalin Ubuntu-käyttöjärjestelmällä varustettuun virtuaalikoneeseen, joka nimettiin raportointipalvelimeksi. Asennus piti tehdä komentotulkin kautta, sillä desktop-näkymä puuttui. Asennus tehtiin SSH-yhteydellä kehityskoneelta käyttäen Wget-komentoriviohjelmaa ja Splunkin lataussivun tarjoamaa URL-linkkiä. Asennuspaketiksi valittiin .deb-tiedosto, sillä Ubuntu käyttää formaattia binaarilatauksiin. Debian-latauksessa huomioitavaa oli, että Splunkin pystyy asentamaan ainoastaan oletuspolkuun /opt/splunk, eikä se voinut olla symbolinen linkki [27]. Komennot ohjelman lataukseen ovat komentotulkkiesimerkissä 3.

```
# sudo wget -O latausosoite
# sudo dpkg -i splunk-6.5.0-59c8927def0f-linux-2.6-amd64.deb
# sudo /opt/splunk/bin/splunk start
```

Komentotulkkiesimerkki 3. Splunk-ohjelmiston asentaminen.

Lisenssiehtojen hyväksymisen jälkeen komentotulkki ilmoitti käyttöliittymän löytyvän osoitteessa http://193.66.8.176:8000. Käyttöliittymään pääsy vaati kuitenkin, että Splunkin HTTP/HTTPS-portti 8000 avattiin palomuurikoneen porttien uudelleenohjauksesta. Käyttöliittymän ensimmäisellä kirjautumiskerralla käytettiin oletustunnuksia: admin/changeme, minkä jälkeen käyttäjätunnus ja salasana vaihdettiin.

Enterprise-lisenssiä varten oli tehtävä palomuurinavauspyyntö IT-itsepalveluun, jotta yhteys talon sisäisen linsenssipalvelimen porttiin 8089 mahdollistuisi. Tämän jälkeen oli varmistettava eSalin palomuurikoneen porttien uudelleenohjauksesta, että tarvittava sisäisen liikenteen portti 8000 on myös auki, jotta lisenssipalvelimen yhteys kulkeutuu palomuurikoneelta Splunk-palvelimelle.

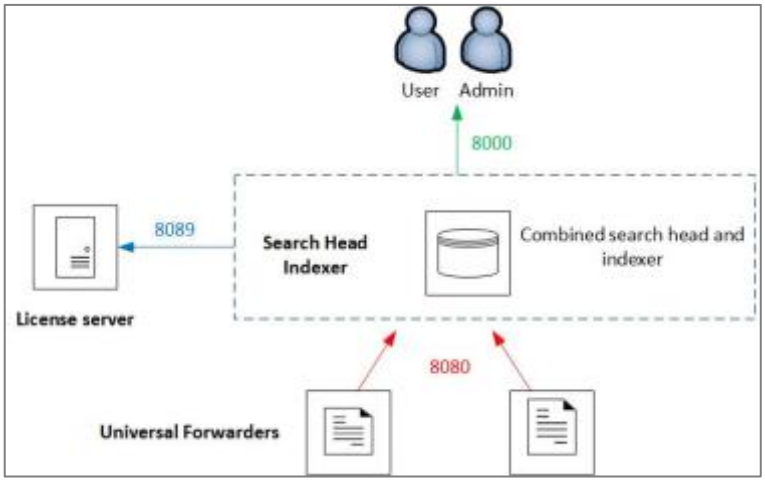
Palomuuriavauksen jälkeen lisenssipalvelimen tiedot syötettiin Splunkin käyttöliittymän tarjoamiin lisenssiasetuksiin. Splunk-palvelin alkoi näin toimia isäntälisenssipalvelimen orja-instanssina kommunikoiden jatkuvasti kaikesta, mitä järjestelmässä tapahtuu. Jos kommunikointiyhteys on estynyt, aloittaa orja-instanssi 72 tunnin ajastuksen. Ajastuksen jälkeen tarkistetaan yhteys, ja jos se on edelleen estynyt, kaikki datahaut indekseistä estetään, kunnes yhteys on jälleen aktiivinen. [28.]

5.3.2 Splunkin arkkitehtuuri

Splunk sisältää useita komponentteja: indeksoija, search head, forwarder ja deployment-server. Ne toimivat yhdessä muodostaen kokonaisen lokienhallintajärjestelmän. Seuraavaksi esitellään raportointijärjestelmän käyttämät komponentit ja niiden toiminta.

Yksinkertaisimmillaan Splunk toimii yhtenä instanssina yhdellä alustalla. Tämä yksi instanssi hoitaa kaikki edellä mainittujen komponenttien roolit tai osan niistä. Suuremmissa ympäristöissä instanssit kuitenkin jaetaan toiminnallisuuden mukaan useampaan osaan toimimaan klustereina, jolloin datakuorma jakautuu tasaisesti. [29.]

Kuvassa 10 on esitetty Raportointipalvelimen Splunk-instanssin komponentit ja käytetyt portit. Raportointipalvelimelle tuleva data on niin vähäistä päivätasolla, että toiminta on mahdollista suorittaa yhdellä Splunk-instanssilla, joka hoitaa indeksoinnin ja tiedon haun. Tämän lisäksi käytetään kahta Splunk forwarder -instanssia datan kuljettamiseen. Deployment-server toimii isoimmissa järjestelmissä keskitettynä konfiguraatioiden hallintaohjelmana, joten sitä ei tarvita.



Kuva 10. Raporttipalvelimen Splunk-instanssin komponentit ja käytetyt portit.

Indeksoija on komponentti, joka indeksoi dataa. Tyypillisesti tämä tapahtuu forwarderien kautta, mutta data voidaan indeksoida myös manuaalisesti lataamalla tai määrittää indeksoija monitoroimaan tiettyä datalähdettä. Indeksoija muuttaa saapuvan datan tapahtumiksi indeksiin, joka toimii tiedostomuotoisena säilytyspaikkana datalle. Indeksoija etsii myös indeksoitua dataa vastauksina sille tuleviin hakupyyntöihin. [30.]

Indeksoijia voi olla useita hajautetussa järjestelmässä. Tällöin indeksoijat toimivat parhaiten klustereina varmistaen datan hyvän saatavuuden. Klusterissa toimiville indeksoijille määritellään oma tehtävä: ne indeksoivat saapuvaa dataa, ja tässä yhteydessä pakolliset Search head -komponentit suorittavat hakukyselyjä indekseistä ja tulosten palauttamista käyttäjälle. [30.]

Raportointijärjestelmän Splunk-ympäristössä tarvitaan vain yksi indeksoija, joka hoitaa kaikki edellä mainitut tehtävät: indeksoi dataa, muuttaa datan tapahtumiksi, tallentaa tapahtumat indekseihin, etsii dataa indekseistä vastauksina hakukyselyihin sekä hoitaa hakujen hallinnan.

Forwarder-komponentiksi kutsutaan Splunk-instanssia, joka lähettää dataa toiselle forwarderille tai indeksoijalle. Tyypillisesti nämä komponentit suorittavat Splunkin syötevaihetta, jossa tietovirta edelleen lähetetään indeksoijalle jäsennystä varten. Forwarder toimii ilman käyttöliittymää, ja sitä ohjataan komentotulkin kautta komennoin, jotka noudattavat syntaksia

./splunk <command> [<object>] [[-<parameter>] <value>]...

[31.]

Käytettäviä forwarder-instansseja on kaksi erilaista:

- *Universal Forwarder* on Splunkin kevyt versio, joka sisältää valmiudet ainoastaan tiedon edelleen lähettämiseen. Instanssi ei tarvitse erillistä lisenssiä toimiakseen, vaan asennuspaketti sisältää oman, joka otetaan käyttöön automaattisesti.
- *Heavy Forwarder* pystyy lähettämisen lisäksi tarvittaessa myös indeksoimaan, muuttamaan ja etsimään dataa. Instanssi käyttää Splunk Enterprisen kanssa samaa lisenssiä, johon sille on määriteltävä oikeudet. [31.]

Elisan Tampereen WLAN-laboratorio ja Helsingin testilaboratorio ovat paikkoja, joista Splunk kerää dataa analysoitavaksi forwarderien kautta. Näitä datalähteitä varten päätettiin ottaa käyttöön kevyt Universal Forwarder. Se lähettää näistä lähteistä raakaa jäsentämätöntä dataa indeksoijalle TCP:llä käyttäen Splunkin omaa lähetysprotokollaa.

Forwarderin konfigurointiosuus esitetään luvussa 5.3.4 sen jälkeen, kun on käsitelty olennainen käsite ”Splunk-sovellus” ja sen konfigurointimahdollisuudet.

5.3.3 Splunk-sovellus

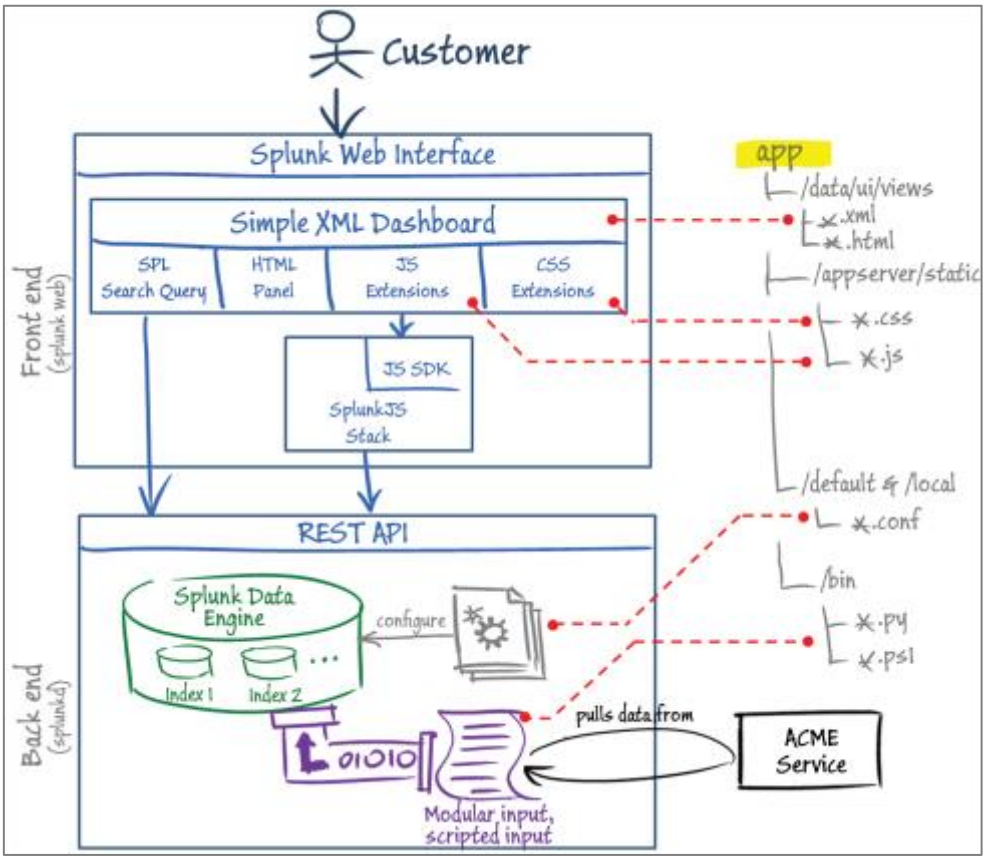
Splunkwebin päällä suoritetaan sovelluksia, jotka ovat käyttäjille avoimen lähdekoodin ohjelmia, sillä lähdekoodi on vapaasti luettavissa ja muokattavissa. Splunk-sovellusten avulla kehittäjät voivat laajentaa Splunkin käyttömahdollisuuksia omiin tarpeisiinsa

