Jussi Pöllänen

Keskitetyn lokienhallinnan hyödyt verkonvalvonnassa

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikka

Insinöörityö

14.5.2013

Metropolia

Tekijä Otsikko Sivumäärä Aika	Jussi Pöllänen Keskitetyn lokienhallinnan hyödyt verkonvalvonnassa 54 sivua 14.5.2013
Tutkinto	insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Tietoverkkotekniikka
Ohjaaja	Tietohallintojohtaja Pasi Ulkuniemi Lehtori Erik Pätynen

Tämän opinnäytetyön tarkoituksena on tutkia lokienhallinnan teoriaa sekä selvittää mitä hyötyjä keskitetyllä lokienhallinnalla saavutetaan organisaation tietoverkossa. Käytettynä lokienhallintajärjestelmänä toimii ohjelmisto nimeltä Splunk, jonka toimintaan ja rakenteeseen perehdytään yksityiskohtaisesti.

Työn yleisessä osassa käydään läpi tietoverkon hallintamalli sekä useimmin käytetyt verkonvalvontamenetelmät. Työssä kuvataan valvontamenetelmien roolit verkonhallinnassa sekä näiden suhde erilliseen lokienhallintajärjestelmään.

Ennen itse lokienhallintajärjestelmään tutustumista selvitetään lokienhallintaan liittyvät haasteet sekä suunnitteluun vaikuttavat seikat. Työssä tarkasteltu Splunk-ympäristö on toteutettu virtuaalisesti hallittavuuden helpottamiseksi. Tämä mahdollistaa datan vaivattoman tuonnin järjestelmään sekä paremman skaalautuvuuden.

Työn empiirisessä osassa keskityttiin lähinnä tietoverkkolaitteiden tuottaman lokidatan tutkimiseen. Tämän tiedon pohjalta huomattiin hyötyjä vianselvityksessä, tietoturvassa ja yksityskohtaisten räätälöityjen raporttien luonnissa.

Avainsanat	Lokienhallinta, Splunk, verkonvalvonta

Author	Jussi Pöllänen
Title	Benefits of centralized log management in network monitoring
Number of Pages	54 pages
Date	14 May 2013
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Networking
Instructor(s)	Pasi Ulkuniemi, CIO Erik Pätynen, Senior Lecturer

The aim of this thesis is to examine the theory of log management and to determine what benefits can be gained by using centralized log management in an organization's data network. The log management system used was software called Splunk. This thesis describes the functionality and components of a Splunk deployment.

During the first part of the thesis, the network management model and the most frequent traditional network monitoring methods are introduced. The thesis also discusses the relationship between traditional network monitoring and log management.

Before exploring Splunk, several instances of common challenges and design patterns for successful log management are examined. The Splunk deployment described in this thesis was implemented on a virtual platform. Overall, the implementation allowed better scalability and easier management.

The empirical part of the thesis project concentrated mainly on log traffic generated by network devices. Based on the results, benefits can be seen regarding incident troubleshooting, improving security and generating custom reports.

Keywords	Log management, Splunk, network monitoring

Sisällys

Sanasto

FCAPS	*Fault, configuration, accounting, performance, security*. ISO:n määrittelemä verkonhallintamalli.
IETF	*Internet Engineering Task Force*. Internet-protokollien standardoinnista vastaava organisaatio.
IPSec	TCP/IP-perheeseen kuuluva protokollajoukko internet-yhteyksien salaukseen.
ISO	*Internal Organization for Standardization*. Kansainvälinen standardisoimisjärjestö.
MIB	*Management Information Base*. Tietoverkon laitteiden tiedon jakamiseen käytettävä tietokanta.
NIST	*National Institute of Standards and Technology*. Yhdysvaltalainen virasto, joka kehittää mittaustekniikoita ja standardeja.
RADIUS	*Remote authentication dial in user service*. Käyttäjän tunnistukseen ja valtuutukseen käytettävä protokolla.
RFC	*Request for comments*. IETF-organisaation määrittelemien standardien kutsumanimi.
SSO	*Single sign-on*. Menetelmä, jonka avulla yhdellä kirjautumisella päästään useisiin eri palveluihin.
TCP/IP	*Transmission protocol/Internet Protocol*. Internet-standardien mukainen protokollaperhe.
UDP	*User datagram protocol*. Kuljetuskerroksen yhteydetön protokolla tiedon siirtoon.

1 Johdanto

Tietoverkon toiminnan laatuun vaikuttavat useiden tekijöiden summa. Rakenteellisesti tietoverkot monimutkaistuvat resurssien lisääntyessä jatkuvasti, eikä niiden hallintaan riitä enää manuaaliset ja yksinkertaiset työkalut. Perinteisen verkonvalvonnan lisäksi on huomattu, että organisaation tietoverkossa olevat sovellukset tuottavat jatkuvasti informaatiota toiminnastaan, mutta tätä informaatiota harvoin käytetään hyödyksi sen monimuotoisuuden vuoksi. Käsitteenä big data on ylläpitäjille tuttu, mutta useita eri lokitietoja hyödyntävät järjestelmät ovat hankalia pystyttää ja ylläpitää.

Nykypäivänä organisaation on usein järkevämpi vaihtoehto ulkoistaa tietoliikenteen hoito ulkopuoliselle taholle, jolloin organisaation omat IT-resurssit voivat keskittyä omien prosessiensa kehittämiseen. Forte Netservices on DNA:n tytäryhtiönä toimiva globaaleja tietoturva- ja tietoverkkopalveluja yrityksille tarjoava palveluntarjoaja. Jo pitkään tuotannossa olleiden perinteisten laite- ja sovellusvalvontapalveluiden lisäksi uutena vaihtoehtona on tuotettu pilvipalveluna toimiva skaalautuva keskitetty lokienhallintajärjestelmä. Järjestelmä hyödyntää eri lähteistä tekstipohjaista tietoa kirjoittavia sovelluksia.

Tässä insinöörityössä tutkitaan keskitetyn lokienhallintajärjestelmän tuomia lisäominaisuuksia organisaation tietoverkon ja sovelluksien valvonnassa ja ylläpidossa. Työn tarkoituksena on selvittää, kuinka lokienhallintajärjestelmä toimii ja kuinka sillä voidaan parantaa aktiivisten ympäristöjen tietoturvaa ja vianselvitystä.

Työn kokonaisuus koostuu kolmesta yhtenevästä osasta. Ensimmäisessä luvussa käydään läpi verkonhallinnan peruselementtejä ja valvontamenetelmiä. Toisessa luvussa tutustutaan lokienhallinnan teoriaan tutkimalla ympäristön pystyttämiseen liittyviä vaatimuksia ja kriteereitä. Kolmannessa luvussa tutkitaan, miten pilvipalveluna käytettävä Splunk-järjestelmä toimii käytännössä.

2 Verkonhallinta

2.1 Yrityksen tietoverkko

Jatkuvasti kasvavan liikennemäärän myötä tietoverkkojen ylläpito on nykypäivän yrityksessä tärkeässä asemassa. Heikko suorituskyky tai mahdolliset katkot voivat pahimmassa tapauksessa aiheuttaa yritykselle selviä tulojen menetyksiä. Esimerkiksi langattoman yhteyden autentikaation epäonnistuminen logistiikkakeskuksessa voi aiheuttaa kuljetuksen keskeytymisen tuotteiden kirjaamisen ollessa mahdotonta. Suuremmassa verkkolaitteessa tapahtuvan virheen myötä koko konttori voi lamaantua vian ajaksi. Pidempiaikaiset pienemmät tietoliikenneongelmat aiheuttavat henkilöstön keskuudessa työmoraalin heikentymistä.

Tietoverkon valvonnalla pyritään minimoimaan mahdolliset häiriötilanteet sekä pitämään kriittiset toiminnat normaalina. Yrityksen verkko koostuu monesta eri komponentista, joihin lukeutuvat esimerkiksi reitittimet, kytkimet, palvelimet ja näiden prosessit. Nämä voidaan jakaa eri tasoihin, jotka ovat sovellus-, palvelin- ja verkkotaso. Erilaiset ohjelmistokokonaisuudet tukevat osa-alueiden hallintaa ja niitä sovelletaan käyttötarpeiden mukaan. Esimerkiksi toista ohjelmistokokonaisuutta voidaan käyttää fyysisten laitteiden tilan valvontaan ja toista palvelimilla ajettavien prosessien käytön mittaamiseen.

Palveluntuottajan ja asiakkaan välillä on sovittu jatkuvan palvelun laadun määrityksestä, josta käytetään nimitystä palvelutasosopimus (SLA). Siinä on kyse etukäteen sovituista tasoluokista ja -tavoitteista, joiden toteumista voidaan myös mitata erilaisilla valvontasovelluksilla.

2.2 Verkonhallinnan vastuualueet

Verkonhallinnan toteuttamiseen käytetään useita eri tapoja, jotka keräävät verkon komponenteista informaatiota. Kokonaisuuden täytyy kuitenkin olla helppo käyttää ja tarvittavat ominaisuudet helposti saatavilla. Tärkeitä verkon asetuksiin ja suunnitteluun

vaikuttavia mittareita ovat esimerkiksi verkon käyttöaste ja viiveet. Tietoverkon kasvun myötä toimintojen kuten raporttien automatisointi on myös välttämätöntä [2;3].

Kansainvälisen standardointiorganisaatio ISO:n määrittelemät FCAPS-verkonhallinnan alueet ovat seuraavat:

- vianhallinta
- konfiguroinnin hallinta
- käytönhallinta
- suorituskyvyn hallinta
- turvallisuuden hallinta

Vianhallinta sisältää alueena vian paikallistamista, eristämistä sekä korjaamista. Pääasiallisena tavoitteena on vian korjauksen lisäksi niiden ehkäiseminen, eli niiden toistumisen estäminen. Tärkeimmät työkalut liittyvät lokitiedon hallitsemiseen ja valvottavien komponenttien hälytyksiin [2;3].

Konfiguraation hallinta kattaa laitekohtaiset konfiguraatiot. Yksittäiset laitteet tukevat useita eri toimintoja, joten niiden yhteistyökyvykkyyden varmistaminen on tärkeää. Prosessit ovat riippuvaisia laitteiden välisistä yhteyksistä, jolloin yksittäinen muutos voi aiheuttaa vikatilanteen. Edellisen tapauksen varalta tehtyjä muutoksia on hyvä monitoroida, jotta tarvittaessa toimivaan tilanteeseen on helppo palata [2;3].

Käytönhallinnalla tarkoitetaan verkon resurssien todellisen käytön mittaamista. Ylläpitäjän täytyy pystyä seuraamaan ryhmä- tai käyttäjätasolla, kuinka palvelua käytetään. Alueeseen liittyy myös tietoturvaseikat, sillä jokaisella käyttäjällä tai ryhmällä tulee olla vain tietyt oikeudet. Kattavan informaation myötä voidaan suunnitella verkon käyttö tehokkaammaksi ilman, että yksittäinen ryhmä voi väärinkäyttää tai kuormittaa palvelua muiden kustannuksella. Usein palvelun käyttöä myös laskutetaan sen käytön perusteella [2;3].

Verkon suorituskyvyn tehokkuus on resurssien jakamisen myötä loppukäyttäjien näkökulmasta tärkeimpiä seikkoja. Monet sovellukset ovat hyvin virhealttiita ja pahimmassa tapauksessa niiden käyttö on mahdotonta huonosti optimoidulla verkkokokoonpanolla. Suorituskyvyn mittaamisessa tärkeimpiä mittareita ovat verkon kapasiteetin käyttöaste, viiveet sekä mahdolliset pullonkaulat. Analyysin tekemiseksi ylläpitäjän on valittava joukko laitteita ja suhteuttaa niistä saadut tulokset verkon normaalitoimintaan.

Turvallisuuden hallinta käsittää verkon ja sen ylläpitoon liittyvien laitteiden pääsyn kontrollointia. Lisäksi alue kattaa näiden laitteiden perusteella tuotetun informaation saatavuutta. Esimerkkinä ovat lokitiedostot ja luodut raportit. Tiukentuneen tietoturvasuojan myötä pääsyt rajataan vain niitä tarvitseville tahoille [1].

2.3 Verkonhallintajärjestelmä

Verkonhallintajärjestelmä (Network management system) koostuu kokoelmasta verkkotyökaluja, jotka on sulautettu yhteen. Yleisenä tavoitteena on pitää eri käyttöliittymien ja ohjelmistojen määrä mahdollisimman pienenä, mutta samalla tarjota kaikki tarpeelliset toiminnot verkon ylläpitämiseksi [2].

Tietoverkon infrastruktuuri näkyy järjestelmässä yhtenäisenä, jolloin siitä käy ilmi jokaisen verkkolaitteen tiedot ja tila. Verkon aktiiviset komponentit viestittävät järjestelmälle jatkuvasti oman tilansa konfiguroitujen asetusten mukaisesti [2].

Jokaisella verkon laitteella on jokin ohjelma (Network management equipment), joka mahdollistaa verkon yli kommunikoimisen. Tämä verkonhallintayksikkö hoitaa vähintään seuraavia toimintoja:

- kerää tietoja verkkoliikenteeseen liittyvästä aktiviteetistä
- säilyttää informaatiota lokaalisti

- vastaa managerin pyytämiin komentoihin (esimerkiksi informaationhaku tai parametrin muutos)

- raportoi tilamuutokset tarpeen vaatiessa

Yksi tai useampi verkonhallintayksikkö toimii verkonhallintakeskuksena (NMA manager), jolta käsin verkon hallinta tapahtuu keskitetysti. Tämän verkonhallintakeskuksen kautta voi siis hallitusti syöttää komentoja verkon asiakaslaitteille. Edellä mainituilla asiakaslaitteilla sijaitsevaa hallintayksikköä kutsutaan myös agentiksi. Agentti voi sijaita millä vain verkon laitteella, esimerkiksi reitittimellä tai palvelimella. Usein verkonhallintakeskuksia sijaitsee tietoverkossa useampi mahdollisten vikatilanteiden varalta [2;3].

Verkon agenteilta saatava informaatio voidaan jakaa sekä staattiseen että dynaamiseen informaatioon. Staattinen informaatio käsittää harvoin muuttuvan tiedon, kuten esimerkiksi ohjelmiston versionumeron tai verkkoliitäntöjen määrän. Dynaamisella tiedolla tarkoitetaan jatkuvasti muuttuvaa informaatiota, kuten esimerkiksi muistin ja prosessorin käyttöaste [2].

2.4 Tietoverkon tilan valvontamenetelmiä

SNMP-protokolla (Simple Network Protocol)

Aiemmin läpikäytyyn verkonhallintajärjestelmämalliin perustuen on käytössä useita eri valvontatapoja. Tekniikat eroavat toisistaan ja ne vaihtelevat kompleksisuudeltaan. SNMP on yleisimpiä standardeja tietoverkkojen hallinnassa ja se on IETF:n (Internet Engineering Task Force) määrittämä TCP/IP-protokollaperheen protokolla. Se koostuu protokollasta itsesestään sekä MIB-tietokannasta. Hallitut resurssit kuvataan objekteina jotka kuvaavat jotakin agentin ominaisuutta. Objektit esitetään standardoituna puuna MIB-tietokannassa, josta manageri pyytää tietoja. Kaikilla SNMP:n versioilla on seuraavat perusoperaatiot:

- Luku (Get) hakee agentin MIB:stä objektin arvon.

- Lue seuraava (GetNext) hakee seuraavan objektin arvon.
- Kirjoitus (Set) asettaa agentin MIB:n objektille arvon.
- Ilmoitus (Trap) on tapa, jolla agentti voi lähettää ilmoituksen managerille.

Trap-viesteillä laitteelle voidaan asettaa esimerkiksi prosessorinkäytölle tietty raja, jonka täyttyessä siitä ilmoitetaan managerille. SNMP toimii UDP-porteilla 161 ja 162. SNMP:stä on olemassa kolme versiota (SNMPv1, SNMPv2, SNMPv3). Versiot 1 ja 2 ovat hyvin lähellä toisiaan, mutta versiossa 3 lisättiin datan salaus sekä vahva todennus. SNMP:n versioiden päästandardit ovat RFC 1155-1157 (v1), RFC 1901-1910 (v2) ja RFC 3411-3418 (v3) [2;5;16].

RMON-tarkastelu (Remote Network Monitoring)

RMON on SNMP-standardiin kehitetty lisäosa MIB-tietokannan laajentamiseen. Kyseessä on enemmän liikennepohjaisen datan analysointiin kehitetty tekniikka perinteiseen SNMP:hen verrattuna, sillä varsinaisella SNMP:lla mitataan laitteiden yksittäisten objektien tiloja. RMON tarvitsee myös erillisen agentin (kutsutaan myös nimityksellä probe) päätelaitteelle, joka kerää datan omaan tietokantaansa. Erona on myös kerätyn datan säilyminen agentilla, jolloin liikenne managerin ja päätelaitteen välillä vähenee. Manageri pyytää dataa vain tarvittaessa. Toisaalta tämä kuormittaa puolestaan päätelaitteen omaa kapasiteettia entistä enemmän [2;17].

