

KESKITETTY LOKIENHALLINTA LABRANET-YMPÄRISTÖSSÄ

Jani Hallberg

Opinnäytetyö

Joulukuu 2013

Tietotekniikan koulutusohjelma

Tekniikan ja liikenteen ala





Tekijä(t) Hallberg, Jani	Julkaisun laji Opinnäytetyö	Päivämäärä 6.12.2013
	Sivumäärä 134 + 6	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty (X)
Työn nimi KESKITETTY LOKIENHALLINTA LABRANET-YMPÄRISTÖSSÄ		
Koulutusohjelma Tietotekniikan koulutusohjelma		
Työn ohjaaja(t) Piispanen, Juha		
Toimeksiantaja(t) Jyväskylän ammattikorkeakoulu Oy Jokinen, Juha		
Tiivistelmä <p>Opinnäytetyön toimeksiantajana toimi Jyväskylän ammattikorkeakoulu (JAMK). Työn toimeksiantona oli toteuttaa Jyväskylän ammattikorkeakoulun LabraNet -ympäristöön lokienhallintajärjestelmä. LabraNet on Jyväskylän ammattikorkeakoulun Dynamo kampuksella sijaitseva opiskeluverkko, joka toimii erillään ammattikorkeakoulun omasta verkosta. Sen tarkoitus on mahdollistaa ICT-koulutusohjelmien laboratorioympäristöjen helppo ja joustava toteutus sekä tarvittaessa nopeiden muutosten suoritus.</p> <p>Lokitieto on dokumentti tietynä ajan hetkenä toteutuneesta tapahtumasta. Lokitietoa tuottavat erilaiset laitteet ja järjestelmät, kuten verkkolaitteet, palvelimet, työasemat ja automaatiojärjestelmät. Lokitapahtumien käsittelyyn on kehitetty useita menetelmiä, esimerkiksi Syslog, SNMP ja Event Log. Lokitiedon keräämisen tavoitteet ovat organisaation liiketoiminnan jatkuvuuden varmistaminen, liiketoiminnan tehostaminen sekä tietoturvan hallinnan parantaminen.</p> <p>Lokijärjestelmän tärkein tavoite oli lokiviestien kerääminen LabraNet-ympäristön eri järjestelmistä ja viestien keskitetty arkistointi yhteen paikkaan. Viestejä tuli myös pystyä tarkastelemaan vaivattomasti sekä suodattamaan niiden joukosta tietyn tyyppiset tapahtumat. Järjestelmän tavoitteena oli täyttää kansallisen turvallisuusauditointikriteeristön (KATAKRI) perustason asettamat vaatimukset organisaation lokienhallintajärjestelmälle.</p> <p>Työn tuloksena LabraNet-ympäristön toiminnan kannalta kriittisten järjestelmien lokiviestit pystytään keräämään, analysoimaan ja arkistoimaan. Järjestelmän avulla viesteistä pystytään suodattamaan helposti haluttu tieto ja havaitsemaan mahdolliset tietomurrot tai vikatilanteet.</p>		
Avainsanat (asiasanat) AMQP, Event log, Graylog2, KATAKRI, Logstash, Loki, Syslog		
Muut tiedot		



Author(s) Hallberg, Jani	Type of publication Bachelor's Thesis	Date 6.12.2013
	Pages 134 + 6	Language Finnish
		Permission for web publication (X)
Title CENTRALIZED LOG MANAGEMENT SYSTEM IN LABRANET ENVIRONMENT		
Degree Programme Information Technology		
Tutor(s) Piispanen, Juha		
Assigned by JAMK University of Applied Sciences Jokinen, Juha		
Abstract <p>This bachelor's thesis was assigned by JAMK University of Applied Sciences. The goal of the thesis was to design and implement a log management system to the LabraNet study network. LabraNet is an independent study network for ICT department of School of Technology located at Dynamo campus. The network is a flexible environment, which enables different laboratory tasks for ICT degree programs.</p> <p>Log is a document of an event happening at a given time. Logs are produced by different devices and systems, such as network devices, servers, workstations and automation systems. Log management can be handled using many different applications, for example: Syslog, SNMP or Event log. Log management aims to ensure business continuity, enhance effectiveness of the business and improve information security management.</p> <p>The main goal for the log management system was to collect logs from different systems of the LabraNet environment and archive them into a centralized location. Administrators should be able to browse and filter events through easy to use interface. The system was designed to meet the requirements of the national security auditing criteria (KATAKRI) for organization log procedures.</p> <p>As result of the thesis, log messages can be collected, analyzed and archived from mission critical systems. The system enables administrators to easily filter desired information from log messages and to identify possible intrusions or system faults.</p>		
Keywords AMQP, Event log, Graylog2, KATAKRI, Logstash, Loki, Syslog		
Miscellaneous		