sopiviksi. Sovellukset liittyvät visualisointiin, datan analysointiin ja siihen liittyviin toimintoihin. Sovelluksiin kuuluu myös joukko lisäosia (engl. add-ons), jotka liittyvät yleensä eri datatyyppien mahdollistavaan käsittelyyn. Splunkbase-kirjastosta on saatavilla satoja sovelluksia ja lisäosia erilaisiin käyttötarkoituksiin, osa maksullisia, osa ilmaisia. [32.]

Splunk tarjoaa oletuksena käyttäjälle haku-sovelluksen, joka mahdollistaa Splunkin ydinominaisuudet: datahaun Splunkin omalla hakukielellä (SPL), pivot-työkalun hakujen muodostamiseen sekä raporttien, hälytyksien ja työpöytien muodostamisen. Sovellus riitti hyvin vastaamaan järjestelmän tarpeisiin.

Splunk-sovellus on siis jäsennelty sarja konfiguraatioita ja muita määrittelyjä, joita hyödynnetään datan indeksoinnissa ja visualisoinnissa. Kaikkien Splunk-sovellusten konfiguraatiot ja skriptit ovat luettavissa ja muokattavissa, jolloin on mahdollista luoda oma räätälöity instanssi valmiina olevan ohjelman rinnalle. [33.]

Kuvassa 11 selvennetään Splunk-sovelluksen rakentuminen. Sovelluksen front-end-puolen kanssa keskustelee Splunkweb-palvelu, joka tarjoaa interaktiivisen käyttöliittymän Splunkille selaimen kautta. Back-end-puoli keskustelee splunkd:n kanssa, joka on järjestelmän pääprosessi: deamon. Kuvan vieressä on esitetty sovelluksen hakemiston rakennetta ja hakemiston yhteys sovelluksen rakennusosiin. [33.]

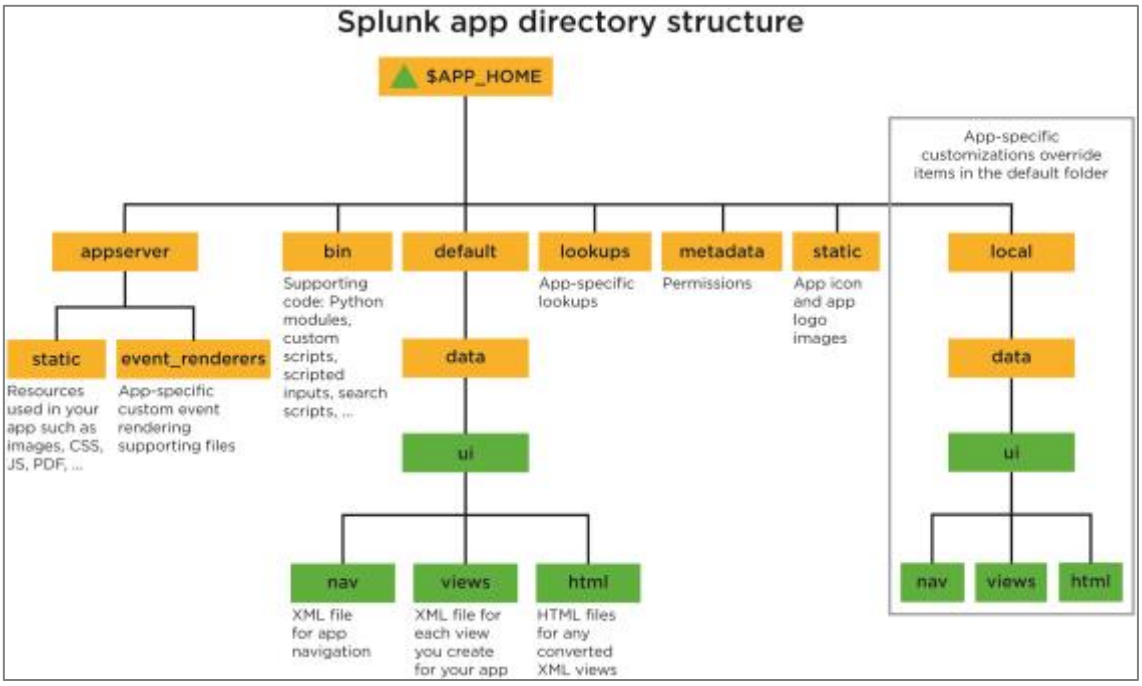


Kuva 11. Splunk-sovelluksen osat [33].

5.3.4 Sovelluksen konfiguraatio

Raportointijärjestelmä käytössä on siis oletuksena tarjottu haku-sovellus, jonka konfigurointi esitetään seuraavaksi. Ennen konfiguroinnin kuvaamista on kuitenkin hyvä tarkemmin vielä esitellä sovelluksen hakemiston rakenne, jonka ymmärtäminen auttaa konfiguroinnin hahmottamisessa.

Kuvassa 12 on esitetty Splunk-sovelluksen hakemiston rakenne, jossa esitetään kaikki pakatut komponentit, myös konfigurointitiedostot päätteellä .conf. Kuten kuvasta nähdään, hakemistosta /local löytyvät kaikki muokattavat kappaleet. Tärkeää onkin tehdä aina kaikki muokkaukset tämän hakemiston alle ja säilyttää /default-hakemiston alla kaikki alkuperäiset konfiguraatiot sovelluksesta.

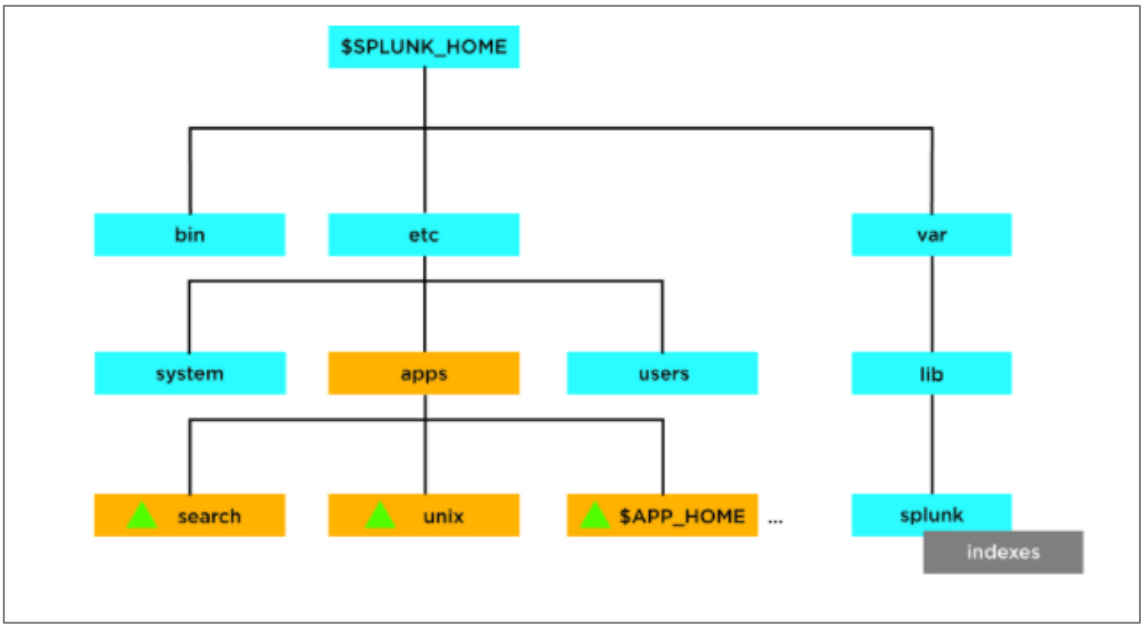


Kuva 12. Splunkin sovelluskohtainen hakemistorakenne [33].

Muutoksia konfigurointitiedostoihin voi tehdä tekstieditorilla, komentotulkilla tai vaihtoehtoisesti splunkwebin kautta, jolloin Splunk-ohjelma ottaa kopiot alkuperäisistä konfigurointitiedostoista, muokkaa niitä ja tallentaa /local-hakemistoon, jos tiedosto ei ole jo valmiina siellä.

Indeksit

Indeksit toimivat datan varastona, johon data säilötään tapahtumatiedoiksi muutettuna. Indeksien tallennuspaikkana toimii /var/lib/splunk-hakemisto. Kuten kuvasta 13 huomataan, tämä hakemisto on järjestelmätasoinen hakemisto eikä sovelluksen sisällä.



Kuva 13. Splunkin ohjelmakohtainen hakemistorakenne [25].

Indeksoitu data siirtyy porrastetusti iän ja koon mukaan vielä eri alihakemistoihin, joita kutsutaan Splunkin terminologiassa bucketeiksi. Nämä bucketit ovat hot, warm, cold, freeze. Viimeisen kohdan vaiheessa data poistetaan lopullisesti indeksistä tai se voidaan vaihtoehtoisesti arkistoida myöhempää käyttöä varten. [34.]