RMON tunnetaan kahtena eri versiona. Ensimmäisen version liikenteen tarkastelu oli mahdollista vain OSI-malliston kahden alimman kerroksen osalta. Versio lisäsi MIB-tietokantaan hyödyllisiä ryhmiä, kuten linjan käyttöstatistiikka, konekohtainen lähiverkon statistiikka, yhteyksien määrä tiettynä ajanjaksona sekä mahdollisuus pakettikaappaukseen [6;17].

Päivitetyssä RMON2-versiossa lisättiin ylemmät OSI-mallin kerrokset mukaan, sekä kymmenen uutta lisäryhmää hallintatietoihin. Tämä mahdollisti liikenteen tarkkailun ja analyysin IP-tasolla. Tärkeimmät RMON-standardit ovat RFC 2819 ja RFC 4502 [2;6;17].

Flow-analysaattorit

NetFlow on Cisco Systemsin kehittämä protokolla IP-liikenteen analysoimiseen. Siitä on kehittynyt standardi verkkoliikenteen seuraamiseen ja siihen löytyy tuki Ciscon lisäksi useista muista high end -verkkolaitteista. Määritelty standardi on RFC3954. Verkkolaitteessa tuki konfiguroidaan verkkoporttiin ja laitteen keräämä data lähetetään kerääjäpalvelimelle, jolla hoidetaan liikenteen varsinainen analyysi [7].

Flow tarkoittaa reitittimen läpi kulkevia paketteja, joiden tietoja tutkimalla muodostetaan varsinainen tallennettava data. Flow siis tunnistetaan etukäteen määriteltyjen seikkojen perusteella, jotka Ciscon standardin mukaan ovat seuraavat:

- verkkoliitäntä
- lähdeportti
- kohdeportti
- lähdeosoite (IP)
- kohdeosoite (IP)
- IP-protokolla
- IP ToS.

NetFlow-paketit kulkevat kerääjäpalvelimelle UDP:lla. Standardiportti on 2055, mutta esimerkiksi portti 9995 on usein käytetty [7;18].

SFlow on samantyylinen liikenteen keräys- ja analyysitekniikka, mutta se on kehitetty eri tahon toimesta. InMonin luoma SFlow perustuu laitteelle erikseen integroituun mikrosiruun, mikä pienentää hieman laitteeseen kohdistuvaa rasitetta NetFlowiin verrattuna. SFlow toimii myös UDP-protokollalla ja oletusporttina käytetään porttia 6343 [7].

SFlowissa näytteiden otto kokonaisliikenteestä perustuu agentilla määritettyihin skaalaaviin ottotiheyksiin. Koska sFlow ei ota huomioon kaikkia laitteen läpi kulkevia paketteja, täydellisen tarkka analyysi ei onnistu. Sflow.org-liitoksella on ottotiheyksiin omat suositukset linkin nopeuden mukaan. sFlow-tuki löytyy statistiikan mukaan Netflowia useammin verkkolaitteista, erityisesti kytkimistä [7;4].

Huomionarvoista on myös mainita, että sFlow ja Netflow ovat suosituimpia, mutteivät ainoita flow-vaihtoehtoja. Muita Netflow-implementaatioita ovat muun muassa Jflow sekä NetStream.

2.5 Tietoverkon valvonnan laajentaminen

Valvontaympäristöjen tyypit

Yrityksen tietoverkossa on lukematon määrä laitteita, joiden kirjoittama lokitieto on tärkeä informaationlähde suorituskyvyn ja tietoturvan parantamiseksi. Näiden yksittäinen manuaalinen hallinta on hyvin vaivalloista ja epäkäytännöllistä, mistä johtuen niitä ei läheskään aina käydä läpi säännöllisesti. Perinteisen tilavalvonnan lisäksi on olemassa keskitettyjä järjestelmiä, jotka lukevat laitteen luomaa lokitietoa talteen. Kun informaatio on saatavilla keskitetysti, sitä voidaan monitoroida ja analysoida vaivattomammin ja tehokkaammin. Useissa ratkaisuissa mukaan voidaan liittää valvonta ja raportointi, joka on täysin riippumaton loppusovelluksen tyypistä [10].

Nykypäivänä on olemassa kolme erityyppistä ympäristöä, jotka ovat Security Event Management (SEM), Security Information Management (SIM), sekä Security Information Event Management (SIEM) [8].

Security Event Management (SEM)

SEM-tyypissä keskitytään tietoturvan kannalta oleellisiin verkkolaitteisiin, kuten palomuureihin ja tunkeutumisen havaitsemisjärjestelmiin (IDS). Järjestelmä lukee sille

konfiguroitujen laitteiden luomaa lokidataa reaaliajassa ja reagoi siihen mahdollisesti automaattisesti hälytyksen muodossa. Tieto luetaan usein Syslog-tyyppisestä lähteestä.

Ratkaisusta riippuen valtavaa lokimäärää voi jäsentää, jättämällä lopulliseen arkistoon vain tarvittavan informaation. Eri ohjelmistosta riippuu myös se, kuinka tehokkaasti eri laitteiden tuottamaa dataa voidaan analysoida [10].

Security Information Management (SIM)

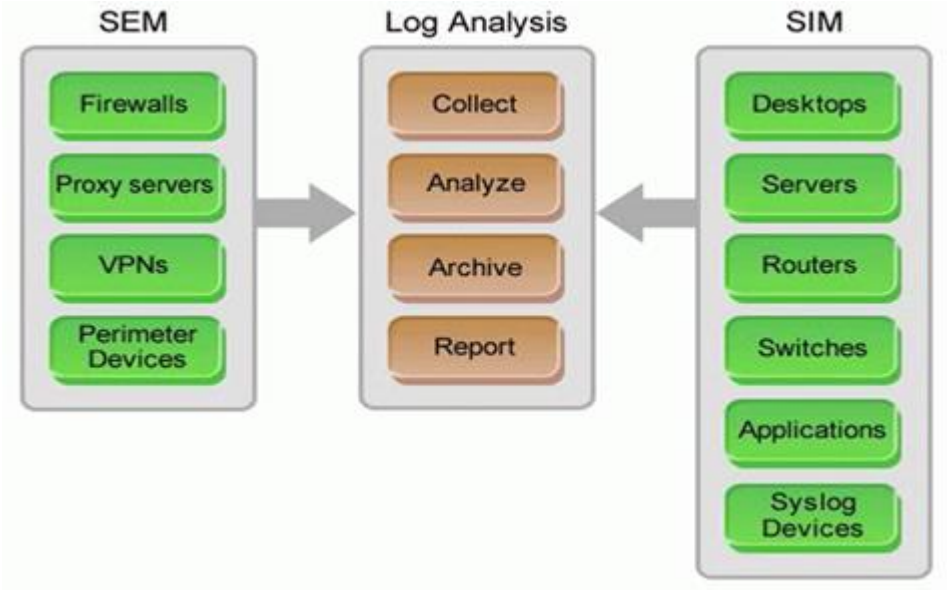
SIM-mallin työkalut mahdollistavat sovellusten ja käyttöjärjestelmien lokien hallinnan. Järjestelmä kerää ohjelman tuottamaa tietoa itsestään myöhempää analyysia ja arkistointia varten. Tallessa olevan datan perusteella voidaan luoda yksityiskohtaisia raportteja.

Ratkaisussa asiakaskoneelle asennetaan yleensä agentti, joka määrittää, kuinka lopullinen data tuodaan keskuspalvelimelle. Dataa voidaan lukea esimerkiksi laitteella sijaitsevista lokitiedostoista, joiden muutoksia agentti valvoo [8].

Security Information and Event Management (SIEM)

SIEM tarkoittaa aiemmin läpikäytyjen ympäristöjen yhdistämistä. Käytännössä tämä mahdollistaa kaikentyyppisen lokidatan lukemisen, analysoimisen ja arkistoimisen [19].

Toiminnat integroidaan yhtenäiseen käyttöliittymään, joka mahdollistaa kokonaisuuden luontevan hallinnan. Kuvassa 1 nähdään SIEM-järjestelmän perusarkkitehtuuri [11].



Kuva 1. SIEM-järjestelmän perusrakenne [19].

Erilaisia SIEM-tuotteita ja ratkaisuja on olemassa useita kymmeniä, mutta niillä jokaisella on ominaisuuksina seuraavat perusperiaatteet:

- tiedon keräys
- korrelaatio
- automaattinen valvonta
- graafinen käyttöliittymä
- sisäinen valvonta
- tiedon säilytys.

Perinteisiin monitorointimenetelmiin verrattuna SIEM-järjestelmät siis hyödyntävät suurempaa määrää verkonvalvonnasta saatavasta datasta. Perusvalvontaan nähden keskitetyissä lokienhallinta- ja analyysiratkaisuissa on mahdollista yhdistää valvontaan lähteet, jotka muuten jäisivät huomioimatta. Reaaliaikaisen keräyksen seurauksena lokitiedon sisältö ei muutu ja niiden säilytys on loogista [19].

Seuraavan luvun aikana käydään läpi niin kutsutun machine datan, eli eri prosessien ja sovelluksien tuottaman tiedon lähdettä, sisältöä ja tarkoitusta NIST 800-92 -standardia mukaillen.

3 Lokienhallinta

3.1 Lokitieto

3.1.1 Lokitiedon määritelmä

Lokilla tarkoitetaan tallennetta, johon on kirjattu organisaation tietoverkossa tai järjestelmässä käytyjä tapahtumia. Tarkemmin määriteltynä lokit koostuvat tietueista, joista jokainen sisältää paljon erilaista informaatiota tietystä tapahtumasta. Alun perin lokien tarkoituksena oli lähinnä auttaa ongelmanselvityksessä, mutta nykyään ne ovat olleet jo jonkin aikaa osa prosessien kehitystyötä ja tietoturvan ylläpitoa [9].

Organisaation tietoverkon kannalta relevantti lokitieto voidaan jakaa kahteen osaan:

- tietoturva- ja verkkolokit
- käyttöjärjestelmä- ja ohjelmatason lokit.

Sovellustason tuottama lokitieto voi liittyä myös vahvasti tietoturvaan. Esimerkiksi sovelluksen raportoimat kirjautumis- ja tunnistustiedot ovat tärkeätä informaatiota tietoturva-auditoinnissa. Useimmissa tapauksissa lokitettavat prosessit ovat koko ajan käynnissä, jolloin lokidataa tallennetaan jatkuvasti. Normaalisti lokiin sisällytetään kaikki mahdollinen tieto tapahtumasta, jonka vuoksi järjestelmälle varatut resurssit voivat kasvaa suuriksi. Tästä syystä lokista on tarve käsitellä vain tarvittavat kentät [10].

3.1.2 Tietoturvalokit

Organisaation tietoverkkoon kuuluu paljon verkko- ja tietoturvalaitteita jotka ovat suuri osa kokonaislokimäärää.

Yleisiä lähteitä tämän kategorian lokeille ovat muun muassa:

- palomuurit

- VPN-laitteet
- todennuspalvelimet
- reitittimet
- kytkimet
- välityspalvelimet

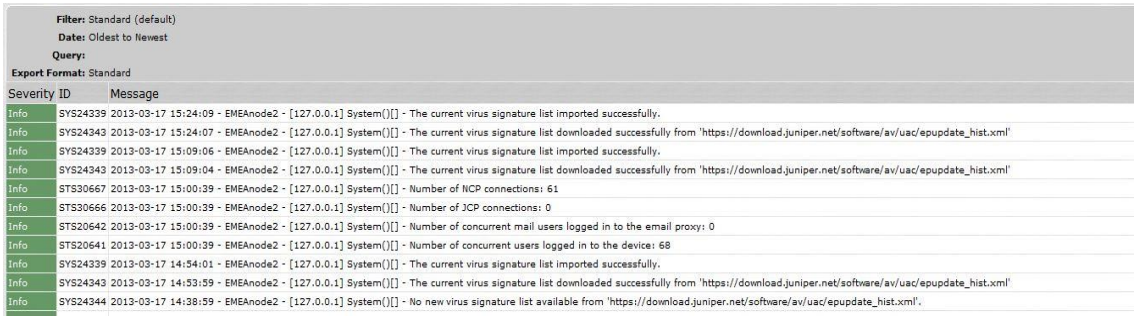
Palomuurit ja reitittimet käsittelevät liikennettä sääntöpohjaisesti. Reitittimellä voi esimerkiksi olla pääsylistoja, joiden perusteella reititetään vain tietynlainen liikenne. Palomuurit laajentavat pääsylistojen käsitettä monipuolisemmalla tilatietoisella verkkoliikenteen hallinnalla. Riippuen palomuurin UTM-ominaisuuksista lokiin kirjoitetaan lähteen, kohteen ja säännön lisäksi paljon muutakin informaatiota kuten sisällönsuodatukseen ja liikenteen tyyppiin liittyvää tietoa. Kuvassa 2 nähdään palomuurin lokia FortiOS 4.0 -käyttöjärjestelmän kautta tarkasteltuna. Palomuurin IDS-valmiudet mahdollistavat yksityiskohtaisen tallennuksen koskien mahdollisia tunkeutumisyrityksiä ja hyökkäyksiä [15].

2011-08-10 21:45:17		auth		FSSO-logoff	FSSO-logoff event from server:
2011-08-10 21:45:17		auth		FSSO-logon	FSSO-logon event from server:
2011-08-10 21:45:17		voip		permit	
2011-08-10 21:45:17		ha			HA device(interface) fail
2011-08-10 21:45:17		system		login	user admin logged into the fw -
2011-08-10 21:45:00		admin	http(10.10.20.3)	login	Administrator admin logged in s

Kuva 2. Esimerkki Fortigate UTM-palomuurilaitteen lokitietueista.

VPN-yhteyksille ja todennuspalvelimille ovat yhteistä käyttäjätietojen hallinta. Useimmat VPN-ratkaisut tallentavat vähintään käyttäjän sisään- ja uloskirjautumiset aikamerkinnällä. Kehittyneemmissä laitteissa, kuten dedikoiduissa SSL VPN ratkaisuissa käyttäjän toimia voi seurata hyvin tarkasti. Perustiedon lisäksi kirjataan käytetyt IP-osoitteet, avatut resurssit sekä siirretty datamäärä session aikana. Todennäköisesti käytössä on myös keskitettyjä käyttäjätietokantoja ja hakemistopalveluita, jotka sijaitsevat erillisillä autentikaatiopalvelimilla. Nämä toiminnot varmistavat lopullisen kirjautumisen onnistumisen tai epäonnistumisen. Autentikaatiotapoja on useita ja

tavallisten autentikaatioprotokollien lisäksi palveluissa käytetään muun muassa vahvaa tunnistautumista sekä kertakirjautumista tietoturvan ja käyttömukavuuden lisäämiseksi. Kuvassa 3 on esimerkki SSL VPN -laitteen luomasta lokista.



Filter: Standard (default)
Date: Oldest to Newest
Query:
Export Format: Standard

Severity	ID	Message
Info	SYS24339	2013-03-17 15:24:09 - EMEAnode2 - [127.0.0.1] System()[] - The current virus signature list imported successfully.
Info	SYS24343	2013-03-17 15:24:07 - EMEAnode2 - [127.0.0.1] System()[] - The current virus signature list downloaded successfully from 'https://download.juniper.net/software/av/uac/epupdate_hist.xml'
Info	SYS24339	2013-03-17 15:09:06 - EMEAnode2 - [127.0.0.1] System()[] - The current virus signature list imported successfully.
Info	SYS24343	2013-03-17 15:09:04 - EMEAnode2 - [127.0.0.1] System()[] - The current virus signature list downloaded successfully from 'https://download.juniper.net/software/av/uac/epupdate_hist.xml'
Info	STS30667	2013-03-17 15:00:39 - EMEAnode2 - [127.0.0.1] System()[] - Number of NCP connections: 61
Info	STS30666	2013-03-17 15:00:39 - EMEAnode2 - [127.0.0.1] System()[] - Number of JCP connections: 0
Info	STS20642	2013-03-17 15:00:39 - EMEAnode2 - [127.0.0.1] System()[] - Number of concurrent mail users logged in to the email proxy: 0
Info	STS20641	2013-03-17 15:00:39 - EMEAnode2 - [127.0.0.1] System()[] - Number of concurrent users logged in to the device: 68
Info	SYS24339	2013-03-17 14:54:01 - EMEAnode2 - [127.0.0.1] System()[] - The current virus signature list imported successfully.
Info	SYS24343	2013-03-17 14:53:59 - EMEAnode2 - [127.0.0.1] System()[] - The current virus signature list downloaded successfully from 'https://download.juniper.net/software/av/uac/epupdate_hist.xml'
Info	SYS24344	2013-03-17 14:38:59 - EMEAnode2 - [127.0.0.1] System()[] - No new virus signature list available from 'https://download.juniper.net/software/av/uac/epupdate_hist.xml'.