Sisältö

Lyhenteet	6
1 Lähtökohdat.....	8
1.1 Toimeksiantaja.....	8
1.2 Toimeksianto ja tavoitteet	8
2 LabraNet	9
3 Vaatimusmäärittely.....	10
3.1 KATAKRI.....	10
3.2 Lokiviestien lähteet ja kerättävät tapahtumat.....	12
3.2.1 Audit.....	12
3.2.2 Verkkolaitteet.....	12
3.2.3 Palvelimet	13
3.3 Tarkastelu ja analysointi.....	13
3.4 Säilytys ja poisto	14
3.5 Suojaus	14
3.6 Skaalautuvuus ja luotettavuus.....	15
3.7 Suorituskyky	16
4 Lokit	16
4.1 Lokin määritelmä	16
4.2 Lokin elinkaari	19
4.2.1 Elinkaaren osat	19
4.2.2 Lokipolitiikka	20
4.2.3 Konfiguraatio	23
4.2.4 Keräys	23
4.2.5 Normalisointi ja indeksointi	24
4.2.6 Korrelaatio ja baselining	24
4.2.7 Hälytykset ja raportointi	25
4.2.8 Säilytys ja poisto	25
4.2.9 Suojaus	26
4.3 Lokienhallinnan merkitys organisaation toiminnalle.....	27
4.4 Lokienhallinnan haasteet.....	29
5 Syslog.....	31
5.1 Yleistä.....	31
5.2 Protokollan rakenne	32
5.3 Viestiformaatti	33
5.3.1 PRI	33
5.3.2 HEADER	35
5.3.3 MSG.....	36
5.3.4 Esimerkit Syslog-viesteistä	36
5.4 Tietoturva	37
5.5 Syslog-toteutukset.....	38
5.5.1 Syslog-ng	38
5.5.2 Rsyslog.....	39
6 SNMP	40

6.1	Yleistä.....	40
6.2	Arkkitehtuuri.....	40
6.3	SMI ja MIB	41
6.4	Versiot	42
6.5	SNMP lokitapahtumien keräämiseen	42
7	Windows Event log.....	43
7.1	Tapahtumatyytit	44
7.2	Tapahtumien rakenne.....	44
7.3	Haut ja suodatus.....	46
8	Common Log Format ja Extended Log Format	47
9	Logstash	48
9.1	Yleistä.....	48
9.2	Rakenne.....	48
9.3	Web-käyttöliittymä	50
10	Elasticsearch.....	52
10.1	Yleistä.....	52
10.2	Rakenne.....	52
10.3	Datan tallennus ja haku	54
10.4	Klusterointi ja replikaatio	56
11	AMQP	56
11.1	Yleistä.....	56
11.2	Protokollan toiminta.....	57
11.3	Vaihteet	57
11.4	Jonot ja sidokset.....	60
11.5	AMQP-viestit	61
11.6	Yhteydet ja kanavat.....	61
11.7	RabbitMQ.....	62
12	Graylog2	63
12.1	Yleistä.....	63
12.2	Käyttöliittymä ja ominaisuudet.....	63
12.3	GELF	64
13	Käytännön toteutus	65
13.1	Lokienhallintajärjestelmän rakenne	65
13.2	Lokiviestien lähteet.....	66
13.3	Keräyspalvelimet	67
13.3.1	Kuvaus ja resurssit.....	67
13.3.2	Logstash-asennus	67
13.3.3	Logstash-asetukset	68
13.4	AMQP-vaihde.....	70
13.5	Kuvaus ja resurssit.....	70
13.6	Parsinta- ja tarkastelupalvelin	76
13.6.1	Kuvaus ja resurssit.....	76
13.6.2	Graylog2-asennus	77
13.6.3	Logstash-asetukset	83
13.7	Kytkimet	84
13.7.1	Yleistä.....	84