Käyttöön luotiin kolme erillistä indeksiä kullekin datanlähteelle. Indeksit oli helppo luoda suoraan käyttöliittymän kautta, minkä jälkeen ne kirjautuivat indexes.conf-tiedostoon search-sovelluksen /local-hakemistoon. Kuvassa 14 on ote tiedostosta.

```
[viser_ftp]
coldPath = $SPLUNK_DB/viser_ftp/colddb
enableDataIntegrityControl = 0
enableTsidxReduction = 0
homePath = $SPLUNK_DB/viser_ftp/db
maxTotalDataSizeMB = 512000
thawedPath = $SPLUNK_DB/viser_ftp/thaweddb

[wlanlabra]
coldPath = $SPLUNK_DB/wlanlabra/colddb
enableDataIntegrityControl = 0
enableTsidxReduction = 0
homePath = $SPLUNK_DB/wlanlabra/db
maxTotalDataSizeMB = 512000
frozenTimePeriodInSecs = 125798400
thawedPath = $SPLUNK_DB/wlanlabra/thaweddb
```

Kuva 14. Ote indexes.conf-tiedostosta.

Indexes.conf-tiedostossa attribuutti-arvoparit määrittelevät säännöksiä liittyen datan elinkaareen ja säilytykseen. Yksi kappale (engl. stanza) määrittelee aina otsikoidun indeksin asetukset. Kiinnostavin attribuutti liittyy datan elinkaaren viimeiseen vaiheeseen, siihen, kuinka kauan data säilyy indeksoijan käytössä. Attribuutti FrozenTimePeriodInSec määrittelee tämän ja on asetettu oletuksena kuuteen vuoteen. Oletusaika haluttiin pitää mobiilitestituloksissa, mutta WLAN-laboratorion kohdalla säilytysaikaa lyhennetään neljään vuoteen.

Forwarder

Käyttöön valittu Universal Forwarder asennetaan datan lähtevään päähän ja konfiguroidaan toimimaan yhdessä vastaanottavan pään kanssa. Asennustapa ja konfigurointi riippuvat käytettävästä käyttöjärjestelmästä. Raportointijärjestelmän forwarderit asennetaan Windows- ja UNIX-alustoille. Koska Windowsin asennus on hieman yksinkertaisempi, käytetään seuraavaksi esimerkkiä Forwarderin asennuksesta WLAN-laboratorion UNIX-käyttöjärjestelmään, jossa konfigurointi suoritetaan komentotulkkia käyttäen. Windows-puolella voi konfiguroinnin hoitaa ohjelman asennuksen yhteydessä.

Ubuntu-alustalle valittiin .deb-asennuspaketti Splunkin kotisivuilta. Forwarder on asennuksen jälkeen käynnistettävä ja lisenssisopimus hyväksyttävä, ennen kuin konfigurointi voidaan aloittaa. Jokaisen tehdyn muutoksen jälkeen on Forwarder myös käynnistettävä uudelleen, jotta muutokset tulevat voimaan. Komentotulkkiesimerkissä 4 on Forwarderin peruskomennot. Komennoissa on käytetty $SPLUNK_HOME-ympäristömuuttujaa, johon on tallennettu polku Splunkin kotihakemistoon.

```
# cd $SPLUNK_HOME/bin
# ./splunk start
# ./splunk restart
# ./splunk stop
```

Komentotulkkiesimerkki 4. Splunk-Forwarderin peruskomennot.

Latauksen jälkeen aloitettiin konfigurointiosuus. Konfiguroinnissa määritellään, mitä dataa lähetetään ja minne lähetetään. Komentotulkkiesimerkissä 5 on komennot liittyen konfigurointiin. Rivillä 1 asennettiin oikea isäntäpalvelin ja valittiin käytettävä portti 8080.

Yhteyden statuksen voi tarkistaa rivin 2 komennolla. Yhteyden pitäisi olla aktiivinen. Rivin 3 komennolla valittiin, mitä paikallista hakemistoa Forwarder monitoroi, ja annettiin sille jo tässä vaiheessa indeksin nimi. Forwarder kysyi käyttäjältä tämän jälkeen kirjautumistietoja. Oletuksena ovat tunnukset admin/changeme. Kun kirjautuminen on suoritettu onnistuneesti, aloittaa Forwarder datan monitoroinnin välittömästi. Näiden komentojen jälkeen on hyvä taas käynnistää Forwarder uudestaan.

```
1. # ../splunk add forward-server 193.66.8.176:8080
2. # ./splunk list forward-server
3. # ./splunk add monitor /path/to/app/logs/ -index wlanlabra
```

Komentotulkkiesimerkki 5. Splunk-Forwarderin konfigurointi datan lähettäjäksi.

Toinen tapa konfiguroinnille olisi ollut muokata suoraan tarvittavia konfigurointitiedostoja tekstieditorilla. Konfigurointitiedostot ovat Inputs.conf, joka määrittelee miten Forwarder kerää dataa, ja outputs.conf, joka määrittelee miten Forwarder lähettää dataa palvelimelle. [35.]

Inputs.conf-tiedostoon voidaan myös määritellä whitelist- ja blacklist-sääntöjä, joiden avulla kerrotaan indeksoijalle, mitä tiedostoja monitoroidusta hakemistosta ladataan ja mitä ei ladata. Nämä säännöt toimivat itsenäisesti toisista erillään ja käyttävät hyödyksi säännöllisen lausekkeen (engl. regular expression) syntaksia. WLAN-laboratoriossa ei näitä sääntöjä tarvita, mutta testilaboratorion Forwarderin haluttiin lähettävän indeksoijalle ainoastaan CSV-tiedostoja. Seuraava whitelist-sääntö luotiin tähän tarkoitukseen:

```
whitelist = /.csv$
```

Lopuksi, ennenkuin data siirtyy Forwarderista onnistuneesti indeksoijaan, pitää vastaanottava puoli konfiguroida vastaanottamaan dataa valitusta portista. Tämän voi tehdä komentotulkkia käyttäen tai muokkaamalla suoraan inputs.conf-tiedostoa (kuva 15).

```
[default]
host = Splunk-serveri

[splunktcp://8080]
disabled = 0
index=wlanlabra
~
~
~
```

Kuva 15. Splunkin inputs.conf-konfigurointitiedosto.

Näiden toimintojen jälkeen, kun palomuurin porttien uudelleenohjaukset ovat kunnossa, on datan liikkuvuus paikasta toiseen mahdollistettu. Splunkin käyttöliittymästä on mahdollista vielä tarkistaa, että Forwarder-instanssit näkyvät aktiivisina.

Datan käsittely

Props.conf-tiedostossa määritellään muun muassa ohjeita jäsennellyn datan parsimiseen ja attribuutti-arvoparien muodostamiseen. Kappaleiden otsikoissa käytetään lähdetyypin nimeä. Muita mahdollisia otsikoita olisivat lähteen ja isäntäpalvelimen nimi [35]. CSV-tiedostojen suhteen ei tarvinnut määritellä lisäasetuksia datan parsimiseen, mutta JSON-tiedostojen kohdalla oli paikallaan muutama lisäys.

Kuvassa 16 on esitetty JSON-tiedostojen kappale props.conf-tiedostossa. Kappaleen attribuutti-arvomalliksi (KV_MODE) piti vaihtaa JSON. Erilliset JSON-tiedostot käsitellään indeksissä yhtenä monirivisenä tapahtumana.

```
[json-3]
DATETIME_CONFIG = CURRENT
KV_MODE = json
MAX_EVENTS = 10000
NO_BINARY_CHECK = true
SHOULD_LINEMERGE = false
TRUNCATE = 0
disabled = false
```

Kuva 16. Ote Splunkin props.conf-tiedostosta.

Yksi ongelma matkan varrella olivat kadonneet rivit JSON-tiedostoissa indeksoinnin jälkeen. Loppujen lopuksi selvisi, että JSON-tiedoston tapahtumissa oli automaattinen katkaisu 257 rivin jälkeen. Nostamalla MAX_EVENTS-attribuutin arvoa pystyttiin vaikuttamaan siihen, kuinka monta riviä voidaan sisällyttää yhteen tapahtumaan. TRUNCATE-attribuutti taas kuvaa yhden tapahtuman merkkien maksimimäärää. Kun sille annetaan arvo 0, toiminto ei ole käytössä eikä merkkien määrällä tapahtumassa ole väliä. Nämä toimenpiteet ratkaisivat JSON-tiedostojen pilkkoutumisen.

Käyttäjähallinta

Käyttäjien ja roolien luominen ja hallinta onnistuivat helposti splunkwebin kautta. Haku-sovellukseen luotiin kaksi käyttäjää: admin ja user, joille luotiin omat roolit käyttöön. Admin-käyttäjä toimii järjestelmävalvojana ja perii kaikki admin-tason oikeudet. User-käyttäjän tunnuksilla pääse katsomaan kaikkia kolmea luotua dashbordia ja tekemään datahakuja, mutta editointimahdollisuuksia ei ole.

Kuvassa 17 on ote authorize.conf-tiedostosta, jossa määritellään user-roolin oikeudet muun muassa datahakuihin (get_metadata, search), onnistuneeseen kirjautumiseen järjestelmään (rest_properties_get,) ja tiedostojen lisäämiseen syötteeksi tai tulosteeksi (input_file, output_file).