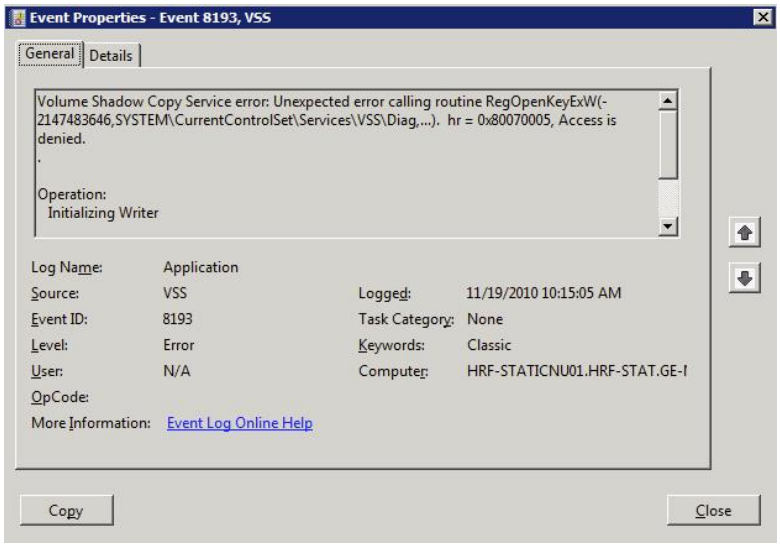
Kuva 3. Esimerkki Juniper Secure Access –etäratkaisun tuottamasta lokitietueesta.

Välityspalvelimet ovat tehokas tapa hallita organisaation julkista verkkoliikennettä. Asiakaskone ottaa yhteyden välityspalvelimeen, joka vasta muodostaa yhteyden kohdepalvelimeen. Ulospäin lähtevälle liikenteelle saadaan yksi kiintopiste ja samalla liikennettä voidaan tarvittaessa puskuroida kokonaisliikenteen vähentämiseksi. Autentikaatiotavoista riippuen käyttäjä voidaan yhdistää tätä koskevaan liikenteeseen ja valvonta helpottuu. Organisaation säännöistä riippuen internet-proxylle voidaan asettaa keskitetysti rajoitteita sallitulle liikenteelle. Esimerkiksi käyttäjille on teoriassa mahdollista sallia hakemistopalvelun ryhmien perusteella tiettyjä kategorioita internetistä ja tallentaa näiden käyttö lokiin. Välityspalvelinta käytetään usein myös pelkkään web-selailuun, jolloin vain HTTP-liikenne kierrätetään välityspalvelimen kautta. Nämä ratkaisut tunnetaan nimityksellä web-välityspalvelin [10].

3.1.3 Käyttöjärjestelmä- ja sovellustason lokit

Edellä läpikäytyjen lähteiden lisäksi lähes kaikki mahdolliset sovellukset ja ohjelmistot tuottavat lokia. Käsite on hyvin laaja, sillä jonkin tasoinen käyttöjärjestelmä löytyy jokaisesta operoitavasta laitteesta. Tyypilliset käyttöjärjestelmätyyppiset tapahtumat ovat esimerkiksi laitteen tai prosessin käynnistykset ja sammutukset. Tämäntyylisiä merkintöjä ei välttämättä tule kovin usein, mutta eri kohteilla on omia säädettäviä ominaisuuksia, joiden ilmoittamistiheyttä voi itse säädellä. Esimerkiksi VPN-laitteella voidaan kirjoittaa lokiin jokainen kerta, kun IPSec-tunnelin tila muuttuu [10].

Kategoriaan kuuluu myös auditointiin käytettävät tiedot, kuten verkkolaitteisiin kirjautumisyritykset, konfigurointimuutokset ja tiedostojakojen käyttö. Käyttöjärjestelmien lokista saa mahdollisesti myös erikseen informaatiota niillä pyörivistä ohjelmista. Lokeja hyödynnetään usein, kun vika on saatu paikannettua tiettyyn kohteeseen.



Kuva 4. Esimerkki Windows 2008 r2 -käyttöjärjestelmän lokitietueesta.

```
Feb 17 15:17:01 (none) /USR/SBIN/CRON[9667]: (root) CMD (   cd / && run-parts --report /etc/cron.hourly)
Feb 17 16:17:36 (none) /USR/SBIN/CRON[12114]: (root) CMD (   cd / && run-parts --report /etc/cron.hourly)
Feb 17 17:09:39 (none) winbindd[1338]: [2013/02/17 17:09:39.607577,  0] winbindd/idmap.c:201(smb_register_idmap_alloc)
Feb 17 17:09:39 (none) winbindd[1338]:   idmap_alloc module tdb already registered!
Feb 17 17:09:39 (none) winbindd[1338]: [2013/02/17 17:09:39.607645,  0] winbindd/idmap.c:149(smb_register_idmap)
Feb 17 17:09:39 (none) winbindd[1338]:   Idmap module passdb already registered!
Feb 17 17:09:39 (none) winbindd[1338]: [2013/02/17 17:09:39.607675,  0] winbindd/idmap.c:149(smb_register_idmap)
Feb 17 17:09:39 (none) winbindd[1338]:   Idmap module nss already registered!
Feb 17 17:09:39 (none) winbindd[1338]: [2013/02/17 17:09:39.614668,  0] winbindd/idmap.c:201(smb_register_idmap_alloc)
Feb 17 17:09:39 (none) winbindd[1338]:   idmap_alloc module tdb already registered!
Feb 17 17:09:39 (none) winbindd[1338]: [2013/02/17 17:09:39.614692,  0] winbindd/idmap.c:149(smb_register_idmap)
```

Kuva 5. Esimerkki Debian Wheezy -käyttöjärjestelmälokista.

Tyypillisesti sovellukset kirjoittavat toiminnoistaan tietoa omiin lokitiedostoihin, mutta on mahdollista, että ne hyödyntävät käyttöjärjestelmän omaa lokijärjestelmää. Informaation määrää riippuu täysin siitä, kuinka sovelluskehittäjän on lokituksen ja lokituksen määrän suunnitellut. Kuvissa 4 ja 5 nähdään eri käyttöjärjestelmien tapahtumalokia.

Seuraavassa on yleisiä sovelluksen lokityyppejä, jotka ovat hyödyllisiä organisaatiolle:

- asiakasohjelman kyselyt palvelimelta
- kirjautumistiedot
- sovelluksen käyttöstatistiikat.

Web-palvelimen lokitiedot ovat hyvä esimerkki hyödyllisestä informaatiosta, jota ei välttämättä osata hyödyntää. Tietueesta näkee, mistä osoitteestä pyyntö on tullut, mikä on tarkka haettu polku sekä mistä mahdollinen ohjaus sivulle on tullut. Oikeassa ympäristössä näitä voi hyödyntää ja muodostaa helposti tilastoja, kuten kävijämäärät ja eniten haetut sivut. Ongelmanselvityksessä lokeista nähdään tarkka kellonaika, milloin lokivirheilyt ovat alkaneet. Web-palvelimet tuottavat valtavan määrän lokia, jolloin suurilla kävijämäärillä tarkan tiedon säilyttämisen kanssa voi tulla ongelmia [13].

Useissa sovelluksissa käytetään myös kirjautumistietoja ja lokiin kirjoitetaan kattavasti käyttäjistä. Sovelluskohtaisista autentikaatiolokeista selviää mahdolliset väärinkäytökset ja lähdeosoite josta pyynnöt ovat tulleet. Itse sovelluksien käyttöä kuvaavista ominaisuuksista keräämällä saadaan statistiikkaa muun muassa siitä, mitkä ominaisuudet ovat eniten käytettyjä ja kuinka nopeasti komponentit toimivat [9].

3.2 Syslog

Syslog on yleinen standardi lokitukselle, ja sen määritelmänä on IETF standardi RFC5424. Syslogin avulla kohdealustalta voidaan siirtää lokitietoa keskitetylle järjestelmälle. Kohdelaitteella on käynnissä taustaprosessi, joka konfiguroidaan lähettämään tietyntasoista tietoa keräävälle palvelimelle. Palvelimella on syslogia tukeva ohjelma, joka ominaisuuksiensa mukaisesti käsittelee, suodattaa ja tallentaa tiedon [27].

Syslog-prosessi luo tietueelle prioriteetin kahden attribuutin mukaan:

- viestityyppi (facility)
- vakavuus (severity)

Viestityypillä tarkoitetaan merkintää, joka kertoo lokia tuottavan prosessin tyypin. Eri tyypit jaotellaan luokittain nollasta kahteenkymmeneenkolmeen. Syslog-protokolla kehitettiin alunperin Unix-pohjaiseksi, joten luokat mukailevat Unix-palveluprosesseja. Muille laitteille on olemassa kahdeksan paikallista tasoa, joita käytetään esimerkiksi verkkolaitteissa. Nämä luokat sisältyvät aiemmin mainittuun 24:n pääluokkaan [27].

Vakavuus-luokka määrittelee, minkä tasoista lokia välitetään eteenpäin. Syslogissa on seuraavat vakavuuden määrittävät tasot:

1. Emergency – korkein luokka, järjestelmä on käyttökelvoton
2. Alert – välitön korjaustarve – esimerkiksi primääriyhteyden toimimattomuus
3. Critical – kriittinen virhe – esimerkiksi varayhteyden toimimattomuus
4. Error – virhe – ei-kiireelliset virheilmoitukset
5. Warning – varoitus – esimerkiksi kiintolevyn tilan täyttyminen
6. Notice – huomautus – normaalit järjestelmäviestit
7. Debug – virheiden paikallistamista varten tuotettava tarkka lokitus

Viestityypin ja vakavuuden perusteella jokaiseen syslog-tietueeseen lasketaan prioriteetti kaavalla: Prioriteetti = Viestityyppi x 8 + Vakavuus. Syslog-liikenne kulkee

oletuksena UDP-protokollalla portin 514 kautta. On myös mahdollista, että eräillä laitteilla käytetään TCP-protokollaa tiedon lopullisen perillepääsyn varmistamiseksi [14].

3.3 Lokienhallintaan liittyvät haasteet

Lokienhallinnan haasteiden jaottelu

Oikein toteutettu lokienhallinta voi hyödyntää organisaation tietoturvatarpeita monella tavalla. Se mahdollistaa tärkeän informaation säilömisen ennalta määritetyksi ajaksi. Säännöllisin väliajoin käytävien lokianalyysien avulla pystytään ennakoimaan ja havaitsemaan mahdolliset tietoturvariskit ja operaationaaliset ongelmat. Säilötyn tiedon perusteella muutoin löydettyihin ongelmiin voi löytyä myös tärkeää tietoa. Lokitiedon perusteella luotujen trendiraporttien perusteella pystytään reagoimaan mahdollisiin palveluiden kapasiteettiongelmiin [11].

Usein lokitiedon hallinnassa kuitenkin esiintyy ongelmia, jotka syntyvät jatkuvasti lisääntyvän lokimäärän ja -lähteiden lisäännyttyä. Yleisimmät lokienhallintaan liittyvät haasteet voidaan jakaa kolmeen kategoriaan seuraavasti:

- lokin tuottamiseen liittyvät haasteet
- lokin säilyttämiseen liittyvät haasteet
- lokitiedon analysoimiseen liittyvät haasteet.

Lokitiedon tuottamisen haasteet

Keskitetyn lokienhallinnan tarkoituksena on hyödyntää mahdollisimman montaa eri lokilähdettä, ja organisaation sisällä näitä ovat eri käyttöjärjestelmät, verkkolaitteet ja ohjelmat. Samasta laitteesta voidaan generoida lokia useammasta eri prosessista samaan aikaan. Ongelmaksi muodostuu useiden eri lokiformaattien vaihteleva sisältö. Eri lokilähteet tallentavat muistiin eri tietoa, joten lokienhallinnan on vaikea huomata

yhteyksiä sisältöjen kanssa. Toisessa lokissa voi esimerkiksi lähteenä IP-osoite, kun taas toisessa käytetään käyttäjänimeä.

Poikkeavien tietojen lisäksi samaa asiaa tarkoittavien arvojen merkintätavoissa on eroja. Esimerkisi päivämäärä voidaan merkintä monella tavalla, mikä voi aiheuttaa ongelmia. Verkkolaitteissa liikenteessä käytettävä palvelu voidaan merkitä toisessa laitteessa nimellä ja toisessa porttinumerolla. Lokeissa käytetään myös normaalisti lokilähteen paikallista aikaa, joka ei ole välttämättä oikein. Tämä hankaloittaa perusteellista analyysiä, sillä tapahtumankulkua ei voi päätellä suoraan aikajanalta [10].

Hyvin erilaisten lokilähteiden lisäksi lokitiedon formaatit vaihtelevat tapauksittain. Lokia tallennetaan esimerkiksi tekstitiedostoihin, erilaisiin tietokantoihin, XML-tiedostoihin sekä binääritiedostoihin. Formaattikirjoon lukeutuu myös aiemmin läpikäydyt SNMP ja syslog.

Täyden hyödyn saavuttamiseksi usein lokienhallintajärjestelmät sisältävät tavan muuntaa vastaanotettava lokitieto yhtenäiseksi keräämisvaiheessa. Jatkuvasti tarkkailtavaan tietoon on mahdollista sisällyttää myös aikatiedot automaattisesti, sillä lokienhallintajärjestelmä näkee tallennetun tiedon reaaliajassa [10].

Lokin säilyttämisen haasteet

Modernin organisaation tietoverkko tuottaa päivittäin massiiviset määrät lokitietoa. Riippuen lokitettavien lähteiden määrästä datamäärät voivat nousta satoihin gigatavuihin päivässä. Ilman lokien säilyttämistä käsitteleviä sääntöjä levytilan hallinnan kanssa tulee ongelmia, jos tietoa ei säilytetä keskitetysti. Lokitettavalla lähteellä voi olla myös erikseen määritettävä lokin maksimikoko, jonka täyttyessä vanha data päällekirjoitetaan tai uuden kirjoitus lopetetaan. Pidempää säilömistä tai arkistointia varten täytyisi turvautua erilliseen arkistointijärjestelmään [10;14].

Lokitieto sisältää myös tietoturvan kannalta arkaa materiaalia. Esimerkiksi sähköpostit, tunnustiedot ja käyttäjän vierailemat verkko-osoitteet ovat helposti väärinkäytöksiin sovellettavaa, mutta normaalia lokitettavaa tietoa. Henkilön yksityisyyden suoja

turvataan henkilötietolaissa, joka määrää erikseen sovellettavat ohjeet tiedon käyttöön ja säilytykseen. Vajaavaisesti suojatun lokitiedon eheyteen liittyy myös riskitekijöitä. Älykäs haittaohjelma voi siivota jälkensä automaattisesti, tai lokitietoa voidaan muuttaa manuaalisesti [9;14].

Lokitiedon analysoimisen haasteet

Perinteisesti lokien aktiivisella analysoimisella organisaatiossa on pieni prioriteetti. Tehtävä kuuluu infrastruktuurin ylläpitäjille, joiden pääasiallinen aika kuuluu välittömien ongelmien ja operaatioiden suorittamiseen. Analysointiin ei ole erityisiä työkaluja, eikä henkilökunnalla ole aihealueen vaatimaa koulutusta. Prosessia pidetään yleisesti enemmän reaktiivisena kuin proaktiivisena, eli toimenpiteet alkavat vasta vian esiinnyttyä.

Ilman analysointiin suunniteltua ohjelmistoa eri lähteiden lokitietueiden mahdolliset yhteydet on hankala huomata. Laajoissa ympäristössä tämä muodostuu peräti mahdottomaksi, sillä lokien manuaalinen läpikäyminen on työlästä. Ratkaisua haettaessa turvaudutaan usein manuaalisesti erilaisten lokitietueiden priorisointiin, jolloin normaalilokista pystytään löytämään nopeammin poikkeamat. Ongelmaksi muodostuu kuitenkin lokien monimuotoisuus ja erilaiset prioriteettiluokittelut [10;14].