13.7.2	Juniper-kytkimet.....	85
13.7.3	Cisco-kytkimet.....	88
13.8	Palomuuuri.....	89
13.9	Linux-palvelimet.....	94
13.10	Nimipalvelimet.....	95
13.11	NTP-palvelin.....	96
13.12	RADIUS-palvelin.....	97
13.13	Sähköpostipalvelimet.....	99
13.14	Cisco UCS.....	99
13.15	Windows-palvelimet.....	100
13.16	LDAP-palvelin.....	106
13.17	Student-palvelin.....	107
13.18	Lokipalvelimet.....	108
13.19	Graylog2-verkkokäyttöliittymä.....	109
13.20	Graylog2 streams.....	115
13.21	Säilytyspalvelin.....	120
13.21.1	Kuvaus ja resurssit.....	120
13.21.2	Logstash asetukset.....	120
14	Työn tulosten arviointi.....	122
14.1	Lokitallenteiden kattavuus.....	122
14.2	Lokiviestien tarkastelu ja haku.....	124
14.3	Viestien arkistointi ja poisto.....	125
14.4	Tietoturva.....	126
14.5	Skaalautuvuus ja luotettavuus.....	126
14.6	Suorituskyky.....	127
15	Yhteenveto.....	133
15.1	Työn toteutus ja tulokset.....	133
15.2	Tulevaisuuden kehityskohteet.....	134
	Lähteet.....	135
	Liitteet.....	138
	Liite 1. Ohje uuden lokilähteen lisäykseen.....	138

Kuviot

Kuvio 1.	Syslog-tasot ja funktiot.....	32
Kuvio 2.	Syslog-viesti esimerkki 1.....	36
Kuvio 4.	SNMP hallinta-aseman ja agentin suhde.....	41
Kuvio 5.	Event Viewer pääikkuna.....	43
Kuvio 6.	Event Viewer XML esimerkki.....	45
Kuvio 7.	Weventutil haku.....	47
Kuvio 8.	ELF-lokiviesti.....	48
Kuvio 9.	Kibanan käyttöliittymä.....	51
Kuvio 10.	JSON esimerkki.....	53
Kuvio 11.	Elasticsearch GET esimerkki.....	54
Kuvio 12.	Elasticsearch POST esimerkki.....	55
Kuvio 13.	Elasticsearch haku.....	55
Kuvio 14.	Direct-exchange.....	58
Kuvio 15.	Fanout-exchange.....	58