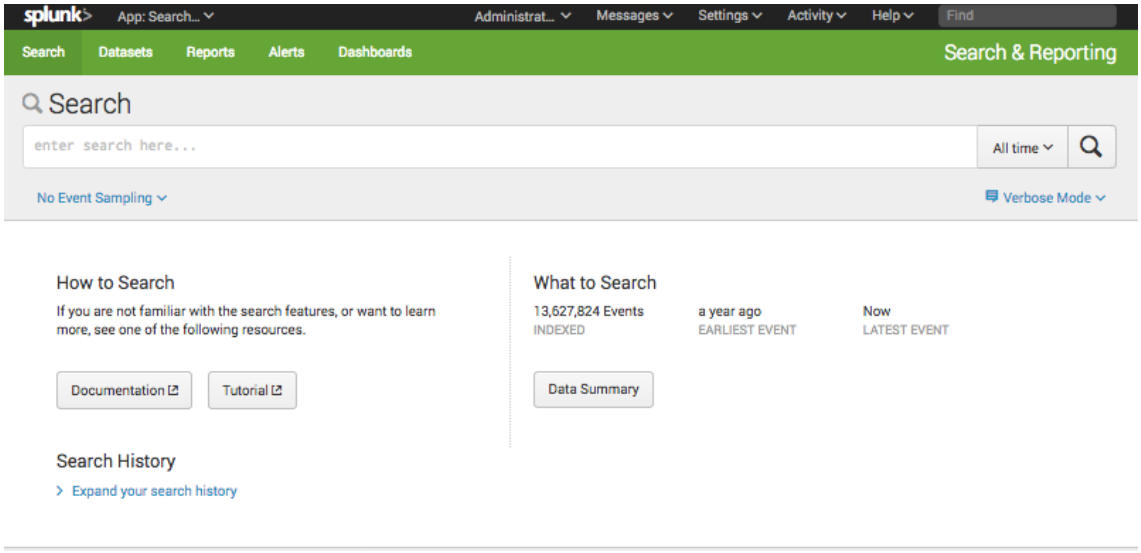
```
[role_user]
# ==== Subsumed roles ====
# ==== Capabilities   ====
change_own_password = enabled
edit_search_schedule_window = enabled
get_metadata        = enabled
get_typeahead       = enabled
input_file          = enabled
list_inputs         = enabled
output_file         = enabled
request_remote_tok  = enabled
rest_apps_view      = enabled
rest_properties_get = enabled
rest_properties_set = enabled
search              = enabled
accelerate_search   = enabled
pattern_detect      = enabled
export_results_is_visible  = enabled
extra_x509_validation = enabled
# ==== Other settings ====
srchIndexesAllowed = *
srchIndexesDefault = main
```

Kuva 17. Ote Splunkin authorize.conf-tiedostosta.

5.3.5 Front-end-puoli

Haku

Tallennettuun ja indeksoituun dataan pääsee käsiksi Splunkin datahaun kautta (kuva 18). Haut kirjoitetaan Splunkin omalla hakukielellä, joka tunnetaan nimellä Search processing language (SPL). SPL-kieltä on helpompi ymmärtää, jos on aikaisempaa kokemusta tietokantojen hakukyselyistä. Vaikka terminologia komennoissa eroaa suuresti, on konsepti melko samankaltainen.

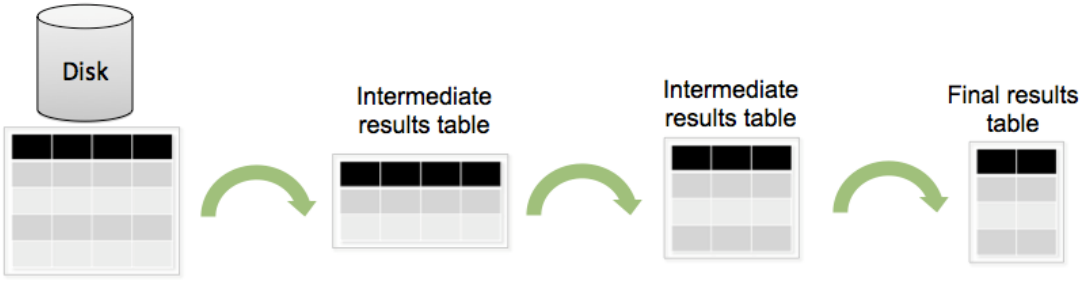


Kuva 18. Splunkin haku-sovelluksen hakunäkymä.

Hakukomennot kertovat Splunkille, mitä tehdä indeksiin tallennettujen tapahtumien kanssa. Esimerkkinä toimista voisi olla turhan tiedon suodattaminen, lisätiedon poiminen, statistiikan laskeminen tai vaikka tulosten uudelleenjärjestys. Tarpeita on yhtä monia kuin järjestelmiä.

Hakukyselyt kirjoitetaan käyttöliittymän hakupalkkiin, johon voi halutessaan myös määritellä aikarajauksen, jolloin haku suoritetaan tietyn ajanjakson sisällä tulleista tuloksista. Haut koostuvat komentojen ja argumenttien sarjoista, jotka liitetään toisiinsa käyttäen (|)-merkkiä (engl. pipe). Tarkoituksena on kirjoittaa mahdollisimman tarkkoja hakuja, jotka antavat juuri sen datan, mitä tarvitaan. Näin haut eivät myöskään kuormita ympäristöä liikaa.

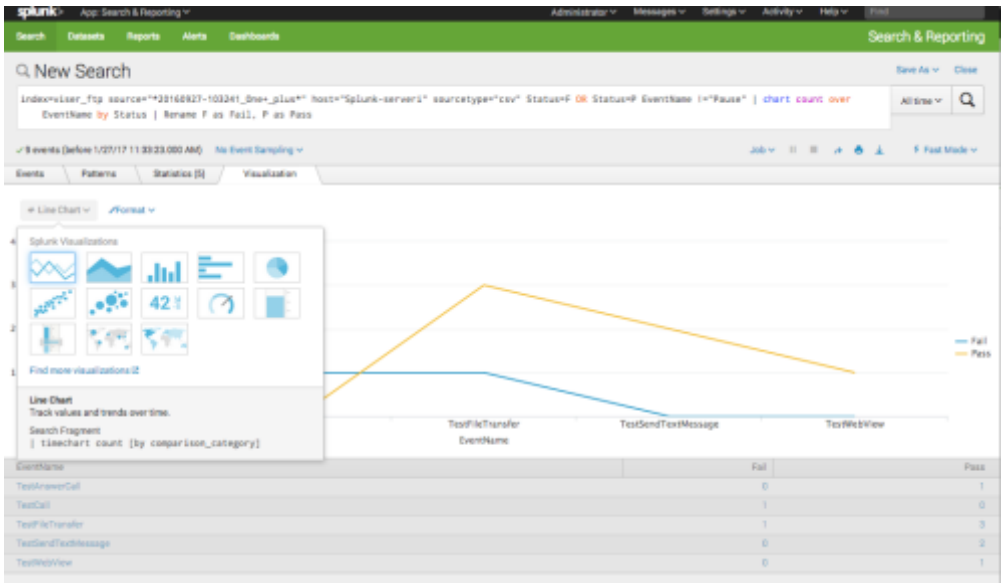
Haut aloitetaan määrittelemällä, mitä tapahtumia palautetaan indekseistä. Hakuehtoina käytetään avainsanoja, boolean-lausekkeita, kenttien nimiä, vertailulausekkeita tai wildcard-lausekkeita. Tämän jälkeen putkitetuilla komennoilla suodatetaan tuloksia kohti lopullista päämäärää (kuva 19). [36.]



Kuva 19. Hakukyselyjen rakenne [27].

Visualisointi

Visualisointiin päästään Splunkin haku-sovelluksen hakujen visualisointivälilehdeltä (kuva 20). Visualisointi edellyttää, että hakujen tulokset ovat tietynlaisessa formaatissa ja datan rakenteet ovat oikeanlaiset.



Kuva 20. Splunk haku -sovelluksen visualisointinäkymä.

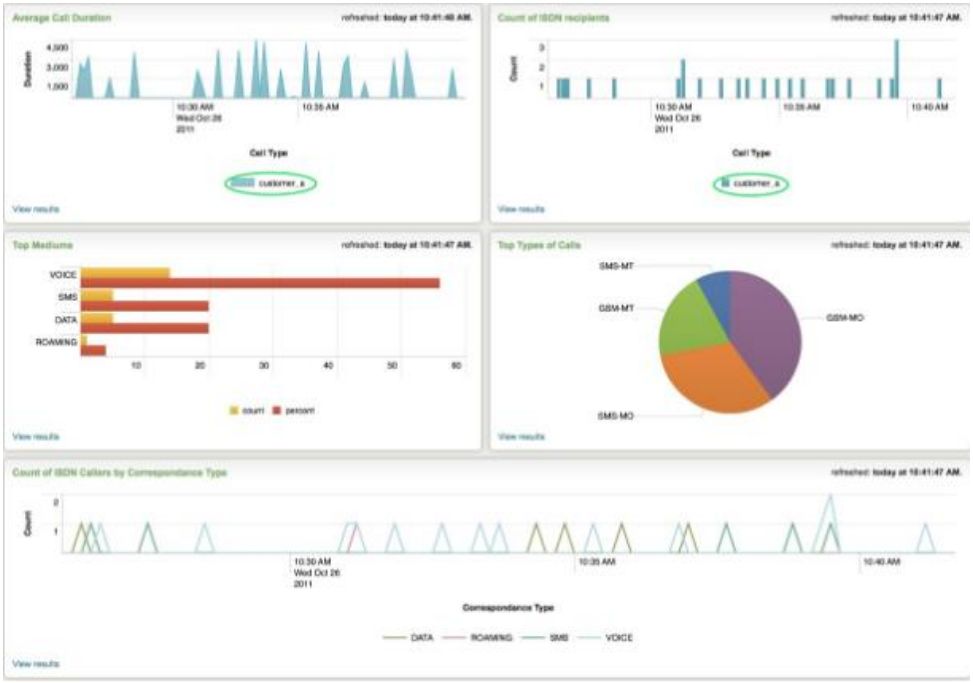
Käyttämällä oikeita statistiikkaan liittyviä komentoja, kuten stats, chart tai timechart, saadaan visualisointi mahdollistettua. Ennen visualisointia on hyvä tarkistaa statistiikkataulukosta, että tuloskentät on luotu oikein. Statistiikan taulukoiden sarakkeiden järjestys ja lukumäärä näyttävät datan rakenteen, joka haulla on saatu luotua. [37.]

Visualisointimalleja on käytettävissä monia, ja niillä on omat vaatimukset siitä, miten datan pitää olla järjestäytynyt. Esimerkiksi kaaviot vaativat vähintään kaksi saraketta, jolloin ensimmäinen sarake vastaa x-akselin arvoja ja toinen y-akselin arvoja. Kun lisättään sarakkeita enemmän, Splunk vertailee niitä keskenään y-akselilla. [37.]

Splunkin tarjoamat esitystavat visualisoinnille ovat pylväsdiagrammi, viivadiagrammi, taulukkonäkymä, ympyrädiagrammi, palkkidiagrammi, hajontakaavio ja arvoa kuvaava mittaridiagrammi [37].

Dashboardit

Dashboardit ovat ikään kuin työpöytiä, joihin tallennetut ja visualisoidut haut kerätään paneeleina (kuva 21). Dashboardin rakentamiseen on mahdollista käyttää useampaa työkalua tai ohjelmistokehystä. Yksinkertaisin tapa luoda ja muokata dashboardeja on käyttää Splunkin tarjoamia syötteitä hakujen käsittelyyn. Tämän rinnalla on kuitenkin hyvä käyttää yksinkertaisen XML-lähdekoodin muokkaamista, jonka käyttöliittymän editori mahdollistaa. [29.]