3.4 Lokienhallinnan toteutuksen suunnittelu

3.4.1 Vastuut ja roolit

Luotettavan ja toimivan lokienhallinnan perustana toimii etukäteen määritellyt säännöt ja valmistelut koskien lokienhallintaa. Näiden avulla kartoitetaan organisaation tarpeet ja hyödyt, mitä lokienhallinnalta haetaan. Osana suunnitteluprosessia on lokienhallinnan roolit ja vastuut organisaation sisällä, sillä henkilön rooli ja vastuu lokidatan käsittelyssä vaihtelee työnkuvan mukaan. Seuraavassa on esimerkkirooleja, joiden toimintaan kuuluu lokitiedon käsittelyä:

- tietohallinnon ylläpitäjät – lokilähteiden konfigurointi, näiden lokitiedon analysointi, raportointi sekä lokienhallintaohjelmistojen ylläpito

- tietoturvavastaavat – lokitusympäristön infrastruktuurin ylläpito verkkotasolla, tietoturvalaitteiden ylläpito, raportointi

- Service Desk – lokitiedon tehokas hyödyntäminen mahdollisissa vikatilanteissa

- ohjelmistokehittäjät – lokilähteen generoiman tiedon määrittely, mitä tietoa lokitetaan

- auditoijat – auditoimiseen käytettävä tieto

- ylempi johto – lokitiedon vaatimuksien täyttyminen, tarpeellisen tiedon määrittely

Tyypillisesti lokienhallinta ei ole toteutettu organisaation sisällä keskitetysti ja eri rooleista vastaavat tiimit pitävät itse huolen lokitiedon käsittelystä. Lokienhallinnan keskittämisellä kokonaisuutta on helpompi hallita, sillä samat säännöt pätevät kaikkiin osallisina oleviin rooleihin. Keskitetyn järjestelmän avulla lukuoikeutta eri lokitietoihin voidaan jakaa ja rajoittaa pienemmällä vaivalla roolista riippuen [10;11].

Organisaation täytyy etukäteen suunnitella kuinka paljon lokianalyysiä suoritetaan tietyllä roolitasolla. Työn kokonaismäärä voi kasvaa suureksi, jos analyysiprosessia ei jaeta järkevästi. Eri roolit näkevät tapahtumat eri kontekstissa, mikä korostaa näiden yhteistyön tärkeyttä lokienhallinnan hienosäädön kannalta. Tärkeintä on löytää lokitiedon sisällöstä tyypit, joita tietty rooli pystyy parhaiten ymmärtämään. Tehokkaan lokienhallinnan edellytyksenä on myös eri roolien koulutus lokitiedon hyödyntämiseen ja oman osansa ymmärtämiseen.

3.4.2 Lokituksen vaatimukset ja tavoitteet

Lokienhallinnan tärkein perusta on järjestelmän tavoitteiden ja vaatimuksien määrittely. Nämä määrittelyt sisältävät voimassa olevat lait, ohjesäännöt sekä organisaation omat

säännöt. Lokien käsittely tulee optimoida siten, että keräämisen ja analysoinnin asettamat tavoitteet täyttyvät ilman turhaa resurssien tuhlausta. Lokitietojen avulla voidaan jäljittää tietoverkon tapahtumia, virheitä, väärinkäyttö- ja tietomurtoyrityksiä sekä näiden yrityksiä. Organisaation lokijärjestelmän tavoitteiden ja vaatimuksen määrittelyssä tulee olla vastaukset seuraaviin seikkoihin:

Lokin luonti

- Minkä tyyppisiä lokilähteitä järjestelmään otetaan?
- Mitkä lokilähteen komponentit tallennetaan (käyttöjärjestelmä, palvelu, sovellus)?
- Minkätyyppisiä tapahtumia tallennetaan?
- Mitä tietoa tietueisiin tallennetaan?
- Kuinka usein tietoa tallennetaan?

Lokin siirto

- Minkätyyppinen tieto siirretään lokijärjestelmään?
- Kuinka tieto siirretään lokijärjestelmään?
- Kuinka usein tieto siirretään lokijärjestelmään?
- Kuinka lokidatan saatavuus, luotettavuus ja eheys turvataan siirron aikana?

Lokin säilytys

- Minkälainen lokikierto (säilöminen ja arkistointi)?
- Kuinka lokitieto suojataan?

- Kuinka pitkään tietty tapahtumaloki säilytetään?
- Kuinka paljon levytilaa lokitiedolle allokoidaan?

Lokianalyysi

- Kuinka usein tietty tapahtumaloki analysoidaan?
- Mitä analysoinnilta haetaan?
- Kenellä on oikeus päästä käsiksi lokitietoon?
- Kuinka toimia poikkeuksellisten lokitapahtumien kanssa?
- Kuinka arkaluontoisen lokitiedon kuten salasanojen kanssa menetellään?

Lokienhallinnan täytyy kokonaisuutena pystyä esittämään riittävän tarkat tiedot valvonnasta olevista prosesseista. Perustietoja ovat tapahtuma, tapahtuman onnistuminen, tapahtuman suorittaja sekä ajankohta. Tarkan tiedon lisäksi järjestelmän tulee auttaa proaktiivisissa toiminnoissa, jotta organisaation infrastruktuuri pysyy jatkuvasti kunnossa. Tärkeitä seikkoja relevantin lokitiedon löytämisessä on keskittyä vähiten tavallisiin viesteihin, sillä nämä ovat poikkeamia normaalista organisaation toiminnasta. Virheisiin ja vastaaviin viittaavat viestit sekä epätavallisten viestin määrän yhtäkkinen nousu on viite tärkeästä lokitiedosta [10;15].

Erilaiset lait asettavat lisävaatimuksia lokitiedon käsittelyyn. Tämä erityisesti silloin, kun kyseessä on tunnistetietoja ja henkilötietoja lokittava lähde. Kaikkea lokitietoja tämä ei koske, eli tässäkin mielessä on ensiarvoisen tärkeää tuntea kerätyn lokitiedon sisältö. Lokitietojen käsittelyyn liittyviä lakeja Suomessa ovat muun muassa henkilötietolaki, julkisuuslaki ja sähköisen viestinnän tietosuojalaki [9].

Tavoitemäärittelyjen jälkeen luotuja sääntöjä täytyy ylläpitää tulevien kokemuksien mukaisesti. Esimerkiksi tallennettavan lokitiedon määrän optimointi tarpeelliseksi

koettujen tietueiden perusteella. Organisaation sisällä tapahtuvien mahdollisten muutoksien varalta lokimäärittelyjen joustavuus on myös tärkeässä osassa.

3.4.3 Lokienhallinnan infrastruktuuri

Lokienhallinnan infrastruktuuri koostuu laitteistosta, ohjelmistosta sekä tietoverkoista jotka yhdessä luovat, kuljettavat, säilövät ja analysoivat lokitietoa. Useimmilla organisaatioilla on yksi tai useampi lokienhallintainfrastruktuuri, jonka merkittävimmät toiminnot voidaan jakaa seuraaviin tasoihin:

- lokin luonti – lokidatan tuottavat laitteet
- lokin analyysi ja säilytys – yksi tai useampi palvelin, jonne lokidata kuljetetaan
- lokin monitorointi – käyttöliittymä lokitiedon tutkimiseen

Liikenne eri tasojen välillä usein tapahtuu organisaation normaalin tietoverkon välityksellä, sillä tarkkailtavat komponentit sijaitsevat luonnollisesti tässä ympäristössä. Eristetty lokituksen liikenteelle varattu fyysinen verkko on myös mahdollinen vaihtoehto, jos halutaan maksimoida tietoturva ja liikennemäärä kasvaa hyvin suureksi. Julkisen tietoverkon kautta siirrettävä lokitieto lähetetään salattuna ja tunneloituna, millä taataan siirrettävän lokitiedon luottamuksellisuuden säilyminen [10].

Lokienhallinnan toisen tason toteutus vaihtelee rakenteeltaan. Yksinkertaisimmillaan se käsittää yhden palvelimen, joka hoitaa määritellyt toiminnot, mutta usein tähän on varattu kaksi tai useampi palvelinta. Roolit voi esimerkiksi olla jaettu siten että toinen palvelin hoitaa analyysin ja lyhytaikaisen säilytyksen kun toiselle lokitieto siirretään pidempiaikaista tallennusta varten. Useamman palvelimen ratkaisut tukevat myös klusterointia, jolloin yhden yksikön vikatilanne ei aiheuta koko lokiprosessin keskeytymistä. Lokidatan keräämistä varten palvelimia voi olla myös useampaa kapasiteettia. Esimerkiksi yksi yleinen skenaario on pienemmät kerääjät tietoverkon reunoilla, jotka lähettävät lokitietoa suuremmalle keskuspalvelimelle.

3.4.4 Lokienhallinnan toiminnot

Lokienhallintaan kuuluu eräitä perustoimintoja, jotka auttavat lokitiedon prosessoinnissa kolmella aikaisemmin mainitulla tasolla. Nämä toiminnot eivät muuta itse lokitietoa, mutta tuovat sen helpommin saataville. Lokienhallintajärjestelmät käyttävät kaikkia tai useimpia toimintoja seuraavista:

- Lokin jäsennys on lokitiedon lukemista lokilähteeltä. Esimerkiksi riveittäin luku etukäteen määrätystä tiedostosta.

- Lokitiedon suodatus tarkoittaa relevantin tiedon erittelyä.

- Lokikierto on toimenpide, kun vanha lokitiedosto suljetaan ja uusi avataan. Menetelmällä pyritään suojaamaan itse lokitietoa sekä pitämään tiedostokoot normaaleina. Prosessi on automaattinen ja sille konfiguroidaan suunnitteluvaiheessa oikeanlaiset arvot suhteutettuna järjestelmän kapasiteettiin.

- Arkistointi on lokitiedon siirto pidempiaikaiseen säilytykseen. Tätä lokitietoa ei käydä aktiivisesti läpi.

- Lokitiedon pakkaus on tyypillisesti arkistoinnin yhteydessä tehtävä toimenpide levytilan säästämiseksi.

- Lokitiedon konvertointi on yleensä jäsennyksen ohessa tehtävä lokitiedon muunto lokienhallintajärjestelmän käyttämään muotoon. Tässä vaiheessa mahdollisesti eriävät lokitietueet normalisoidaan ja niihin voidaan lisätä esimerkiksi aikaleima.

- Eheys turvataan kerätystä lokidatasta luodulla tiivisteellä, johon viittaamalla tieto voidaan todentaa. Suosittuja tiivisteitä ovat muun muassa MD5 sekä SHA1.

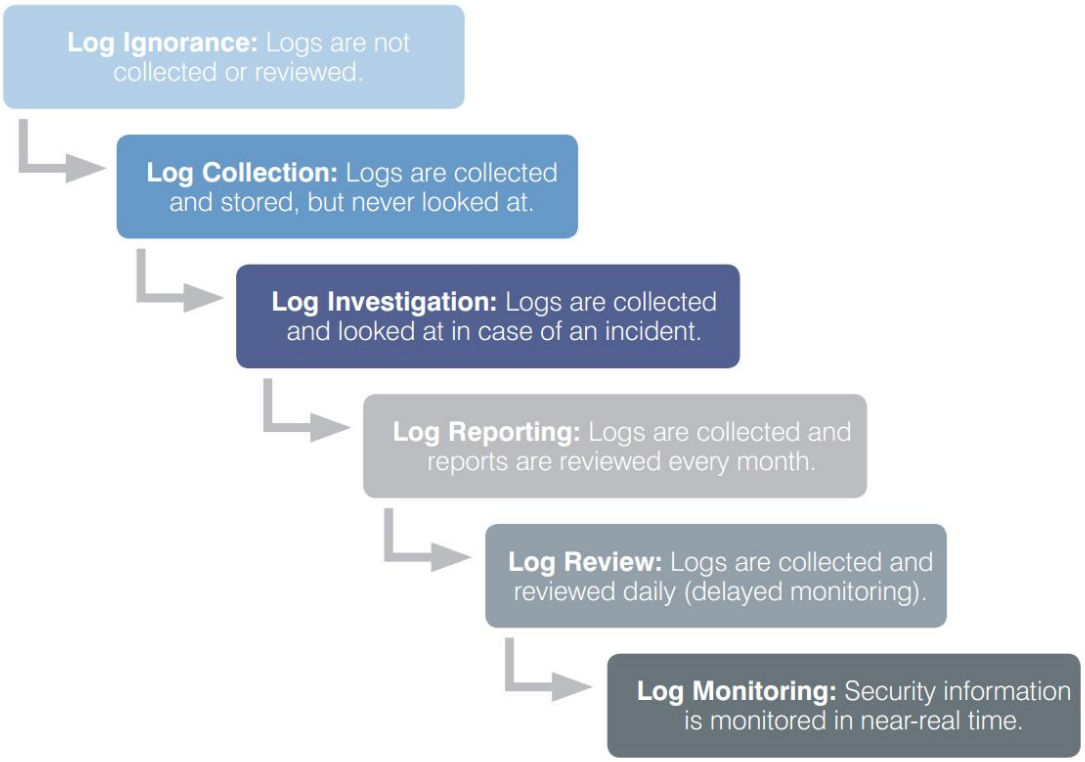
- Tapahtumien korrelaatio tarkoittaa lokitiedon tapahtumien yhtäläisyyksien löytämistä. Esimerkkejä ovat IP-osoitteet, käyttäjätunnukset sekä aikaleimat.

- Lokin tarkastelu on lokitiedon näyttämistä käyttäjän ymmärtämässä muodossa.

- Raportointi tarkoittaa tietyltä ajalta keräytyneen lokitiedon pohjalta luotua esitystä.

- Tietyn ajan jälkeen tarpeettomaksi jäänyt lokitieto hävitetään uuden tieltä.

Lokienhallintajärjestelmän käyttöönotto etenee yleensä organisaatiossa vaiheittain. Aluksi seurantaan voidaan esimerkiksi liittää tietoverkon palomuurit, joiden kanssa lokitiedon käsittelyä voidaan testata ennen syvempää toteutusta. Hälytyksiin reagoimisen ja yleisen säännöstön tultua tutummaksi toteutusta laajennetaan muihin hallinnassa oleviin prosesseihin [11].



Kuva 7. Lokienhallinnan eri vaiheet [11].

Kuvasta 7 voidaan nähdä yleiset lokienhallinnan vaiheet organisaatiossa. Mitä pidemmälle toteutuksessa mennään, sitä nopeammin mahdollisiin poikkeamiin pystytään reagoimaan. Lokienhallinnan kehittämisessä tärkeässä osassa ovat sekä työkalut että selkeä vaiheittain eteneminen. Suurien organisaatioiden aiempien kokemusten perusteella onkin virheellistä ajatella, että prosessi hoituisi niin kutsutulla set it and forget it -tyyppisellä toteutuksella, sillä toimiva lokienhallinta vaatii hienosäätöä ajan kanssa [11].

Seuraavassa luvussa tarkastellaan Splunk-nimisen lokienhallintajärjestelmän rakennetta, toimintaa ja sitä, kuinka se vastaa tässä luvussa läpikäytyjä lokienhallinnan perusmääreitä.

4 Splunk

4.1 Yleistä Splunkista

Lokienhallintajärjestelmä

Lokienhallintajärjestelmäksi kutsutaan tuotetta, jolla hallitaan monen eri lähteen tuottamaa lokidataa. Erilaisia tuotteita on olemassa huomattava määrä, joiden laatu ja hinta vaihtelee kevyestä ja ilmaisesta suuriin ja raskaisiin erikseen lisensoitaviin toteutuksiin. Järjestelmän tarkoituksena on tuoda lokitiedolle säilytyspaikka ja tehokas haku. Useat laitevalmistajat tarjoavat omille tuotteilleen omistettuja ratkaisuja, jotka eivät ole yhteensopivia muiden valmistajien kanssa. Tässä luvussa läpikäytävä työkalu toimii näihin verrattuna päinvastaisesti, sillä se pystyy hyödyntämään käytännössä kaikenlaista lokidataa alustasta riippumatta [10].

Splunk yhtiönä

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 4800 asiakasta yli 85:ssä maassa. Forten lisäksi partnerikuntaan kuuluu suuria nimiä, kuten Cisco, Vmware ja Blue Coat. Nimi splunk tulee englanninkielisestä sanasta spelunk, jolla viitataan luolien tutkimiseen [20].