Kuvio 16. Topic-exchange.....	59
Kuvio 17. Headers-exchange.....	60
Kuvio 18. GELF-viesti	64
Kuvio 19. Lokijärjestelmän toiminta.....	65
Kuvio 20. Logstash konfiguraatioesimerkki	69
Kuvio 21. RabbitMQ hallintapaneeli	72
Kuvio 22. Vaihteen luonti	73
Kuvio 23. Labranet-loki liitokset	74
Kuvio 24. Jonon viestimäärät	74
Kuvio 25. Jonon viestien lähteet ja asiakkaat.....	75
Kuvio 26. AMQP-palvelimen yhteydet.....	75
Kuvio 27. AMQP-palvelimen käyttäjienhallinta	76
Kuvio 28. Graylog2 pääsivu	82
Kuvio 29. Paloalto Syslog-asetukset.....	90
Kuvio 30. Cisco UCS Syslog-asetukset	100
Kuvio 31. Nxlog-palvelun käynnistys.....	103
Kuvio 32. Event log -viesti Graylog2-verkkokäyttöliittymässä	105
Kuvio 33. Graylog2 messages-sivu.....	109
Kuvio 34. Graylog2 viestin tiedot.....	110
Kuvio 35. Graylog2 hakuesimerkki 1	111
Kuvio 36. Graylog2 hakuesimerkki 2.....	112
Kuvio 37. Graylog2 quickfilter-esimerkki 1	113
Kuvio 38. Graylog2 quickfilter-esimerkki 2	114
Kuvio 39. Graylog2 hosts	114
Kuvio 40. Graylog2 stream-kategorian luonti	116
Kuvio 41. Graylog2 viestivirran säännön asetus	116
Kuvio 42. Graylog2 viestivirran kategorian asetus	117
Kuvio 43. Graylog2 viestivirran viestien tarkastelu	117
Kuvio 44. Graylog2 viestivirran graafi.....	118
Kuvio 45. Graylog2 viestivirran asetukset	118
Kuvio 46. Graylog2 viestivirran hälytykset.....	119
Kuvio 47. Graylog2 viestivirran ulostulot	119
Kuvio 48. Graafi päivän viestimäärästä.....	124
Kuvio 49. Yleisimpien domain nimien haku.....	125
Kuvio 50. AMQP-vaihteen viestien lisäys jonoon	127
Kuvio 51. Keräyspalvelimien prosessorikuorma.....	127
Kuvio 52. Keräyspalvelimien muistin kulutus	128
Kuvio 53. AMQP-vaihteen resurssien kulutus	129
Kuvio 54. Parsinta- ja tarkastelupalvelimen prosessorien kuorma	130
Kuvio 55. Parsinta- ja tarkastelupalvelimen muistin kulutus.....	130
Kuvio 56. Parsinta- ja tarkastelupalvelin tallennustilan kulutus	131
Kuvio 57. Säilytyspalvelimen prosessorikuorma ja muistin kulutus	132
Kuvio 58. Säilytyspalvelin tallennustilan kulutus	132
Kuvio 59. Grok Debugger esimerkki.....	139

Taulukot

Taulukko 1. Syslog facility arvot.....	34
Taulukko 2. Syslog severity arvot.....	34
Taulukko 3. Elasticsearch perustyytit.....	53
Taulukko 4. Lokipalvelinten tiedot	66

Taulukko 5. LabraNet-kytkimet	84
Taulukko 6. Spidernet-kytkimet.....	85

Lyhenteet

AD	Active Directory
AMQP	Advanced Message Queuing Protocol
BSD	Berkley Software Distribution
CIDEE	Cisco Intrusion Detection Event Exchange
CLF	Common Log Format
CSV	Comma-separated values
DHCP	Dynamic Host Configuration Protocol
ELF	Extended Log Format
FQDN	Fully Qualified Domain Name
GELF	Graylog Extended Log Format
IDS	Intrusion Detection System
JSON	JavaScript Object Notation
KATAKRI	Kansallinen Turvallisuusauditointikriteeristö
LDAP	Lightweight Directory Access Protocol
LVM	Logical Volume Manager
MAC	Media Access Control
MAC	Message Authentication Code

MIB	Management Information Base
NSM	Network Management Station
NTP	Network Time Protocol
OID	Object Identifier
RADIUS	Remote Authentication Dial In User Service
SDEE	Security Device Event Exchange
SIM	Security Information Management
SIEM	Security Information and Event Management
SLA	Service Level Agreement
SMI	Structure of Management Information
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SSL	Secure Sockets Layer
TLS	Transport Layer Security
UCS	Unified Computing System
VRF	Virtual Routing and Forwarding Instance
XML	Extensible Markup Language
XMPP	Extensible Messaging and Presence Protocol

1 Lähtökohdat

1.1 Toimeksiantaja

Työn toimeksiantajana toimi Jyväskylän ammattikorkeakoulu (JAMK). Jyväskylän ammattikorkeakoulu on kansainvälinen korkeakoulu, joka tarjoaa AMK-tutkintojen lisäksi opettajakoulutusta, avoimia ammattikorkeakoulututkintoja sekä täydennyskoulutus ja oppisopimustyyppistä koulutusta. Opiskelijoita koulussa on yli 8000 ja kampuksia neljä Jyväskylässä sekä yksi Saarijärvellä. JAMK ylläpitää vahvoja suhteita Jyväskylän ja Keski-Suomen alueen yrityksiin, joiden tarpeet vaikuttavat koulutuksen suuntaamiseen ja opetussuunnitelmien sisältöihin. (Tutustu JAMKiin 2013.)