Kuva 21. Spunkin haku -sovelluksen dashboard-näkymä [38].

Edistyneeseen käyttöön on myös mahdollista käyttää CSS- ja JavaScript-koodia hyödyksi. Tarjolla on myös ohjelmistokehyksiä, joilla toimintaa pystyy laajentamaan entisestään. Esimerkiksi JavaScript-ympäristössä työskentelyyn on tarjolla SplunkJS Stack -komponentti, joka sisältää työkalut jQuery, RequireJS ja Backbone.js. XML-lähdekoodi on myös mahdollista kääntää HTML-muotoon.

Omat dashboardit

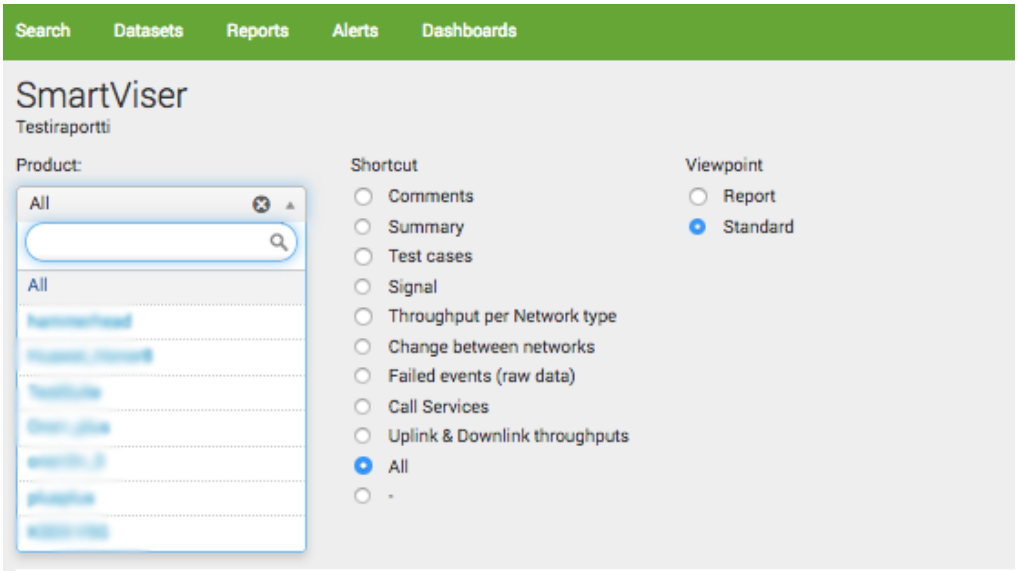
Omat dashboardit on rakennettu lähinnä XML-koodin editorilla. Dashboardeja on rakennettu kaksi: WLAN-laboratorion testituloksille ja mobiilipuolen testituloksille. Dashboardien päätavoitteena oli helpottaa ja nopeuttaa datan tulkitsemista ja mahdollistaa poikkeavuuksien ja yhteyksien havaitseminen. Käyttäjälle haluttiin myös tarjota mahdollisuus rajata näkymää tai porautua tietoon syvemmin.

Dasboardeissa käytetyt kuvaajat valittiin aina kulloiseen tarkoitukseen sopivaksi nojautuen aikaisemmin esitettyyn teoriaan kuvaajien käytöstä. Kuvaajien valinnan jälkeen keskityttiin siihen, että kuvaaja esittää datan selkeästi ja totuudenmukaisesti. Kuvassa 22 on muutamia käytettyjä kuvaajia.

Kuva 22. Muutamia Splunk-työpöydillä käytettyjä kaavioita.

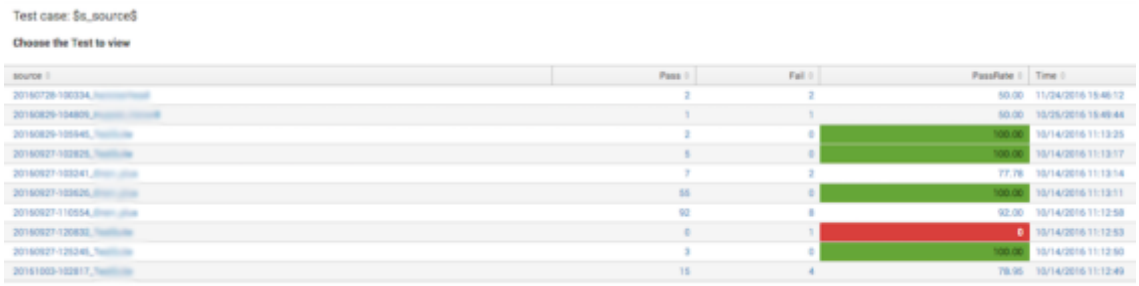
Ympyrädiagrammi esittää onnistuneiden ja epäonnistuneiden testiajojen prosenttiosuudet. Havainnointia helpottamaan kuvaajassa on käytetty statukseen sidottua väriä. Punaisen ja vihreän vastaväriharmonia toimii parhaiten, sillä se on yleisesti käytetty väripari kuvaamaan testien statusta. Kuvaajan haluttiin antavan nopeasti katsojalle yleiskuvan tuloksista. Eri testiajojen läpimenon vertailuun käytettiin siihen sopivaa yksinkertaista summapylväsdiagrammia. Määräasteikko aloitettiin nollasta, ja pylväistä tehtiin tasaleveitä, aikaisemmin mainittujen ohjeiden mukaisesti. Viivadiagrammilla haluttiin kuvata sille sopivaa trendiä signaalin muutoksesta testiajon aikana. X-akselilla valittiin aika ja y-akselille tarkasteltava signaalin voimakkuus. Kuvaajaan haluttiin myös ottaa vertailuun keskiarvosignaalia kuvaava viiva, jolloin on helpompi tarkastella signaalin kehityssuuntaa.

Dashboardien käyttäjillä saattaa olla erilaiset tarpeet ja mielenkiinnon kohteet tarkasteltaessa testituloksia. Tämän vuoksi dashboardeihin lisättiin vuorovaikutuksellisuutta, jolloin ne palvelevat suurempaa yleisöä. Ensimmäinen vastaantuleva vuorovaikutuksellinen elementti on navigointi valitun laitteen testituloksiin. Navigointi tapahtuu mobiilipuolen dashboardilla pudotusvalikon ja radionappien avulla. Pudotusvalikkoon on sisällytetty kaikki testatut laitteet. Koska laitteiden määrä on suuri, on navigointia helpotettu hakukentän avulla. Radionappien avulla taas pystyy valikoimaan tarkasteltavat osa-alueet ja oikeanlaisen näkymän. Radionappeihin päädyttiin, sillä vaihtoehtojen määrä oli suhteellisen pieni. (Kuva 23.)



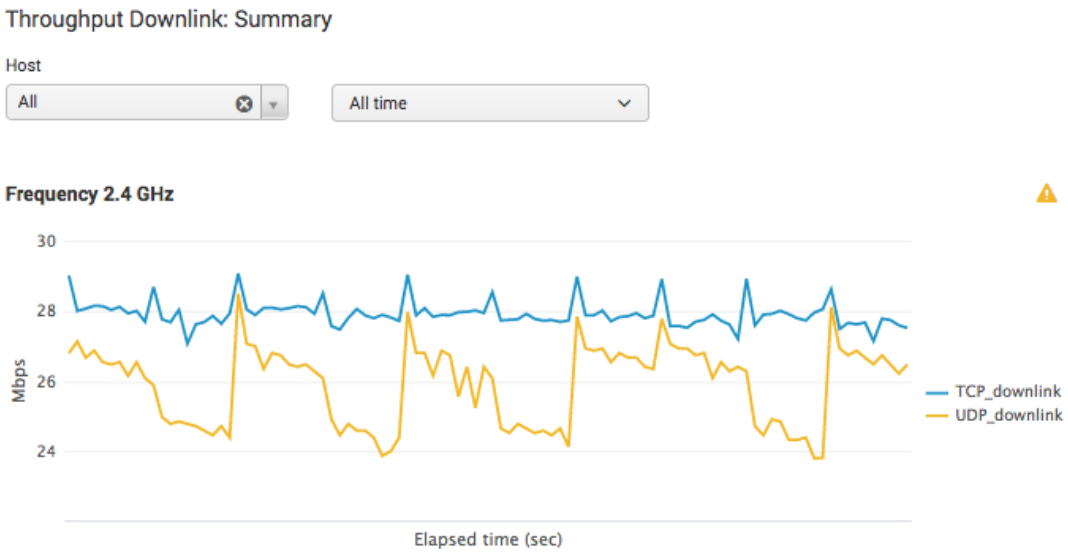
Kuva 23. Splunk-työpöytään rakennettu navigointimalli.

Valintakontrollien tuloksena nähdään laitteet listattuna taulukkomuodossa (kuva 24). Tuloksissa on käytetty korostusvärejä, joilla halutaan siirtää katsojan huomio laitteisiin, joiden kaikki testit ovat joko epäonnistuneet tai vastaavasti onnistuneet täysin. Näitä samoja korostusvärejä käytetään johdonmukaisesti muissakin kuvaajissa, jotta katsojan on helppo tarkastella tuloksia. Taustan ja kuvioiden kontrastiero on haluttu pitää myös jokaisessa paneelissa suurena, jotta hahmottaminen helpottuisi.



Kuva 24. Testatut laitteet Splunk-työpöydällä taulukkomuodossa.

Toinen käytetty vuorovaikutuksellinen elementti on suodatin, jolla käyttäjä pystyy tarkentamaan omaa valittua näkökulmaa entisestään. Yksi esimerkki tästä on WLAN-laboratorion dashboardilla käytetty siirtonopeuden viivadiagrammi (kuva 25). Käyttäjä voi valita tarkemmin suodattimilla, minkä isäntäkoneen tuloksia tarkastellaan ja millä aikavälillä. Kaikissa viivadiagrammeissa on myös mahdollisuus zoomata hiirellä tiettyä kohtaa ja ottaa aikaväli lähempään tarkasteluun.



Kuva 25. Splunk-työpöydällä käytetty kaavio, jossa on suodattimia.