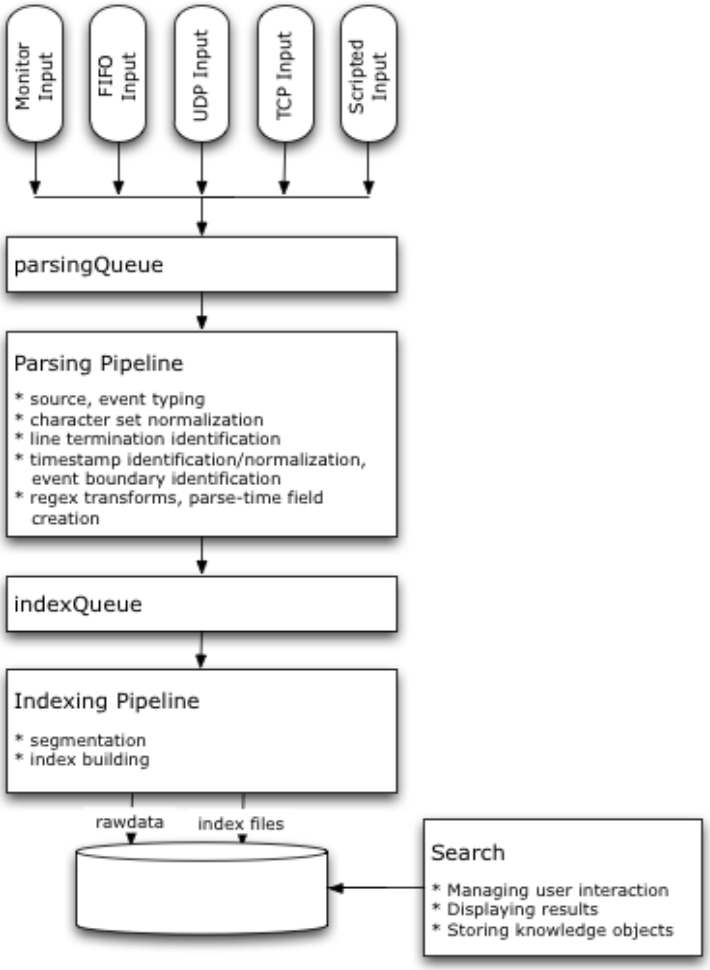
Splunk-yhtiön pääasiallinen tuote on myös niinikään Splunk-nimellä tunnettava lokienhallintatyökalu. Splunk tallentaa, indeksoi ja analysoi konetietoa, jota se voi vastaanottaa lukemattomista eri lähteistä. Tämän jälkeen työkalun avulla tiedosta voi luoda raportteja, käyttöliittymiä sekä erilaisia graafisia esityksiä. Tuotteen pyrkimyksenä on tuoda tärkeiden prosessien luomat lokitiedot yhtenäisesti ja helposti saataville tunnistamalla toistuvia kaavoja ja luomalla yhteyksiä näiden välille.

Splunk-lokienhallintatyökalun toimintamalli

Splunkin tarkoituksena on tehdä koneellisesti luodusta lokitiedosta hyödyllistä organisaatiolle. Eri järjestelmät kirjoittavat jatkuvasti toiminnastaan valtavasti dataa muistiin ja Splunk lukee tätä raakatietoa. Raakatieto jaetaan määriteltyihin tapahtumiin, joiden pohjalta Splunkin hakujärjestelmä toimii. Tieto käy Splunkin sisällä läpi usean eri vaiheen muuntuessaan alkuperäisestä lähteestä lopulliseen tapahtuma-muotoon, jota pystytään hakemaan ja analysoimaan. Tämän raakatiedon Splunkissa kulkema prosessi voidaan jakaa seuraaviin osioihin:

- syöte
- jäsennys
- indeksointi
- haku.

Syöte-vaiheessa Splunk vastaanottaa alkuperäiseltä lähteeltä raakatiedon. Tieto pilkotaan tallenteelle lohkoihin, joihin jokaiseen sisällytetään metatietoa. Nämä metatiedot pätevät kaikkeen lähteeltä tulevaan tietoon. Metatietoja ovat esimerkiksi lähde, lähteen tyyppi sekä lähteen osoite. Metatietoja ovat myös Splunkin käyttämät sisäiset merkinnät, kuten lähdetiedon kooditus sekä valitun indeksin nimi. Tässä vaiheessa Splunk ei myöskään ota kantaa itse raakatiedon sisältöön, eli tiedossa on vain yhtenäinen tietovirta tietyillä konfiguroiduilla ominaisuuksilla. Tapahtumapohjainen järjestely tapahtuu myöhemmin. Prosessia havainnollistetaan kuvassa 8 [21].



Kuva 8. Tiedon kulku Splunkissa.

Raakatiedon vastaanottamisvaiheen jälkeen tapahtuu tarkempi tarkasteluvaihe. Splunk tutkii, analysoi ja muuntaa tiedon kehittyneempään tapahtuma-muotoon. Tämän vaiheen aikana tieto siis pilkotaan niihin osiin, joita myöhemmin käsiteltävä haku palauttaa tuloksina. Jäsennysvaihe sisältää myös useamman sisäisen vaiheen, jotka voidaan jakaa seuraavasti:

- tietovirran pilkkominen yksittäisiin riveihin
- aikaleiman tunnistus, jäsentely ja asetus
- aikaisemmin määriteltyjen metatietojen lisäys yksittäisiin tapahtumiin

- tapahtuma- ja metatietojen muuntaminen erikseen määriteltyjen sääntöjen perusteella

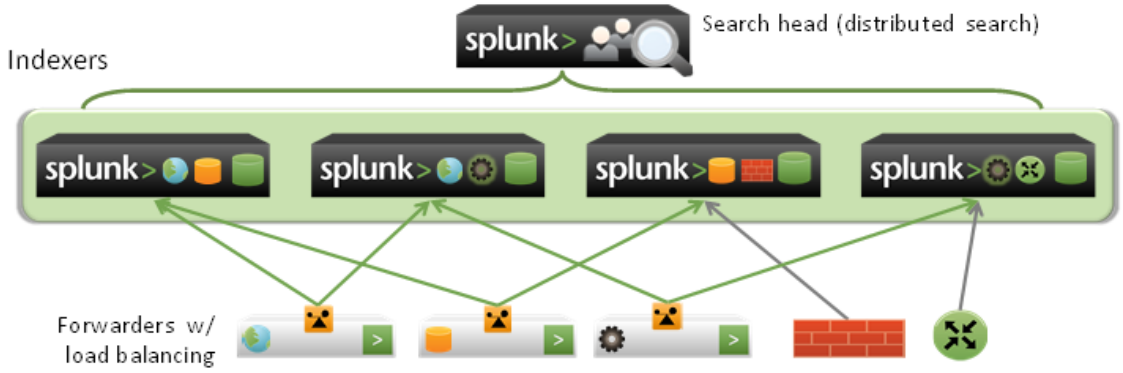
Yksityiskohtaisten tapahtumien jäsentelyn jälkeen tieto kirjoitetaan kiintolevylle indeksiin. Indeksointivaiheessa kirjoitetaan muistiin pakattu tapahtumatieto sekä siihen liittyvät indeksitiedot. Indeksitiedot sisältävät tapahtumat jaoteltuna eri segmentteihin. Indeksoinnin hyödyntäminen nopeuttaa tiedon paikallistamista huomattavasti, ja se on lokienhallintajärjestelmän oleellisimpia ominaisuuksia. Aiemmin erikseen määritelty jäsennysvaihe sisällytetään yleensä indeksointiin, mutta tiedon kulkeman kokonaisjanan kannalta osioiden erottelu on perusteltua [21].

Lopulta talteen kirjoitettua tietoa hallitaan Splunkin tietohaun avulla. Haku hyödyntää aiemmin määriteltyjä indeksejä tiedon paikallistamiseen. Hakuominaisuus on perusta sille, kuinka käyttäjä näkee ja käyttää indeksoitua tietoa. Ominaisuutta hyödyntää ajoitetut haut, raportointi, käyttöliittymät ja hälytykset [21].

4.2 Splunkin arkkitehtuuri

4.2.1 Komponentit

Yksinkertaisimmillaan Splunk toimii yhtenä instanssina yhdellä PC:llä. Tämä yksi instanssi hoitaa yksin kaikki roolit, joita edellisessä luvussa käytiin läpi. Vaikka yksinkertaisin tapa saattaa riittää pienyrityksille ja testaustarkoituksiin, todellisuudessa suuremmissa ympäristöissä instanssit jaetaan toiminnallisuuden mukaan useampaan osaan. Eri komponentit käyttävät käyttävät erisuuruisesti resursseja, joten järjestelmään voidaan esimerkiksi liittää useampi indeksoija, mutta tiedon hakuun vain yksi instanssi. Kaikki komponentit ovat pohjimmiltaan samanlaisia Splunk-ohjelmia erilaiseen tilaan konfiguroituna, mutta tarvittaessa komponenteista on olemassa myös karsittuja versioita jotka esimerkiksi vain lähettävät indekseille tietoa. Kuvassa 9 on nähtävissä komponenttien sijoittelu järjestelmässä [22].



Kuva 9. Splunkin arkkitehtuuri.

Splunkin lisensointi määritellään indeksoidun tietomäärän perusteella. Jokaisella indeksointia suorittavalla instanssilla tulee olla lisenssi. Instanssi kirjoittaa muistiin indeksoidun lokin määrän jokaiselta päivältä, eikä vuorokauden vaihtuessa tietomäärä saa ylittää lisenssin mukaista määrää. Jaetulla järjestelmällä on mahdollisuus asentaa erikseen toimiva lisenssipalvelin, joka ylläpitää kaikkien asennettujen laitteiden lisenssejä. Kokonaisuudessaan Splunkissa on neljää eri lisenssityyppiä:

- Enterprise on standardi Splunk-lisenssi, joka sisältää kaikki tarvittavat toiminnot.
- Free on ilmainen lisenssi, jossa rajoitetut toiminnot ja indeksointirajana 500Mt/päivä.
- Forwarder on tarkoitettu instansseille jotka eivät suorita indeksointia, mutta lähettävät tietoa dedikoiduille indeksoijille.
- Beta sisältää täydet toiminnot, mutta se on rajoitettu Splunkin beta-versioihin.

Hajautetun Splunk-järjestelmän perusinstanssit ovat indexer, forwarder, search head sekä deployment server. Seuraavaksi nämä komponentit käydään läpi seikkaperäisemmin [23].

4.2.2 Indeksoija

Indeksi on säilytyspaikka Splunkin käyttämälle tapahtumatiedolle. Splunk-indeksi sisältää kahdentyyppisiä tiedostoja, jotka ovat normaali tapahtumatieto pakattuna sekä indeksitiedostot, jotka sisältävät viitteitä tapahtuma- ja metatietoihin. Yhdessä tiedostot muodostavat indeksin, jossa nämä tiedostot on jaoteltu tiedon iän mukaan. Tiedon iän mukaan määritellyistä hakemistoista käytetään käsitettä bucket. Hakemistojen tilat on määritelty taulukossa 1. Yksinkertaistettuna siis indeksi koostuu joukosta hakemistoja, joiden tyyppi muuttuu ajan kanssa [22].

Taulukko 1. Bucket-tilat.

Bucket stage	Description	Searchable?
Hot	Contains newly indexed data. Open for writing. One or more hot buckets for each index.	Yes
Warm	Data rolled from hot. There are many warm buckets.	Yes
Cold	Data rolled from warm. There are many cold buckets.	Yes
Frozen	Data rolled from cold. Splunk deletes frozen data by default, but you can also archive it. Archived data can later be thawed.	No

Hakemistojen tilat vaihtuvat tiedon vanhetessa, kunnes ne lopulta poistetaan tai arkistoidaan. Aikamäärät näille tapahtumille on vapaasti konfiguroitavissa samasta tiedostosta kuin indeksit itse, eli indexes.conf.

Indeksoija-komponentti on Splunk-instanssi, joka hajautetussa järjestelmässä keskittyy ainoastaan tulevan tiedon indeksoimiseen ja omaan indeksiin kohdistuneisiin hakuihin. Indeksoijia voi myös asentaa useamman rinnakkain, jolloin yhdistelmää kutsutaan klusteriksi. Klusterin sisällä jäsenet kahdennetaan vikasietoisuutta varten. Klusteriin kuuluu kolme erityyppistä yksikköä seuraavasti:

- Master-yksikkö, joka hallitsee klusterikokonaisuutta.

- Peer-yksiköt hoitavat indeksointia ja niille osoitettuja hakuja. Yksikköjen välillä on automaattinen failover-toiminto, mikä turvaa jatkuvan tiedon indeksoinnin, vaikka yksikön yhteys järjestelmään häviäisi.

- Yhdellä tai useammalla haku-instanssilla koordinoidaan yksityiskohtaiset haut klusterille.

Yleisenä suosituksena yhdelle indeksoijalle pidetään 100 gigatavua käsiteltyä tietoa päivässä. Tämä sisältää kahdesta kolmeen samanaikaista käyttäjää tekemässä perushakuja. Jos näiden arvojen yli mennään tai instanssilla hoidetaan jatkuvasti syvempiä hakuja ja raportteja, on toisen indeksoijan lisääminen perusteltua [22].

4.2.3 Forwarder-komponentti

Forwarder-komponentiksi kutsutaan Splunk-instanssia, joka lähettää tietoa toiselle forwarderille tai indeksoijalle. Vastaanottavaa päätä kutsutaan sanalla receiver. Tyypillisesti nämä komponentit suorittavat aiemmin tutuksi tullutta Splunkin syöte-vaihetta, jossa tietovirta edelleenlähetetään indeksoijalle jäsennystä varten. Tämä toimii esimerkiksi tilanteessa, jossa on joukko Windows- ja Linux-pohjaisia lokitietoa kirjoittavia laitteita, joiden tiedot täytyy saada samalle indeksoijalle.

Forwarder-instansseja on kolme erilaista:

- Universal forwarder on kevyt ja riisuttu versio Splunkista, joka sisältää ainoastaan valmiudet tiedon edelleenlähettämiseen.

- Heavy forwarder on normaali Splunk-instanssi, jonka toimintoja on suljettu resurssien säästämiseksi. Tämä instanssi pystyy tarvittaessa esimerkiksi indeksoimaan tietoa paikallisesti. Paikallisesti tapahtuvan jäsennyksen ansiosta tietoa voidaan reitittää eri tavoin tapahtumien perusteella.

- Light forwarder on kevyt instanssi, joka versiosta 4.2 lähtien on korvautunut universal forwardilla. Tuki tälle löytyy lähinnä vanhentuneita järjestelmiä varten.

Forwarder-komponentti pystyy tyypistä riippuen edelleenlähettämään raakaa, jäsentämätöntä sekä jäsennettyä tietoa. Raa'an tiedon lähetys toimii suoraan TCP:llä ilman Splunkin yhteysformaattia. Tästä syystä tiedon lähetys onnistuu myös toisiin järjestelmiin, jotka eivät välttämättä ole Splunk-pohjaisia. Kaksi ensimmäistä tapaa

onnistuu kaikilla forwarder-tyypeillä, mutta jäsennettyyn tietoon vaaditaan heavy forwarder. Forwarder-asetukset ovat vapaasti muokattavissa tiedostossa outputs.conf [22].

4.2.4 Search head -instanssi

Hajautetussa Splunk-järjestelmässä on erityinen instanssi hakujen koordinointiin. Käyttäjät kirjautuvat web-pohjaiseen käyttöliittymään, joka sijaitsee tällä instanssilla. Search head -instanssilla ajettu haku lähettää pyynnön järjestelmään kuuluville indeksoijille, jotka palauttavat hakutuloksen käyttäjälle. Search head -yksikköön yhteydessä olevia indeksoijia kutsutaan nimityksillä search peer tai indexer node. Jos kyseessä on klusteri, hakuinstanssin avulla pystytään myös jakamaan yksittäisen indeksoijayksikön kuormaa levittämällä haku useammalle yksikölle, mikä helpottaa suuren tietomäärän prosessointia. Klusteriin kuuluvaa indeksoijaa kutsutaan nimellä peer node.

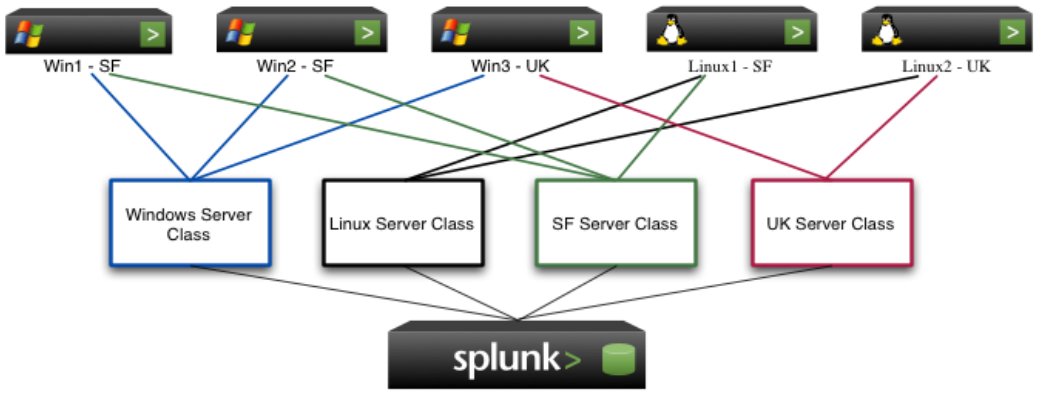
Hakuinstanssilla voidaan myös rajoittaa käyttäjien oikeuksia päästä käsiksi tietoon. Esimerkiksi tietyn sivukonttorin käyttäjät voidaan rajoittaa vain omien tietojen näkemiseen, vaikka hakutoiminnot on toteutettu keskitetysti. Search head voidaan myös konfiguroida käyttämään vain tiettyjä indeksoijia, jolloin näiden ulkopuolisiin ei saada edes yhteyttä.

Hakuprosessissa search head kopioi paikalliset hakuobjektit yhteydessä oleviin indeksoijiin. Hakuobjektit sisältävät muun muassa tallennettuja hakuja ja tapahtumatyyppejä. Indeksoijat käyttävät näitä tietoja hyväksi hoitaessaan hakuyksikön käskyjä. Hajautetussa ympäristössä indeksoijilla itsellään ei ole mitään paikallisia tallennettuja hakuobjekteja. Tämä tarkoittaa myös sitä, että search head yksikölle asennetut lisäosat kopioituvat automaattisesti indeksoijille. Sama pätee myös käyttöoikeuksiin, eli ne määritellään ainoastaan hakuyksiköllä [22].

4.2.5 Deployment server -komponentti

Deployment server on Splunkin käyttämä komponentti hajautetun järjestelmän yksiköiden ylläpitoon. Sen avulla voi keskitetysti päivittää konfiguraatioita sekä sisältöpäivityksiä mille tahansa muulle komponentille, oli se forwarder, indeksoija tai search head. Usein deployment server -komponentin käyttötarkoituksena on hallita forwardereita, joiden yksittäinen ylläpito on hankalaa.

Jotta Splunk-instansseja pääsisi konfiguroimaan keskitetysti, täytyy etäinstanssi ensin konfiguroida asiakaslaitteeksi. Konfiguroinnin jälkeen asiakaslaite kyselee palvelimelta tietyin väliajoin päivityksiä määritellyille ominaisuuksille. Jos päivitys löytyy, palvelin kertoo siitä asiakkaalle, joka konfiguroinnista riippuen asentaa päivityksen. Asiakkaita voi tarvittaessa liittää myös erilaisiin palvelinluokkiin. palvelinluokat mahdollistavat samantapaisten asiakaslaitteiden asetusten ryhmittämisen. Kuvassa 10 näkyy palvelinluokkien malli käytännössä [22].



Kuva 10. Deployment server –rakenne.

4.3 Toiminnot

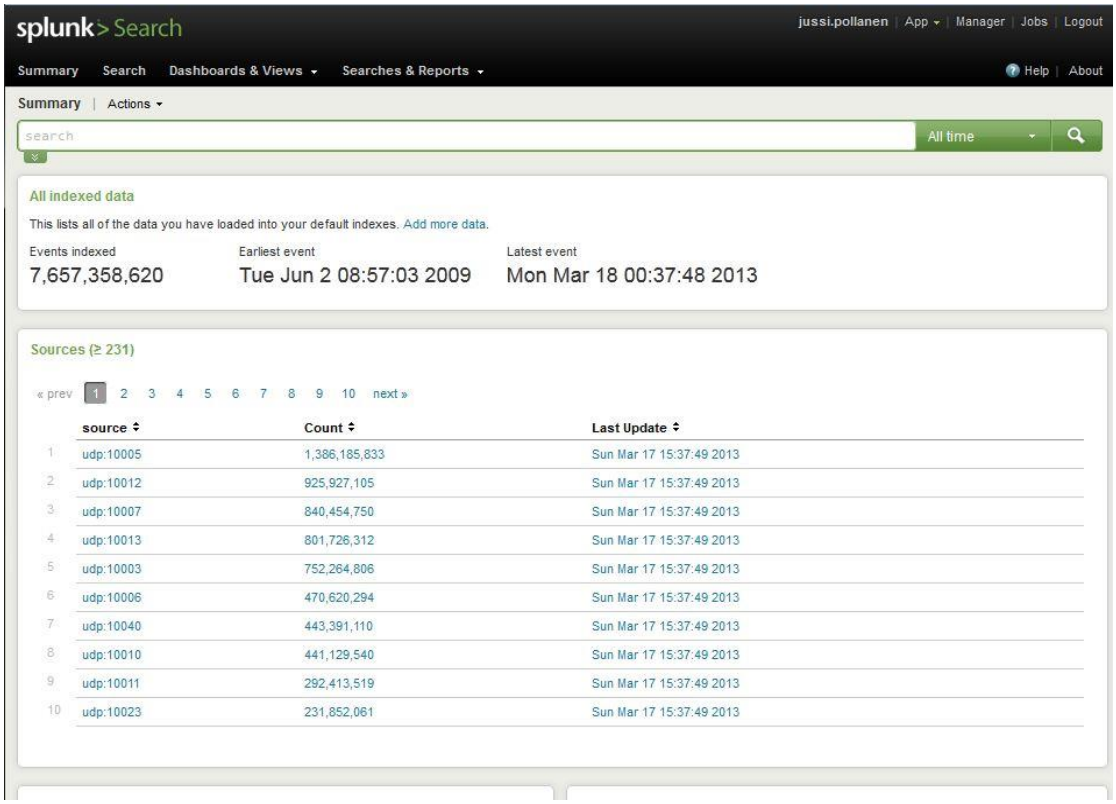
4.3.1 Sovellukset

Splunkin pääasiallinen käyttö tapahtuu valitulla instanssilla pyörivän Splunk Web -taustaprosessin kautta. Normaalisti se toimii search head -yksiköllä, mutta on mahdollinen myös mahduttaa mille tahansa indeksoijalle. Hallinta onnistuu paikallisesti myös komentoriviltä. Splunk Webin päällä pyörii sovelluksia (App), joista yleisin on

oletuksena löytyvä haku. Splunkin virallisilla internet-sivuilla on myös monia sovelluksia, jotka sisältävät valmiiksi määriteltyjä hakuja ja raportteja esimerkiksi Linuxiin tai Windows palvelimen Active Directoryyn liittyen. Seuraavissa luvuissa tutkitaan Splunk Webin kautta Splunkin ydinsovelluksia, joihin muiden sovelluksien toiminta perustuu.

4.3.2 Haku

Haun päänäkymästä (kuva 11) näkee käyttäjälle oletuksena määriteltyjen indeksien suurpiirteisen sisällön. Näkymästä näkee heti, kuinka paljon indeksiin tai indekseihin on tallennettu tapahtumia ja miltä aikaväliltä. Indeksin sisällöstä näkee lähteen (source), lähdetyypin (source type) ja lähteen osoitteen (host). Lähteellä tarkoitetaan lokilähdettä, joka voi olla esimerkiksi tiedosto, tiedostovirta (TCP/UDP) tai Windows-rekisteri. Lähdetyypissä määritetään tiedon tyyppi tai formaatti, kuten esimerkiksi palomuurin liikenne. Lähteen osoite voi olla joko IP-osoite tai konfiguraatiosta riippuen DNS:n kautta saatu nimi [13].



Kuva 11. Haku-näkymän etusivu.

Haku palauttaa sitä vastanneet indeksiin aiemmin tallennetut tapahtumat. Ilman haussa tehtyä kentän erikseen määrittelyä kaikki tapahtumaan tallennettu tieto on haettavissa, eli esimerkiksi hakukenttään kirjoitettu error palauttaa jokaisen indeksin sisällä olevan tapahtuman, jossa on mainінta tästä sanasta. Kyseinen haku on kuitenkin hyvin epätarkka ja kuormittava, joten sellaisen läpikäyminen suuressa ympäristössä on hankalaa ja hidasta [13].

Splunk käyttää haun optimoimiseen eräänlaista hakukieltä nimeltä SPL (Search processing language). Menetelmä antaa mahdollisuuden putkittaa komentoja yhteen eli suorittaa edellisen määrittelyn tuloksille uusi. Voidaan esimerkiksi suorittaa seuraavanlainen haku:

```
index=”fortigate” host=”10.10.10.1” dns_name=”*” | top limit=5 dns_name
```

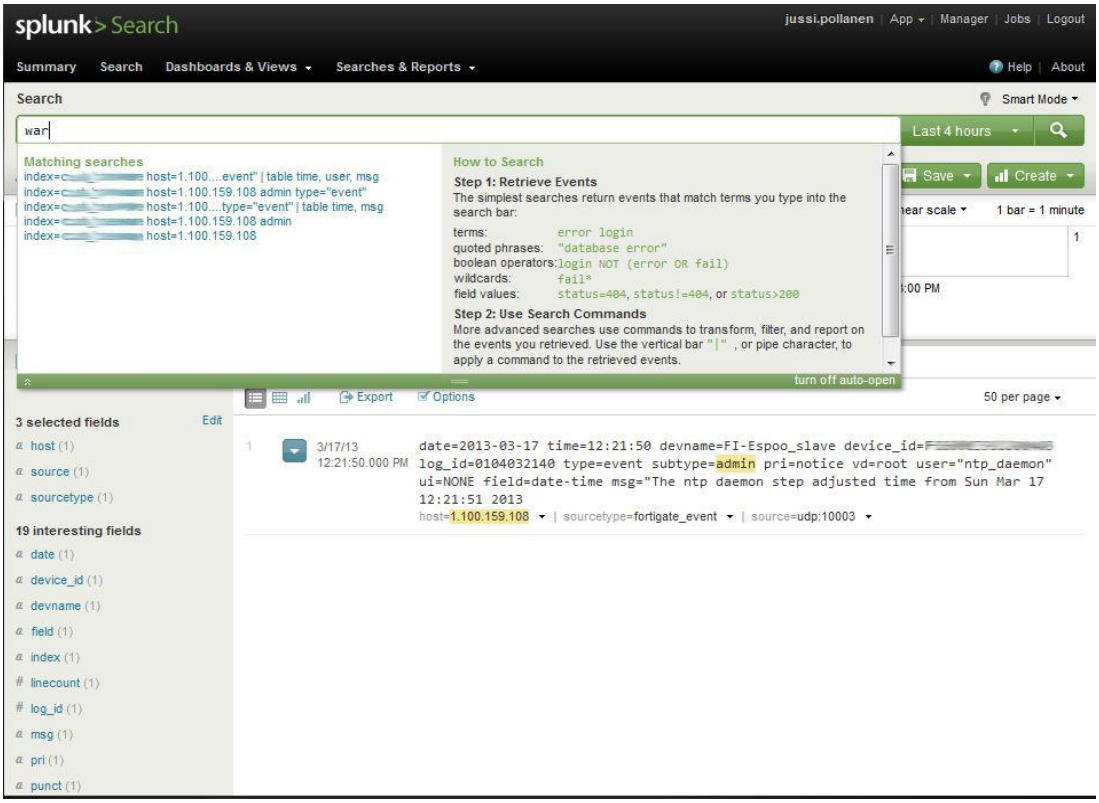
Haun ensimmäisessä osassa haetaan indeksistä fortigate, osoitteesta 10.10.10.1 kaikki tapahtumat, jotka sisältävät dns_name-kentän. Näistä saaduista tuloksista tehdään top-komennon avulla viiden suosituimman DNS-haun lista, joka tulostetaan käyttäjälle. Hauissa toimivat myös normaalit Boolen lausekkeet. Edelläolevassa haussa Boolen hakuehto AND on implikoitu, sillä haku palauttaa vain tapahtumat, joissa on kaikki määritellyt kentät. Haun alkuun on myös lisätty automaattisesti komento search, joka aloittaa itse haun. Putkitetuissa osissa search-komento on määriteltävä uudestaan, jotta toinen haku suoritetaan. Alahaussa, eli hakasulkeissa määritellyn yhden osan sisällä tehdyssä haussa komento on myös erikseen määriteltävä [13;24].

Haun palauttamia tapahtumia voi hallita usein eri tavoin. Seuraavassa on muutamia yleisiä komentoja hakutulosten hallintaan:

- sort – palautettujen tapahtumien rajoitus ja uudelleenjärjestely
- dedup – duplikaattien tapahtumien poisto tuloksista
- head, tail – ensimmäiset ja viimeiset määritellyt tapahtumat

- top, rare – yleisimmät ja harvinaisimmat tapahtumat
- transaction – samantapaisten tapahtumien ryhmittely
- chart, timechart – taulukkojen luonti
- fields – kenttien poisto hakutuloksista
- eval – mahdollistaa laskutoimitusten tulosten sisällyttämisen kenttään.

Oletuksena Splunkin haku tekee haun koko tallennetun tiedon ajalta. Aikarajaa pystyy säätämään hakumäärittelyn lisäksi suoraan hakupalkista löytyvän valikon alta. Haun jälkeen Splunk järjestää tapahtumien määrän aikajanalle, jota painamalla pystyy myös rajaamaan tuloksia. Historiatiedon lisäksi Splunkista löytyy reaaliaikainen tapahtumienseuranta. Tämä käsittää näkymän, johon tapahtumat päivittyvät sitä mukaan kuin niitä tallennetaan indeksiin. Haku antaa myös reaaliajassa ehdotuksia ja tietoa käytettävistä termeistä (kuva 12).



Kuva 12. Haku-työkalun komentojen täydennys.

Luotuja hakuja voidaan tallentaa muistiin myöhempää käyttöä varten. Tallentaminen voi tapahtua ryhmä- tai käyttäjäkohtaisesti. Samalla hakuja voi kellottaa automaattisiksi tietyin väliajoin. Käyttäjä voi myös jakaa omatekemiään hakuja muiden käyttäjien kanssa. Näitä itse tallennettuja hakuobjekteja kutsutaan käsitteellä knowledge object [24].

Splunk-haun ollessa käynnissä prosessi etsii tapahtumista eriteltyjä kenttiä. Oletuskenttien (lähde, lähdetyyppi, osoite) lisäksi se automaattisesti erottelee kenttiä, jotka näkyvät käyttäjälle hakutulosten vieressä. Jokaista kenttää voi erikseen painaa, jolloin käyttäjä saa välittömästi kenttää koskevaa statistiikkaa. Hakuun voidaan myös erikseen lisätä uusia kenttiä. Kenttien hallintaan liittyvät tägit (tag). Tägit ovat ryhmiä, johon on lisätty yksittäisiä kenttiä. Tämä mahdollistaa samanlaisten kenttien ja näiden arvojen yhdistämisen saman nimen alle. Esimerkiksi palvelimien nimet voivat poiketa selvästi toisistaan, mutta tägin avulla ne saavat yhteisen merkityksen.

4.3.3 Visualisointi

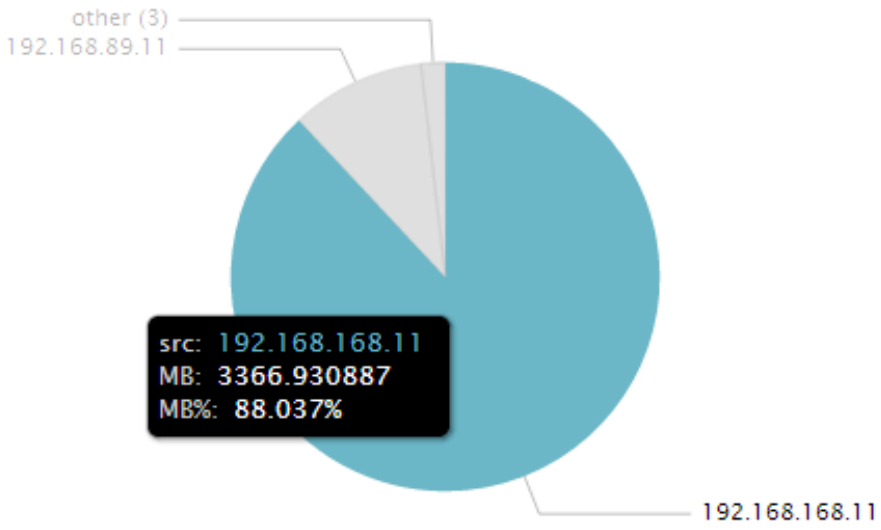
Splunkissa on useita eri tapoja näyttää saadut hakutulokset graafisena esityksenä. Ennen graafin piirtoa käsiteltävä tieto täytyy kuitenkin esittää oikeassa muodossa. Kaaviot esimerkiksi vaativat tiedot jaettuna vähintään kahteen sarakkeeseen, jolloin ensimmäisen arvot asetetaan x-akselille ja toisen y-akselille. Lisäsarakkeet voidaan lisätä sarjaan y-akselille esimerkiksi arvojen vertailua varten. Sarakemuotoon tieto saadaan muun muassa komennoilla stats ja chart [13].

```
index=fortigate host=10.10.10.1 rcvd="*" | chart sum(eval(rcvd/1024/1024)) as MB
by src | sort 5 MB desc
```

Edellä oleva esimerkkihaku hakee määritellyltä laitteelta tapahtumat, joissa on maininta vastaanotetun paketin bittimäärän koosta. Yhteenlasketulle bittimäärälle tehdään tarvittavat jakolaskut, jotta arvo saadaan muunnettua megatavuiksi. Lopuksi listataan viisi eniten kaistaa käyttänyttä lähdeosoitetta, joista muodostetaan kuvassa 13 näkyvä graafinen esitys.