Käyttäjän on mahdollista myös tallentaa valitut tulokset PDF-tiedostona. Tätä ajatellen on dashboardeihin lisätty tekstikenttä vapaavalintaiselle kommentille, johon esimerkiksi testaaja voi kirjoittaa omia huomioita, joita ei välttämättä tuloksista käy ilmi.

5.4 Raportointijärjestelmän käyttöönotto ja käyttäjäpalaute

Raportointijärjestelmällä on mobiilipuolen testauksessa tällä hetkellä yksi aktiivinen käyttäjä, joka käyttää raportointijärjestelmää viikoittain tarkastellessaan viSerilla ajettujen testien tuloksia. Koska WLAN-laboratorio on vielä kehitysvaiheessa, on raportointijärjestelmä myös siellä kehittäjän esikäytössä. WLAN-laboratorion kehittäjän kanssa on käyty projektin aikana alusta alkaen keskustelua siitä, miten raportointijärjestelmä ja WLAN-laboratorio toimivat yhdessä. Sieltä saatu palaute on ollut arvokasta työn etenemisen kannalta. Seuraavaksi esitetään mobiilipuolen testaajan käyttäjäpalaute raportointijärjestelmän käytettävyydestä.

Testatun laitteen tulokset löytyvät testaajan mukaan helposti Splunkin käyttöliittymän kautta. Käyttöliittymään on rakennettu vetovalikko ja hakukenttä, joilla etsiä tuloksia. Uusimmat tulokset näkyvät automaattisesti käyttäjälle ensimmäisessä taulukkonäkymässä. Suuri etu on myös testaajan mukaan siinä, että tulosten tarkastelu ei ole laiteriippuvainen, vaan verkkokäyttöliittymään on mahdollista päästä kiinni miltä laitteelta tahansa aina selaimen kautta.

Visuaalinen ulkoasu helpottaa testaajan mukaan tulosten sisäistämistä nopeasti. Käyttöliittymässä on käytetty hyödyksi värejä, jotka on sidottu tiettyyn tuloksen statukseen, sekä informatiivista otsikointia kuvaajissa. Käyttöliittymässä on myös käytetty aina tilanteeseen sopivia kuvaajia esittämään tieto parhaalla mahdollisella tavalla.

Testaajan mukaan tulevaisuudessa testiajoja lisätään varmasti nykyisestä. Datan määrän noustessa on myös helppo lisätä kapasiteettia konesaliin, jotta suorituskyky pysyy hyvänä. Testaajan mukaan kiinnostusta olisi myös päästä vertailemaan kahden tai useamman laitteen tuloksia rinnakkain. Tämä on kehitysajatus, joka on helposti toteutettavissa Splunkilla käyttäen Splunkin omia syöttö-elementtejä ja alkio-muuttujia tai suoraan JavaScriptiä. Testaaja kertoo järjestelmän olevan nyt toistaiseksi käytössä, ja jatkokäyttöä ja kehitystä suunnitellaan.

6 Splunk- ja ELK-ohjelmistot vertailussa

Vertailuun Splunkin kanssa otettiin vastaavanlainen avoimen lähdekoodin lokienhallintajärjestelmä, joka on tällä hetkellä Splunkin ohessa eniten käytetty lokien hallinnassa. Järjestelmä on ELK, joka niputtaa yhteen kolme erillistä ohjelmaa: Elasticsearch, Logstash ja Kibana. Kuvassa 26 on tehty vertailua järjestelmien kiinnostavuudesta käyttämällä mittarina Googlen hakumääriä. ELK vie voiton nykyhetken tilanteessa, ja kiinnostus siihen on selvästi noususuunnassa.

Kuva 26. Google Trend -työkalulla tehty vertailu Splunk- ja ELK-ohjelmiston kiinnostavuudesta. ELK:lla on ollut nopea nousu viime vuosina, ja se on mennyt Splunkin ohi hakumäärissä.

ELK – Elasticsearch, Logstash, Kibana

ELK niputtaa yhteen kolme erillistä avoimen lähdekoodin projektia: Elasticsearch – lokien hakuun, Logstash – tietoputki datan keräämiseen sekä Kibana – datan visualisointiin. Tämä kokonaisuus muodostaa saman järjestelmän, kuin Splunk tarjoaa.

Elasticsearch on avoimeen lähdekoodiin perustuva hakupalvelin. Sen on kehittänyt Shay Banon vuonna 2010. Se on kirjoitettu Java-ohjelmointikielellä, ja se perustuu Apache Lucene -hakusovellukseen. Ohjelma hoitaa datan analysoinnin ja luo datavaraston. Kibana taas luo hakuja datavarastosta ja esittää hakujen tulokset käyttäjälle

käyttöliittymän dashboardeilla. Logstash puolestaan kerää loki- ja tapahtumatietoja eri järjestelmistä ja hoitaa myös parsimisen yhtenäisemmäksi ennen tiedon tallentamista. [39.]

Vertailu Splunkiin

ELK:ssa ja Splunkissa on paljon yhtäläisyyksiä, mutta Splunk on kuitenkin monipuolisin lokienhallintatyökalu. Se tarjoaa toiminnallisuuksia, joita peruskäyttäjä ei todennäköisesti koskaan edes tarvitse. Avoimen lähdekoodin ansiosta ELK antaa kuitenkin enemmän vapautta työkalun käyttämiseen omassa kehityksessä. ELK tarjoaa myös hyvän mahdollisuuden eri formaateissa olevan datan keskittämiseen ja yhdistämiseen.

Taulukkoon 7 on kerätty muutamia eroja järjestelmien välillä. Vertailtavat ominaisuudet on valittu raportointijärjestelmän käyttötarpeiden mukaan ja huomiotta on jätetty raportointijärjestelmälle epäolennaiset asiat.

Taulukko 7. Splunkin ja ELK:n erot.

Ominaisuus	**Splunk**	**ELK**
Datan haku	SPL	Rajallinen
Säilötty data tiivistetään	Kyllä	Ei
Käyttövalmius	Valmis tuote	3 erillistä projektia
Hinta	Maksullinen	Ilmainen
Liitännäiset	Noin 600 kappaletta	Muutamia satoja
Asennus	On-premises	Avoin lähdekoodi

Hakujen tekeminen on joustavampaa Splunkilla, koska sillä on käytössä oma hakukieli SPL. ELK:n haut perustuvat Apache Lucenen hakujen syntaksiin. Datan tuominen järjestelmään on Splunkilla myös hieman helpompaa, sillä Splunkin graafinen käyttöliittymä on erittäin käyttäjäystävällinen. ELK:lla käyttäjän pitää konfiguroida oikeudet ennen datan tuomista indeksoitavaksi.

ELK:ta on huomattavasti monimutkaisempi käyttää kuin Splunkia, sillä järjestelmä tarvitsee kolme sovellusta toimiakseen Splunkin tavalla. Vertailtaessa käytettävyyttä on Splunkilla myös ominaisuuksia, jotka puuttuvat ELK:lta. Näitä ovat dashboardien helppo

visualisoinnin muuttaminen, dashboardien hallinta ja niiden käyttäjähallinta sekä dashboardien vieminen PDF-muotoon.

ELK on maksuton, toisin kuin Splunk, jonka hinnoittelu perustuu lokidatan määrään. Kustannuksia voi ELK:n käytössä tulla kuitenkin laajaan infrastruktuuriin: laitteisto, kapasiteetti ja asiantuntipalvelut [41]. ELK:lla on todella aktiivinen käyttäjä- ja kehittäjäyhteisö internetissä, josta apua on helppo saada.

Pääosin kyse on kuitenkin avoimen lähdekoodin ja maksullisen ohjelman vertailusta, sillä se lienee suurin ero. Vaikka ELK on ilmainen, piilokustannuksia sen käytöstä voi kuitenkin tulla. Splunk tarjoaa vielä toistaiseksi myös monipuolisemmat käyttömahdollisuudet: satoja sovelluksia laajentamaan toimintaa. ELK tulee perässä, mutta ei ole vielä samassa pisteessä. Pienemmät infrastruktuurit, joissa halutaan keskittyä vain lokienhallintaan näyttävillä dashboardeilla, sopivat hyvin ELK:n käyttöön. ELK vaikuttaa kokeilemisen arvoiselta. ELK:n Kibanasta on esimerkkikuvat liitteessä 5.

7 Yhteenveto

Insinöörityön tavoitteena oli rakentaa Elisa Oyj:lle reaaliaikainen raportointijärjestelmä yhteen paikkaan, jonne päätelaitetestauksen testitulokset kulkeutuvat automaattisesti testiajojen jälkeen. Testitulokset haluttiin näyttää käyttäjälle visuaalisina ja informatiivisina raportteina.

Työn tuloksena syntyi toimiva raportointijärjestelmä, joka on rakennettu virtuaaliseen konesaliin käyttäen Elisan omaa tuotetta Elisa eSalia. Lokienkäsittelyn ja käyttöliittymän testiraporteille tarjoaa kaupallinen tuote Splunk Enterprise, jonka ympäristö integroitiin toimimaan virtuaalisalin kanssa. Visuaaliset testitulokset ovat nähtävillä Splunkin kautta kaikkialta verkkokäyttöliittymän kautta. Verkkokäyttöliittymään luotiin kahdet eritasoiset tunnukset.

Raportointijärjestelmällä haluttiin korjata ongelmat, jotka liittyivät testitulosten yhtenäisen säilytyspaikan puuttumiseen ja raporttien manuaalisen rakentamiseen. Näillä korjauksilla haluttiin helpottaa ja nopeuttaa päätöstä testattujen laitteiden hyväksymisestä ja käyttöönotosta. Raportointijärjestelmän automaatio saatiin nostettua halutulle tasolle, ja testitulokset kulkeutuvat ajojen jälkeen automaattisesti parsittaviksi ja visuaalisiksi

raporteiksi käyttöliittymään. Raporteista on selkeästi havaittavissa testiajojen tulokset ja mahdolliset epäkohdat.

Ainoa asia, joka jäi toteuttamatta, oli PDF-raporttien lähettäminen järjestelmästä sähköpostitse asianomaisille aina testiajojen jälkeen. Tässä toiminnossa vastaan tulivat Splunk-ohjelmiston rajoitukset.

Insinöörityössä haluttiin myös perehtyä avoimen lähdekoodin ohjelmakokonaisuus ELK:iin, sillä sen kokeilemista laitedatan esittämiseen mietittiin tulevaisuudessa. Insinöörityössä tekemäni selvityksen perusteella voin ehdottomasti suositella ohjelman käyttöä pienimuotoisissa infrastruktuureissa. Ohjelmakokonaisuus pystyy täysin samaan kuin Splunk, kun kyse on laitedatan esittämisestä visuaalisesti.

Insinöörityö onnistui tuottamaan vaatimusten mukaisen raportointijärjestelmän. Työ oli kokonaisuudessaan erittäin opettavainen prosessi, ja sen avulla pääsin tutustumaan itselleni vieraisiin aihealueisiin, kuten virtuaalisalien toimintaan, verkkoarkkitehtuuriin, tietoturva-asioihin ja järjestelmäintegraatioihin. Suurin tietotaito kertyi kuitenkin Splunk-ohjelmistosta, ja sitä toivon pääseväni vielä hyödyntämään jatkossa.

Lähteet

1 Elisa eSali FAQ. 2017. Verkkodokumentti. Elisa. Saatavilla: <https://admin.ecloud.fi/uniquesiga929d7fed9e979684414487aa62ac37a/uniquesig0/wiki/index.php/FAQ>. Luettu 1.1.2017.

2 Splunk Enterprise Overview. 2017. Verkkodokumenti. Splunk. Saatavilla: <http://docs.splunk.com/Documentation/Splunk/6.5.2/Overview/AboutSplunkEnterprise>. Luettu 29.12.2016.

3 Splunk Installation Manual: System Requirements. 2017. Verkkodokumentti. Splunk. Saatavilla: <http://docs.splunk.com/Documentation/Splunk/6.5.2/Installation/SystemRequirements>. Luettu 29.12.2016.

4 Orloff, J. 2014. Hardening the Linux server. Verkkodokumentti. Ibm. Saatavilla: <http://www.ibm.com/developerworks/linux/tutorials/l-harden-server/>. Luettu 2.12.2016.

5 Capacity Planning Manual: Performance Recommendations. 2016. Verkkodokumentti. Splunk. Saatavilla:< http://docs.splunk.com/Documentation/Splunk/6.5.2/Capacity/Summaryofperformancerecommendations>. Luettu 5.1.2017.

6 Deploying Splunk Enterprice Inside Virtual Environment. 2016. Verkkodokumentti. Splunk. Saatavilla:< https://www.splunk.com/web_assets/pdfs/secure/Splunk_and_VMware_VMs_Tech_Brief.pdf>. Luettu 5.1.2017.

7 Overview of VMware Tools. 2016. Verkkodokumentti. Vmware. Saatavilla: <https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=340>. Luettu 11.1.2017.

8 Getting Data In. 2016. Verkkodokumentti. Splunk. Saatavilla: <http://docs.splunk.com/Documentation/Splunk/6.5.2/Data/WhatSplunkcanmonitor#How_do_I_get_data_in.3F>. Luettu 11.1.2017.

9 viSer. 2016. Verkkodokumentti. SmartViser. Saatavilla: <http://www.smartviser.com/technology/viser>. Luettu 11.12.2016.

10 Heino, Petteri. 2010. Pilvipalvelut. Helsinki: Talentum Media.

11 Elisa eSli-webinaari. 2015. Verkkovideo. Elisa. Saatavilla: <https://www.youtube.com/watch?v=FPCZpcPD4JA>. Katsottu 27.1.2017.

12 Bond, James. 2015. The Enterprise Cloud. O'Reilly Media. Saatavilla: <https://www.safaribooksonline.com/library/view/the-enterprise-cloud/9781491907832/>. Luettu 11.12.2016.

13 Koponen, Juuso. 2012. Informaatiomuotoilu tekee tiedon näkyväksi. Verkkodokumentti. Informaatiomuotoilu. Saatavilla <http://informaatiomuotoilu.fi/asiasanat/informaatiomuotoilun-tehtavat/> Luettu 1.3.2017.

14 Aho, Mika. 2014. Visualisointi ja tiedon jakamisen käytännöt. Verkkodokumentti. Saatavilla: <https://www.slideshare.net/mikaaho/visualisointi-ja-tiedon-jakamisen-kytnnt>. Luettu 1.2.2017.

15 Minkkinen, Mika. 2013. Infografiikan uudet muodot. Opinnäytetyö. Lahden ammattikorkeakoulu, Muotoilu- ja taideinstituutti. Saatavilla: <https://www.theseus.fi/bitstream/handle/10024/57119/minkkinen_mika.pdf?sequence=2>. Luettu 2.2.2017.

16 Best practices: Maximum elements for different visualization types. 2012. Verkkodokumentti. ScribbleLive. Saatavilla: <http://www.scribblelive.com/blog/2012/03/29/maximum-elements-for-visualization-types/>. Luettu 12.1.2017.

17 Farmer, Jesse. 2008. Politics and Tufte's lie factor. Verkkodokumentti. Saatavilla: http://20bits.com/article/politics-and-tuftes-lie-factor. Luettu 3.2.2017.

18 Kuusela, Vesa. 2000. Tilastografiikan perusteet. Edita: Helsinki.

19 Cooper A, Reinmann R, Cronin D, Noessel C. 2014. About face : the essentials of interaction design. Indianapolis: Wiley.

20 Dropdowns, Checkboxes or Radio Buttons? A Quick Guide to Displaying Product Options. 2012. Verkkodokumentti. Volusion. Saatavilla: <https://www.volusion.com/ecommerce-blog/articles/dropdowns-checkboxes-or-radio-buttons-a-quick-guide-to-displaying-product-options/>. Luettu 12.1.2017.

21 Kuutti, Wille. 2011. Linux-käsikirja. WSOYpro.

22 Tietoturvajärjestelmät: Palomuuri. 2016. Verkkodokumentti. Okol. Saatavilla: <http://www.okol.org/verkkokurssit/datanomi/tietojarjestelmien_kehittaminen/tietoturvajarjestelmat/palomuurit/palomuurit.htm>. Luettu 1.12.2016.

23 SSH. 2016. Verkkodokumentti. Linux, wiki. Saatavilla: <https://linux.fi/wiki/SSH>. Luettu 1.11.2016.

24 FTP. 2016. Verkkodokumentti. Linux, wiki. Saatavilla: <https://linux.fi/wiki/FTP>. Luettu 1.11.2016.

25 Splunk Company overview. 2013. Verkkodokumentti. Splunk. Saatavilla: <http://www.splunk.com/web_assets/pdfs/secure/Splunk_Company_Overview.pdf>. Luettu 27.1.2017.

26 Splunk Admin Manual: About Splunk Free. 2016. Vekkodokumentti. Splunk. Saatavilla: <http://docs.splunk.com/Documentation/Splunk/6.5.2/Admin/MoreaboutSplunkFree>. Luettu 26.1.2017.

27 Splunk Installation: Linux. 2016. Verkkodokumentti. Saatavilla: <http://docs.splunk.com/Documentation/Splunk/6.5.2/Installation/InstallonLinux>. Luettu 20.1.12017.

28 Splunk Admin Manual: How Splunk licensing works. Verkkodokumentti. 2016. Saatavilla: <http://docs.splunk.com/Documentation/Splunk/6.5.2/Admin/HowSplunklicensingworks>. Luettu 20.1.2017.

29 Splunk Distributed Deployment Manual. 2016. Verkkodokumentti. Splunk. Saatavilla: <http://docs.splunk.com/Documentation/Splunk/6.5.2/Deploy/Distributedoverview>. Luettu 19.1.2017.

30 Managing Indexers and Cluster of Indexers: Configure Index storage. 2016. Verkkodokumentti. Splunk. Saatavilla: <http://docs.splunk.com/Documentation/Splunk/6.5.2/Indexer/Configureindexstorage>. Luettu 1.11.2016.

31 Splunk Forwarding Data. 2016. Verkkodokumentti. Splunk. Saatavilla: <http://docs.splunk.com/Documentation/Splunk/6.5.2/Forwarding/Aboutforwardingandreceivingdata>. Luettu 19.1.2017.

32 Splunk Admin Manual: Whats An App. Verkkodokumentti. 2016. Saatavilla: <http://docs.splunk.com/Documentation/Splunk/6.5.2/Admin/Whatsanapp>. Luettu 11.11.2017.

33 Splunk Developer Guide. Verkkodokumentti. 2016. Splunk>dev. Saatavilla: http://dev.splunk.com/view/dev-guide/SP-CAAAE2R. Luettu 11.12.2016

34 Kumar A & Yadav T. 2016. Advanced Splunk. Packt Publishing. Saatavilla: <https://www.safaribooksonline.com/library/view/advanced-splunk/9781785884351/ch10s08.html>. Luettu 11.12.2016

35 Splunk Admin Manual: List configuration files. 2016. Verkkodokumentti. Splunk. Saatavilla: < http://docs.splunk.com/Documentation/Splunk/6.5.2/Admin/Listofconfigurationfiles>. Luettu 11.1.2017.

36 Splunk Search Manual. 2016. Verkkodokumentti. Splunk. Saatavilla: <http://docs.splunk.com/Documentation/Splunk/6.5.2/Search/GetstartedwithSearch>. Luettu 22.1.2017.

37 Splunk Dashboards and Visualizations. 2016. Verkkodokumentti. Splunk. Saatavilla: <http://docs.splunk.com/Documentation/Splunk/latest/Viz/Visualizationreference>. Luettu 22.1.2017.

38 Splunk and Multitenancy. 2012. Verkkodokumentti. Splunk Blogs. Saatavilla: <http://blogs.splunk.com/2011/11/04/splunk-and-multitenancy/>. Luettu 22.1.2017.

39 Getting Started with Logstash. 2016. Verkkodokumentti. Elastic. Saatavilla: <https://www.elastic.co/webinars/getting-started-logstash?elektra=home&iesrc=ctr>. Luettu 20.1.2017.

40 Elastic: Hardware. 2016. Verkkodokumentti. Elastic. Saatavilla: <https://www.elastic.co/guide/en/elasticsearch/guide/current/hardware.html>. Luettu 19.1.2017.

41 Splunk vs. ELK. 2016. Verkkodokumentti. Upguard. Saatavilla: <https://www.upguard.com/articles/splunk-vs-elk>. Luettu 19.1.2017.

42 Kibana: Dashboards. 2016. Verkkodokumentti. Elastic. Saatavilla: <https://www.elastic.co/guide/en/kibana/4.5/dashboard.html>. Luettu 19.1.2017.