Käyttöliittymästä löytyvän results chart -option avulla voidaan valita eri esitystapoja. Valittavana ovat seuraavat tyypit:

- pylväsdiagrammi
- viivadiagrammi
- ympyrädiagrammi
- palkkidiagrammi
- hajontakaavio
- arvoa kuvaava mittaridiagrammi.



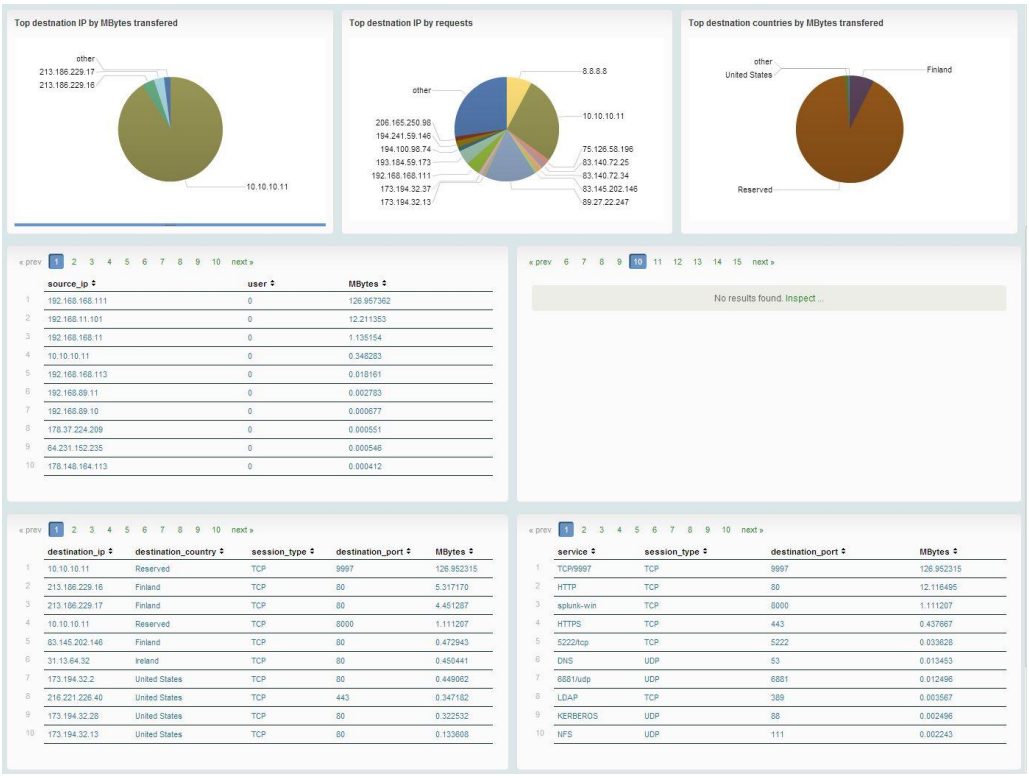
Kuva 13. Haun perusteella tehty ympyrädiagrammi.

Jokainen Splunkin web-käyttöliittymällä selattava sivu on oma määritelty näkymä. Esimerkiksi hakutoiminnon etusivu on sovelluksen oletuksena mukana tuleva näkymä. Ylläpitäjä voi itse luoda omia näkymiä ja jakaa näitä valituille käyttäjille tai käyttäjäryhmille. Näkymät koostuvat niin kutsutuista kojelaudoista sekä interaktiota

tarjoavista lomakkeista. Dashboardilla tarkoitetaan kokoelmaa erikseen määriteltyjä taulukkoja, kaavioita ja tekstiä, jotka muodostetaan valmiin haun perusteella [25].

Kojelauta on mahdollista toteuttaa suoraan web-käyttöliittymältä löytyvällä kojelaudan muokkaus-työkalulla tai muuttamalla näkymän XML-tiedostoa. Ratkaisevana erona on muokattavuus, sillä valmis työkalu on rajoittuneempi ulkoasun määrittelyn suhteen. Valmiilla työkalulla tehtyihin näkymiin ei myöskään voi sisällyttää lomakkeita. Tavallisesti kojelautaan liitetyt haut suoritetaan, kun käyttäjä avaa näkymän, jolloin tieto on aina ajan tasalla. Kuvassa 14 on esitetty esimerkkimalli kojelaudasta.

XML-tiedostoa muokkaamalla näkymän ulkonäköön pystyy vaikuttamaan suoraan HTML- ja CSS-muotoilulla. Piirrettyihin kaavioihin on myös mahdollista sisällyttää lisähakuja, jotka käynnistyvät elementtiä painamalla. Esimerkiksi kaistankäyttöä kuvaavan kaavion lähdeosoitetta painamalla voidaan tulostaa eritelty kaavio käytetyistä verkkoprotokollista [25].

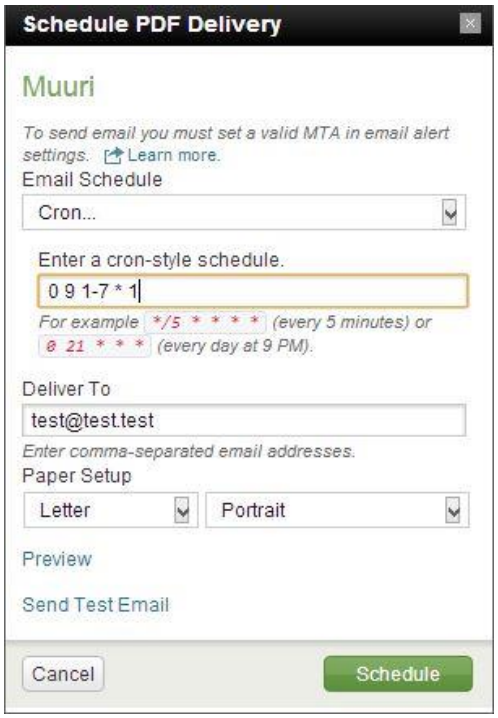


Kuva 14. Hauista muodostettu dashboard.

4.3.4 Raportointi ja hälytykset

Alaluvussa 4.5.3 käydyistä näkymistä on mahdollista tuoda dokumentteja järjestelmän ulkopuolista esitystä varten. Raakatiedot voidaan tuoda CSV- XML- tai JSON-muodoissa. Formaatit ovat tiedonsiirtomuotoja, jotka sisältävän tiedon jäsenneltynä tekstimuodossa. Näitä formaatteja tukevat ohjelmat voivat suoraan hyödyntää tiedostoihin tallennettua tietoa. Normaaliksi, luettavaksi tarkoitettuun muotoon näkymän voi viedä joko PDF-muotoon tai suoraan tulostimelle.

Yksinkertaisimmillaan PDF-raportin tekeminen tarkoittaa halutun näkymän valitsemista ja sivulta löytyvän Generate PDF -toiminnon käynnistämistä. Usein raportit kuitenkin lähetetään säännöllisin väliajoin, esimerkiksi kuukausittain. Tämä onnistuu asettamalla näkymälle ajastettu PDF-tuonti, jonka avulla voidaan lähettää määritettyyn sähköpostiin tiettynä aikana kopio raportista. Aikavälin voi valita valmiista vaihtoehdoista tai määritellä sen itse käyttämällä standardia cron-ajastuspalvelua. Cron-ajastimella määritetään tapahtuma arvottain seuraavassa järjestyksessä: minuutit, tunnit, päivä, kuukausi, viikonpäivä. Kuvan 15 esimerkissä sähköposti lähetetään joka kuukauden ensimmäisenä maanantaina kello 9.00 [24].



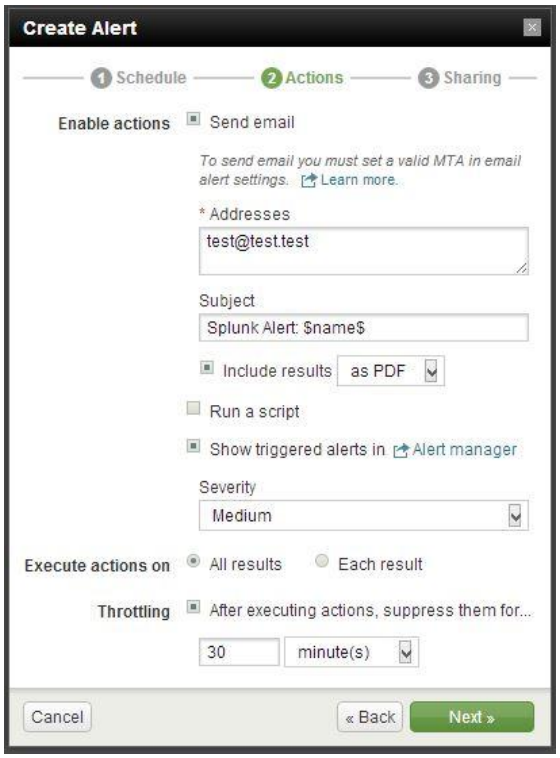
Kuva 15. Sähköpostiraportin ajastus.

Ajastuksen jälkeen järjestelmä lähettää määritellyin väliajoin raportin niin kauan, kunnes ajastus kytketään pois päältä. Raporttien tulostukseen liittyy tiettyjä rajoitteita, jotka pitää ottaa huomioon näkymiä suunnitellessa. Jos näkymässä on käytetty kehittyneitä XML-muokkaustapoja ja lomakkeita, tulostus ei onnistu. Kehittyneillä XML-muokkaustavoilla tarkoitetaan eri moduulien käyttöä, toimintojen kuten automaattisen haun tauottamisen muokkausta sekä reaaliaikaisia näkymiä.

Ajastettujen ja manuaalisten raporttien lisäksi on mahdollista määritellä hakuja, jotka tietyn tapahtuman käydessä toteen lähettävät siitä hälytyksen. Splunkissa on hälytyksille kolme erilaista kategoriaa, jotka ovat seuraavat:

- Reaaliaikainen haku laukaisee hälytyksen tietyn tapahtuman löytyessä.
- Vanhalle tiedolle ajettava haku laukaisee hälytyksen tiettyjen ehtojen käydessä toteen.
- Reaaliaikainen haku ajetaan tietyin väliajoin tietyn ajan verran. Hälytys laukaistaan haun loputtua jos määritellyt ehdot käyvät toteen.

Toisin kuin sähköpostiraportin ajastuksessa, hälytys liitetään tiettyyn hakuun eikä näkymään. Reaaliaikaisiin hakuihin perustuvissa hälytyksissä on myös mahdollista asettaa aika-arvoja, joiden puitteissa hälytys lähetetään. Esimerkiksi kriittisen virheilmoituksen tapahtuessa hälytys lähetetään, mutta vain kerran puolessa tunnissa (kuva 16). Ajoitettujen hakujen perusteella tehtävät hälyt toimivat useimmiten periaatteella, jossa määritetty määrä tapahtumia kuten epäonnistuneet kirjautumisyritykset ylittävät kriittisen määrän [24].



Kuva 16. Esimerkki hälytyksen määrittelystä.

Aktiivisten hälytysten tiloja voi seurata web-käyttöliittymältä löytyvällä alert manager-sovelluksella. Sovellus listaa kaikki aktiiviset hälytykset ja niihin liittyvät tiedot. Listalta voi itse poistaa hälytykset, tai niille voi asettaa automaattisen vanhentumisajan.

4.4 Järjestelmän esimerkkitapauksia

Järjestelmäympäristö

Testatussa ympäristössä on liitetty UDP-tietovirran avulla UTM-laitteita Splunk-järjestelmään. Haun ja tiedon hallinnan vuoksi laitteet ovat lajiteltu kirjoittamaan tietoa ennalta määriteltyihin indekseihin. Tieto kulkee indeksoijalle IPsec-tunnelia pitkin tehokkaan salauksen vuoksi. Ympäristössä on myös Windows- ja Linux-palvelimia, joille on asennettu Splunk forwarder tapahtumatietojen välittämiseksi indeksoijalle.

Todennuksen valvonta

Organisaation tietoverkossa sijaitsee useita laitteita, joten käytössä on myös keskitettyjä todennuspalvelimia. Tämä helpottaa todennuksen valvomista, sillä käyttäjätunnusten muoto ei muutu ja päätelaitteiden konfiguraatiot pysyvät samankaltaisina. Splunkilla pystyy myös valvomaan mahdollisia konfiguraatiovirheitä ja brute-force-kirjautumishyökkäyksiä. Tietyltä ajalta tehdyt konfiguraatiomuutokset saadaan näkymään määrittelemällä haluttu aikaväli ja tapahtumalokin tyyppi.

Seuraavassa esimerkissä ajetaan komento action=login AND status=failed indeksiin, jossa sijaitsee kymmeniä saman asiakkaan verkkolaitteita ympäri maailmaa. Hakutulokset etsitään seitsemän päivän ajalta, jolloin saadaan kuvan 17 tulos laitekohtaisten tapahtumien määrästä.

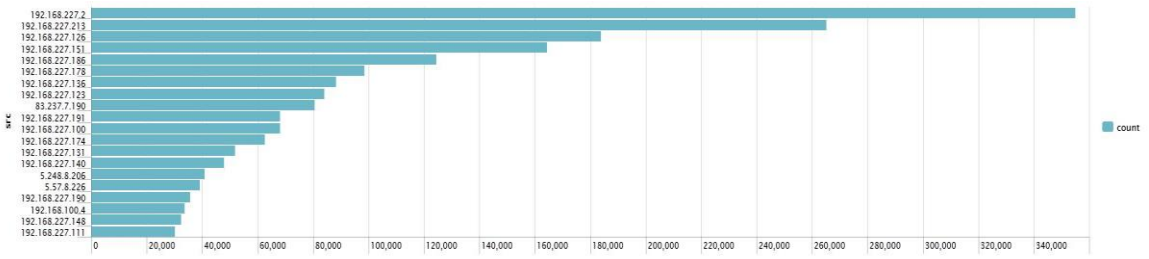
Kuva 17. Epäonnistuneet kirjautumiset indeksissä oleviin verkkolaitteisiin.

Jo viikon hakutulosten perusteella muutamalla laitteella näkyy normaalista poikkeavia arvoja. Tarkemman tutkimisen jälkeen kirjautumisyritykset ovat tulleet internetistä laitteen SSH-porttiin, suosituimpana lähdemaana Thaimaa sekä Ukraina. Suosituimmat kirjautumiseen käytetyt käyttäjänimet ovat root, oracle, webmaster ja nagios. Todennusvirheviestit häviävät kun SSH-kirjautuminen estettiin internetistä.

Tapahtumien määrän ja käyttökaistan tarkkailu

Solmukohdissa sijaitsevat verkkolaitteet mahdollistavat helposti läpi kulkevan liikenteen analyysin. Oleellinen raportti on eniten kaistaa käyttävien osoitteiden listaus. Esimerkissä tutkittiin Venäjällä Pietarissa sijaitsevan palomuuriklusterin läpi kulkevaa liikennettä. Kahden viikon raportin perusteella löytyi muutama selvästi muita korkeammalla oleva lähde, jotka paljastuivat paikallisiksi sähköpostipalvelimiksi ja domain controlleriksi.

Kerätylle tiedolle tehtiin suodatin tunnettujen palvelimien osalta, jolloin jäljelle jäi vain uniikit IP-osoitteet, joiden jakauma nähdään kuvassa 18. Tämän raportin perusteella voidaan kuvata normaalitilanne, joihin mahdollisia poikkeamia voidaan verrata. Keskimääräisestä päivittäisestä tapahtumamäärästä erottuva poikkeama on täten mahdollista huomata. Raporttiin voidaan lisätä vielä esimerkiksi vaihtoehto, jolla listataan liikennetapahtumat, joissa on lähdeporttina alle 1024. Nämä portit vaativat ylläpitäjän oikeudet, eikä tälläistä liikennettä pitäisi olla käyttäjien normaaliliikenteessä. Raportti paljastaa mahdolliset konfiguraatiovirheet ja saastuneet koneet. Myös käytetyimpien porttien perusteella voidaan tehdä tehokasta analyysia kaistankäytön optimoimiseksi.