43 Kibana: Discover. 2016. Verkkodokumentti. Elastic. Saatavilla: <https://www.elastic.co/guide/en/kibana/4.5/discover.html>.Luettu 20.1.2017.

Splunk-järjestelmävaatimukset

Unix operating systems

Operating system	Architecture	Enterprise	Free	Trial	Universal Forwarder
Solaris 10 and 11	x86 (64-bit)	D	D	D	✔
	SPARC				✔
Linux, kernel version 2.6 and later	x86 (64-bit)	✔	✔	✔	✔
	x86 (32-bit)				D
Linux, kernel version 3.x and later	x86 (64-bit)	✔	✔	✔	✔
	x86 (32-bit)				D
PowerLinux, kernel version 2.6 and later	PowerPC				✔
zLinux, 2.6 and later	s390x				✔
FreeBSD 9	x86 (64-bit)				✔
FreeBSD 10	x86 (64-bit)				✔
Mac OS X 10.10 and 10.11	Intel		✔	✔	✔
AIX 7.1 and 7.2	PowerPC				✔
AIX 6.1	PowerPC				D
HP/UX† 11i v3	Itanium				✔
ARM Linux	ARM				A

Kuva 1. Splunkin järjestelmävaatimukset Unix-alustoille.

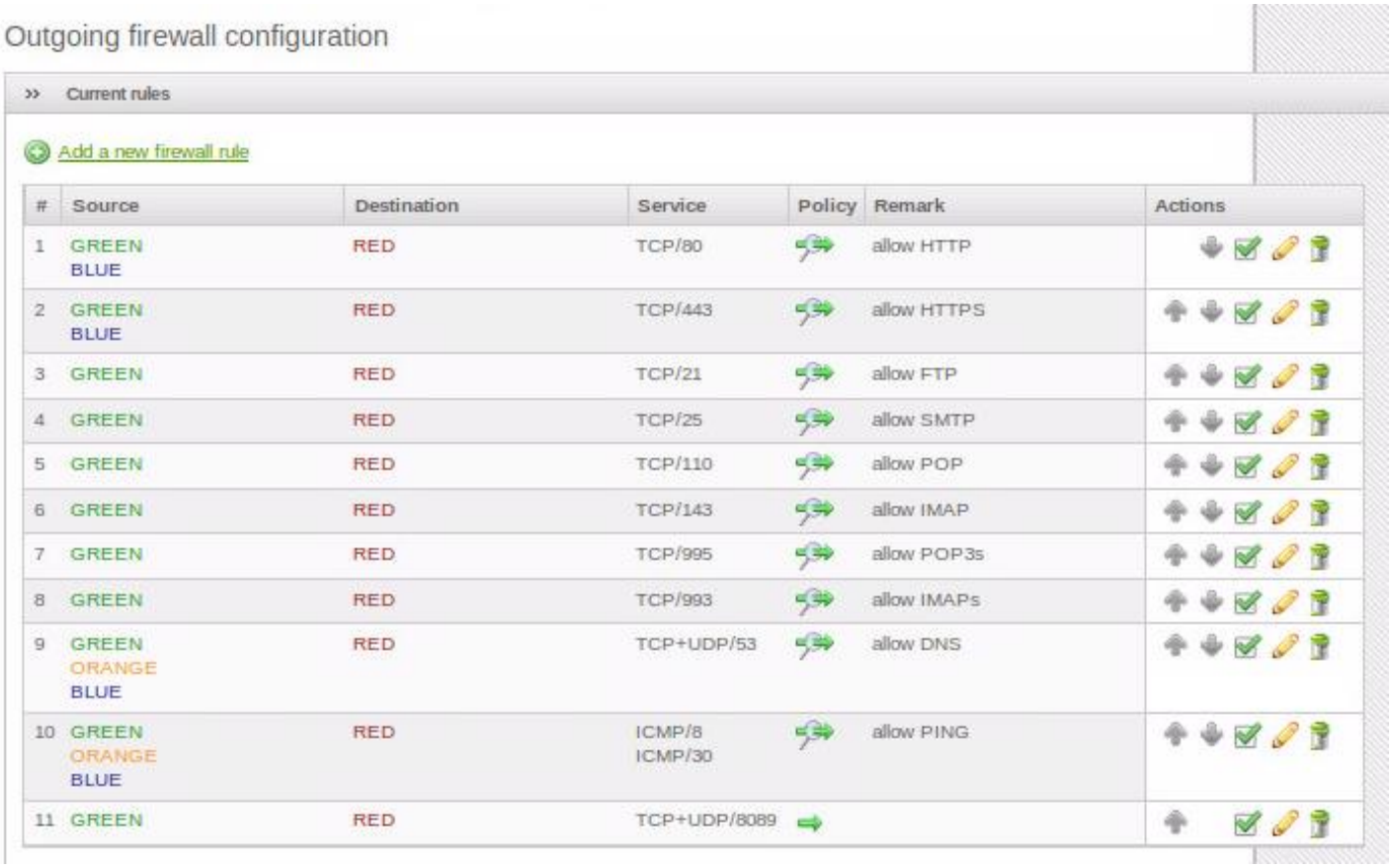
Windows operating systems

The table lists the Windows computing platforms that Splunk Enterprise supports.

Operating system	Architecture	Enterprise	Free	Trial	Universal Forwarder
Windows Server 2008 R2	x86 (64-bit)	D	D	D	D
Windows Server 2012 and Server 2012 R2	x86 (64-bit)	✔	✔	✔	✔
Windows 8, 8.1, and 10	x86 (64-bit)		✔	✔	✔
	x86 (32-bit)		***	***	✔

Kuva 2. Splunkin järjestelmävaatimukset Windows-alustoille.

Palomuurisäännöt: liikenne ulospäin



Outgoing firewall configuration

» Current rules

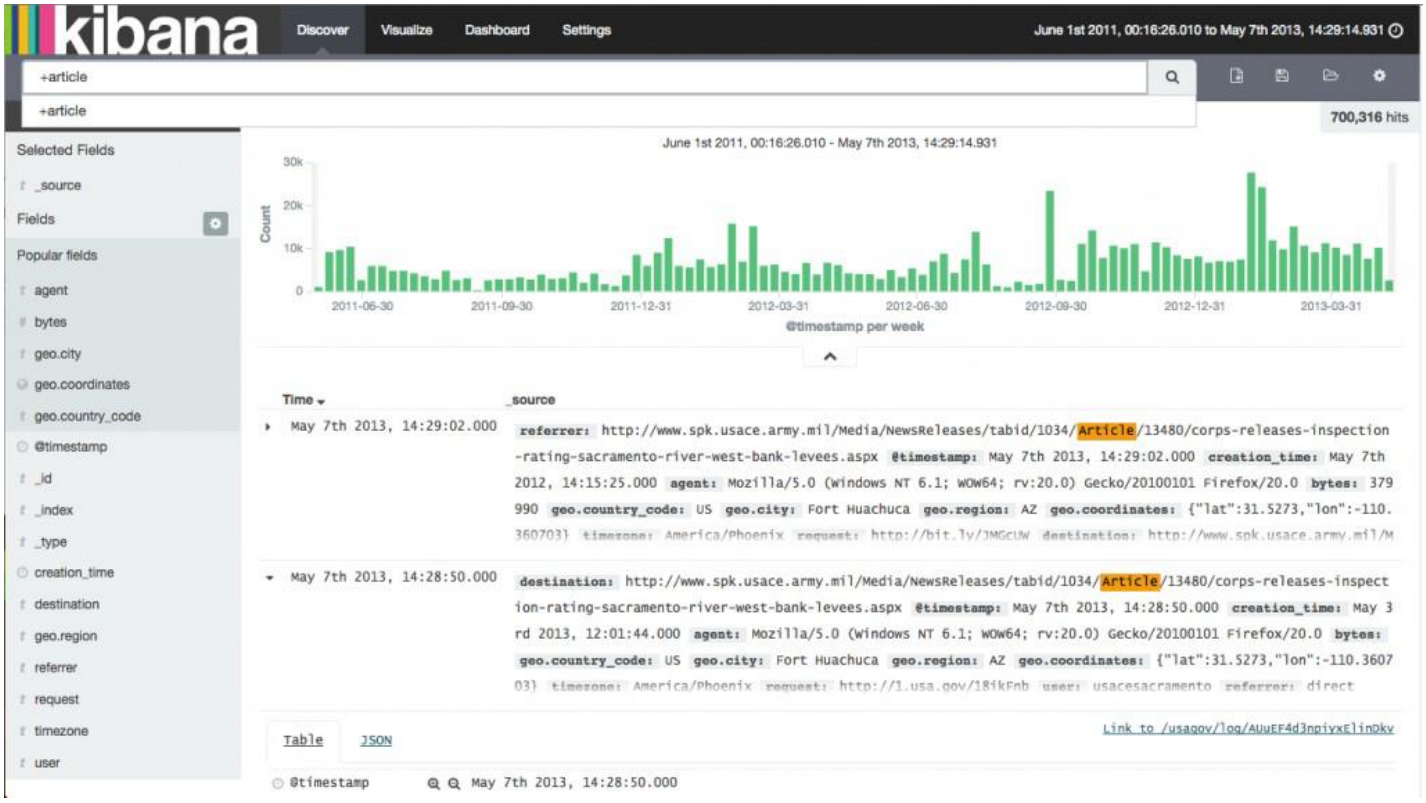
Add a new firewall rule

#	Source	Destination	Service	Policy	Remark	Actions
1	GREEN BLUE	RED	TCP/80		allow HTTP	
2	GREEN BLUE	RED	TCP/443		allow HTTPS	
3	GREEN	RED	TCP/21		allow FTP	
4	GREEN	RED	TCP/25		allow SMTP	
5	GREEN	RED	TCP/110		allow POP	
6	GREEN	RED	TCP/143		allow IMAP	
7	GREEN	RED	TCP/995		allow POP3s	
8	GREEN	RED	TCP/993		allow IMAPs	
9	GREEN ORANGE BLUE	RED	TCP+UDP/53		allow DNS	
10	GREEN ORANGE BLUE	RED	ICMP/8 ICMP/30		allow PING	
11	GREEN	RED	TCP+UDP/8089			

Kuva 1. Palomuurisäännöt ulospäin lähtevälle liikenteelle.

Näyttökuvat Kibanasta

Kuva 1. Näyttäkuva Kibanan dashboardista tummalla teemalla [42].



Kuva 2. Näyttökuva Kibanan datahausta [43].