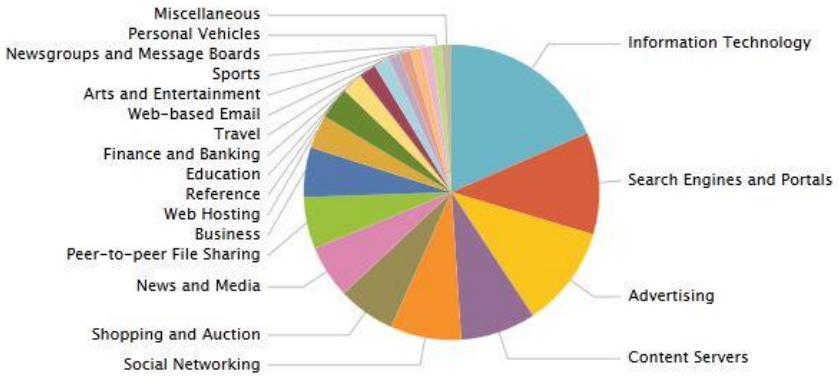
Kuva 18. Tietoverkon osoitteiden kaistan jakautuminen.

Splunkissa on myös mahdollisuus liittää kaikkeen IP-tietoon GeoIP-tiedot. Tämä mahdollistaa yhteyksien lähde- ja kohdesijainnin paikallistamisen, kunhan IP-osoite on julkinen. Yhtiöllä on yleensä tietty maantieteellinen alue, johon ydinbisnes sijoittuu. Alueen ulkopuolisten yhteyksien nopeutta voidaan esimerkiksi rajoittaa.

Tarkan statistiikan keräys

Vaikka kaistankäyttö ja tapahtumien määrä on mielenkiintoinen perusmittari poikkeamien havannoimiseen, on tilanteita, joissa tietyn tarkan mittarin arvioimiselle on tarve. Testiksi ajettiin eräälle suurelle suomalaiselle yhtiölle kuuluvan pääpalomuurin kahden viikon tietomäärälle haku, josta on suodatettu pois normaalit liikennelokin tapahtumat. Tuloksena oli silti kymmeniä miljoonia tapahtumia koskien tietoverkon kulkevaa liikennettä.

Suoraan tietoverkon tietoturvaan liittyvää tietoa olivat käytetyimmät verkkopohjaiset sovellukset, URL-pohjainen osoitteiden luokittelu sekä viruksiin liittyvät ja hyökkäyksen estoon liittyvät tapahtumat. Sovellukset ja web-selailun tyypin kategoriaan liittyvät tapahtumat voidaan jakaa aliverkoittain. Tätä tietoa voidaan hyödyntää yhtiön omien tietoturvapolitiikkojen kehittämiseen. Virusten- ja hyökkäykseneston aktivoineista sormenjäljistä, tyypeistä, osoitteista ja tiedostonimistä oli myös saatavilla listat. Suurimpana antivirussuodatuksen aktivoijana olivat epäilyttävistä osoitteista tulleet sähköpostit. Esimerkkipalomuurissa ei ollut käytössä käyttäjien todennus, mutta kyseisen palvelun ollessa käytössä tapahtumat voidaan tunnistaa käyttäjätasolla. Kuvassa 19 nähdään web-kategorioiden jakauma.



Kuva 19. Web-liikenteen jakautuminen.

Tapahtumalokin seuranta

Splunkin keskitetty rakenne sopii hyvin laitteiden tilasta kertovien tapahtumien tutkimiseen. Yhdellä haulla pystyy etsimään useamman laitteen tilaa samaan aikaan. Esimerkiksi seuraava yksinkertainen komento listaa kaikkien indeksissä sijaitsevien laitteiden prosessorin ja muistin käytön ja tämän hetkiset sessiot. Tulokset listataan prosessorin käytön mukaisessa järjestyksessä.

```
index=indeksin_nimi subtype=perf-historical | table host, cpu, mem, total_session | dedup host | sort 5 cpu
```

Windows server on yleisesti hyvin suosittu palvelinympäristö, mutta yksi sen suuria ongelmia on lokien tarkastelu ja halutun informaation löytäminen. Yksinkertainen tehtävä, kuten RADIUS-toiminnon todennusten käyttäjäkohtainen etsiminen on vaivalloista. Windows-palvelimen lokituksen vienti Splunk-indekserille mahdollistaa edellä mainitun tehtävän määrittelemällä hakuun ainoastaan palvelimen nimen sekä RADIUS-todennukseen käytetyn käyttäjätunnuksen. Oletuksena Windows server kuitenkin kirjoittaa muistiin massiivisen määrän monelle käyttäjälle turhaa tietoa, joten tallennettavan tiedon määrää on syytä rajoittaa. Tämän pystyi suorittamaan joko universal forwarderin asennuksessa, tai suoraan inputs.conf-tiedostoa palvelimella muokkaamalla.

Ongelmanselvitys

Ongelmatilanteissa on toisinaan vaikea hahmottaa vian alkulähdettä. Tämä korostuu tilanteissa, joissa useampi verkkokomponentti keskustelee keskenään. Esimerkkinä autentikaatiopalvelimella sijaitseva tietokantakomponentti voi jumiutua, eikä palvelimeen liittyvään palveluun pysty enää kirjautumaan. Splunkilla voi etsiä virhe-tyypin tapahtumia, joiden perusteella pystytään paikallistamaan virheilevä laite. Tarkemmin laitteen lokia tutkittaessa huomataan syylliseksi osoittautuva komponentti.

Työn aikana tapahtuva ongelmanselvitys sijoittui kuitenkin pääasiassa yksittäiseltä laitteelta löytyvän tiedon hyödyntämiseen. Eräässä ongelmatapauksessa asiakas on ilmoittanut, että Kiinassa sijaitsevan työnhallintapalvelimella suoritettavat työt katkeavat tasaisin väliajoin. Katkeamisista oli tiedossa aikavälit, sekä käytetyn palvelimen kohdeosoite. Lokista tehdyn raportin perusteella päädyttiin siihen, että TCP-session aikana on riittävästi tyhjää aikaa, jolloin sessio lopetetaan ja prosessi jää palvelimella jumiin. Palvelimella sijaitsevan ohjelman toimintaa ei ollut tämän osalta mahdollista muuttaa, joten ongelma kierrettiin lisäämällä TCP-sessioaikoja yhteyteen liittyvien palomuurisääntöjen osalta.

Toisessa esimerkkitapauksessa asiakkaan sivukonttorissa sijaitsevassa tietoverkossa havaittiin selkeää hitautta. Vasteajat internetin suuntaan vaihtelivat suuresti ja ajoittain näkyi pakettihäviötä. Asiakkaan yhteyshenkilö oli ottanut paikalliseen internet-palveluntarjoajaan yhteyttä, josta todettiin linjan olevan kunnossa, mutta todella kovin kuormitettu. Konttorista kulkivat sisäiset yhteydet salattuna VPN-tunnelia pitkin Suomessa sijaitsevaan pääkonttoriin, joten ulkopuoliset eivät liikennettä pysty tarkemmin arvioimaan. Sivukonttorissa sijaitsevaan palomuuriin ajettiin haku vuorokauden ajalta, minkä myötä löytyi eräs laite, joka käytti kaistaa kymmenkertaisesti muihin nähden. Selvisi, että kyseisellä laitteella oli jumiutunut ohjelma, joka tuotti turhaa liikennettä Suomessa sijaitsevaan palvelimeen. Samalla verkosta löytyi huomattava määrä BitTorrent-liikennettä internetiin, mikä estettiin palomuurilla.

Yleisesti Splunk-lokienhallintajärjestelmän käyttö ongelmatilanteissa toi etuja ajankäytön suhteen. Samat tiedot on ajoittain mahdollista löytää lokaalisti laitteilta, mutta keskitetyssä ratkaisussa ne ovat nopeammin saatavilla. Kokonaiskuvassa säästetty aika kertyy ja tuotettavuus parantuu. Ongelmanselvityksessä suuri hyöty tulee tiedon säilyttämisestä, sillä perussyyanalyysin määritteleminen jälkeenpäin on mahdotonta, jos tarvittavaa tietoa ei ole enää tallessa. Keskitetty tallennuskapasiteetti mahdollistaa myös sen, että lokit ovat näkyvillä, vaikka itse laite olisikin saavuttamattomissa. Splunkin indeksoinnilla saavutetaan lähteestä riippuen suuri säästö levytilassa, sillä pakattu tieto on noin 10 % alkuperäisestä.

5 Päätelmät

Verkon aktiivilaitteiden määrän lisääntyessä keskitetyt lokienhallintajärjestelmät ovat houkutteleva vaihtoehto. Skaalautuvat vaihtoehdot antavat mahdollisuuden tutustua tekniikkaan hiljalleen. Itselläni oli aiempaa kokemusta lähinnä suljetuista lokijärjestelmistä, jotka tukevat ainoastaan tiettyjä laitteita ja sisältävät tietyt tehtaalla määritetyt toiminnot ja hakuehdot. Tähän verrattuna Splunkin monimuotoisuus tuli esiin positiivisessa mielessä.

Insinöörityössä oli tavoitteena selvittää, mitä lisäarvoa kyseinen järjestelmä tuo perinteisempään, kyselyihin perustuvaan monitorointiin ja manuaaliseen lokin käsittelyyn. Työn edetessä kävi ilmi, että jo pienillä lokin perusteella luoduilla raporteilla ja analyyseillä voi tehostaa huomattavasti tietoverkon tietoturvaa ja vianselvitystä. Etu syntyi siitä, että saatavilla oli kaikki tieto tapahtumasta vain rajatun minimitiedon sijaan. Poikkeaman löytyessä sen aiheuttajaa oli mahdollista tutkia ja tilanteeseen pystyttiin reagoimaan. Keskitetyn järjestelmän hyödyt näkyivät usean lähteen samanaikaisesta tutkimisesta, joiden perusteella pystyttiin luomaan kattavia raportteja useasta eri lähteestä. Yhtenäisen käyttöliittymän käytön useampaa eri ympäristöä tutkiessa voi laskea myös suureksi eduksi. Järjestelmän hyödyntämisen opetteluun kuluu aikaa, jolloin asiantuntemus on etu palveluntarjoajan kannalta, sillä asiakkaan ei ole kannattavaa itse kouluttaa henkilöstöä tätä varten.

Työn empiirisessä osiossa keskityttiin lähinnä verkkolaitteisiin ja niiden lokien mahdollistamiin seikkoihin. Jotta tietoverkosta saisi täydellisen kokonaiskuvan, tulisi verkossa sijaitsevista palvelimista ja sovelluksista myös lähettää tietoa Splunkiin. Kirjoitetulle työlle tämä olisikin mielenkiintoinen ja luonnollinen laajennus.

Insinöörityö kokonaisuutena oli hyvin opettavainen. Käytetty ohjelmisto oli entuudestaan tuntematon, eikä suurien tietomäärien analysoinnista ollut aikaisempaa kokemusta. Lähdemateriaali hankittiin pääasiassa verkkolähteistä, joista löytyi paljon tietoa aiheesta. Kokonaisuuden läpikäyminen auttoi ymmärtämään, mihin lokienhallinnassa kannattaa kiinnittää huomiota ja kuinka tärkeää lokien tallentaminen on. Splunkin opettelussa auttoi myös huomattavasti se, että ohjelman saa ladattua ilmaislisenssillä, jolloin sen toimintaa oli mahdollista harjoitella myös kotiympäristössä.

Lähteet

1 DiNicolo, Dan. 2007. CCDA Study guide. Verkkodokumentti. <http://www.2000trainers.com/ccda-study-guide/iso-network-management>. Luettu 11.1.2013.

2 Hautaniemi, Mika. 1994. TKK/Atk-keskuksen TCP/IP-verkon valvonta ja hallinta. Diplomityö. <http://www.netlab.tkk.fi/julkaisut/tyot/diplomityot/611/diplomityo.book.html>. Luettu 11.1.2013.

3 Stallings, William. 2008. Business Data Communications, Chapter 20. <http://business.usi.edu/aforough/Chapter%2020.pdf>. Luettu 11.1.2013.

4 Reese, Brad. 2007. Netflow versus sFlow. Verkkodokumentti. Techworld. <http://features.techworld.com/networking/3865/netflow-versus-sflow/>. Luettu 20.1.2013.

5 Simple Network Management Protocol. 2012. Verkkodokumentti. Cisco Systems, Inc. <docwiki.cisco.com/wiki/Simple_Network_Management_Protocol>. Luettu 19.1.2013.

6 Rouse, Margaret. 2010. Remote Network Monitoring. Verkkodokumentti. SearchMobileComputing. <http://searchmobilecomputing.techtarget.com/definition/RMON>. Luettu 20.1.2013.

7 Introduction to Cisco IOS Netflow. 2012. Cisco Systems, Inc. <http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6555/ps6601/prod_white_paper0900aecd80406232.html>. Luettu 20.1.2013.

8 Jamil, Amir. 2009. The difference between SEM, SIM and SIEM. Verkkodokumentti. <http://amirjamil.blogspot.fi/2009/07/difference-between-sem-sim-and-siem.html>. Luettu 20.1.2013.

9 Lokiohje. 2009. Verkkodokumentti. Valtiovarainministeriö. <http://www.vm.fi/vm/fi/04_julkaisut_ja_asiakirjat/01_julkaisut/05_valtionhallinnon_tietoturvallisuus/20090511Lokioh/Vahti_3_NETTI.pdf>. Luettu 2.2.2013.

10 Guide to Computer Security Log Management. 2006. Verkkodokumentti. National institute of Standards and Technology. <http://csrc.nist.gov/publications/nistpubs/800-92/SP800-92.pdf>. Luettu 1.2.2013.

11 Chuvakin, Anton. 2011. The Complete Guide to Log and Event Management. Verkkodokumentti. Novell. <http://www.novell.com/docrep/2010/03/Log_Event_Mgmt_WP_DrAntonChuvakin_March2010_Single_en.pdf>. Luettu 3.2.2013.

12 Ranum, Marcus. 2004. System Logging and Log Analysis. Verkkodokumentti. <http://ranum.com/security/computer_security/archives/logging-notes.pdf>. Luettu 1.2.2013.

13 Carasso, David. 2012. Exploring Splunk. <http://www.splunk.com/web_assets/v5/book/Exploring_Splunk.pdf>. Luettu 19.1.2013.

14 Shenk, Jerry. 2011. Log management survey report. Raportti. <http://www.sans.org/reading_room/analysts_program/logmgt-survey-web.pdf>. Luettu 4.2.2013.

15 Selecting Intrusion Detection System. 2001. Verkkodokumentti. SANS Institute. <http://www.sans.org/reading_room/whitepapers/awareness/process-continuous-improvement-log-analysis_338>. Luettu 8.3.2013.

16 A Simple Network Management Protocol. 1990. IETF. <http://www.ietf.org/rfc/rfc1157.txt?number=1157>. Luettu 8.3.2013.

17 Remote Network Monitoring Management Information base. 2000. Verkkodokumentti. IETF. <http://tools.ietf.org/html/rfc2819>. Luettu 8.3.2013.

18 Cisco Systems NetFlow Services Export Version 9. 2004. Verkkodokumentti. IETF. <http://www.ietf.org/rfc/rfc3954.txt>. Luettu 8.3.2013.

19 Analyzing logs for SIEM. 2007. Verkkodokumentti. ZOHO corp. <http://www.manageengine.com/products/firewall/Analyzing-Logs-for-SIEM-Whitepaper.html>. Luettu 3.1.2013.

20 Company overview. 2013. Verkkodokumentti. Splunk. <http://www.splunk.com/web_assets/pdfs/secure/Splunk_Company_Overview.pdf>. Luettu 9.3.2013.

21 Splunk data pipeline. 2013. Verkkodokumentti. Splunk. <http://docs.splunk.com/Documentation/Splunk/latest/Deploy/Datapipeline>. Luettu 20.2.2013.

22 Components and roles. 2013. Verkkodokumentti. Splunk. <http://docs.splunk.com/Documentation/Splunk/latest/Deploy/Componentsofadistributedenvironment>. Luettu 22.2.2013.

23 How Splunk licensing works. 2013. Verkkodokumentti. Splunk. <http://docs.splunk.com/Documentation/Splunk/5.0.2/Admin/HowSplunklicensing works>. Luettu 22.2.2013.

24 Search manual. 2013. Verkkodokumentti. Splunk. <http://docs.splunk.com/Documentation/Splunk/latest/Search/Aboutsearch>. Luettu 24.2.2013.

25 Visualization reference. 2013. Verkkodokumentti. Splunk. <http://docs.splunk.com/Documentation/Splunk/latest/Viz/Visualizationreference> . Luettu 25.2.2013.

26 Alerting manual. 2013. Verkkodokumentti. Splunk. <http://docs.splunk.com/Documentation/Splunk/latest/Alert/Aboutalerts>. Luettu 25.2.2013.

27 The Syslog Protocol. 2009. Verkkodokumentti. IETF. <http://tools.ietf.org/html/rfc5424>. Luettu 22.2.2013.