JAMKin organisaatio koostuu neljästä koulutusta tuottavasta yksiköstä, joita ovat Ammatillinen opettajakorkeakoulu, Hyvinvointiyksikkö, Liiketoiminta ja palvelut – yksikkö sekä Teknologiayksikkö. Teknologiayksikkö jakautuu asiantuntijatiimeiksi viiteen tulosalueeseen: ICT, Konetekniikka, Logistiikka, Rakentaminen sekä Luonnonvarat. ICT-tulosalue järjestää informaatioteknologian insinöörikkoulutusta, joka koostuu automaatioteknologian, mediatekniikan ja tietotekniikan koulutusohjelmista. (Tutustu JAMKiin 2013.)

1.2 Toimeksianto ja tavoitteet

Työn toimeksiantona oli toteuttaa Jyväskylän ammattikorkeakoulun ICT-tulosalueen LabraNet-ympäristöön lokienhallintajärjestelmä. Järjestelmälle luotiin vaatimusmäärittely, jonka pohjana käytettiin kansallisen turvallisuusauditointikriteeristön (KATAKRI) perustason (IV) asettamia vaatimuksia organisaation lokienhallintajärjestelmälle.

Työn päätavoite oli lokitapahtumien keskitetty kerääminen ja arkistointi LabraNet-ympäristön eri laitteista ja järjestelmistä. Toinen tavoite oli lokiviestien helppo tarkastelu sekä hakujen teko tiettyjen arvojen perusteella, joka edellyttää lokiviestien parsimista ja normalisointia yhtenäiseen formaattiin. Tietyn tyyppisistä tapahtumista olisi myös hyvä tuottaa automaattiset hälytykset verkon ylläpitäjille. Järjestelmän itsessään tulisi olla helposti skaalautuva kasvavalle viestimäärälle sekä toiminnaltaan luotettava.

Järjestelmän toteutuksessa päädyttiin käyttämään useita sovelluksia eri tehtävien hoitoon. Pääasiallisena syynä tähän oli ”all in one” -sovelluksista puuttuva tuki kaikille halutuille ominaisuuksille, kuten AMQP (Advanced Message Queuing Protocol) -protokollalle. Lisäksi tällaiset sovellukset ovat lähes poikkeuksetta kaupallisia tuotteita, joiden lisenssikustannukset kohoavat usein suuriksi. Lokijärjestelmä suunniteltiin modulaariseksi, eli siihen on helppo myöhemmin lisätä uusia komponentteja kuorman tasaamiseksi tai tarpeen mukaan vaihtaa paremmin käyttö-tarkoitukseen soveltuviin.

2 LabraNet

LabraNet on Jyväskylän ammattikorkeakoulun teknologiayksikön ICT-tulosalueen opiskeluverkko, joka toimii erillään ammattikorkeakoulun omasta verkosta. LabraNet sijoittuu Dynamon kampuksen kolmanteen ja neljänteen kerrokseen palvelen noin 900 käyttäjää. Sen tarkoitus on mahdollistaa ICT-koulutusohjelmien laboratorioympäristöjen helppo ja joustava toteutus sekä tarvittaessa nopeiden muutosten teko.

Työasemien lisäksi LabraNet tarjoaa opiskelijoille monia opiskeluun liittyviä palveluja. Student-palvelin mahdollistaa opiskelijoiden verkkosivujen julkaisun sekä muita verkkojulkaisuihin liittyviä palveluja (Mysql, PHP). VMware ESXi-palvelinten avulla voidaan toteuttaa kokonaisia virtuaalisia ympäristöjä kurssveja ja projekteja varten. (Study Network for ICT 2009.)

SpiderNet-tietoverkkolaboratorio mahdollistaa verkko-operaattoritason tekniikoiden ja protokollien testaamisen useiden eri laitevalmistajien laitteilla. SpiderNet sisältää tällä hetkellä noin 50 laitetta muun muassa Cisco Systems, Juniper Networks ja Extreme Networks -laitevalmistajilta. (SpiderNet. 2009.)

LabraNet-verkon palvelut on nykyisin toteutettu lähes kokonaan virtuaalikoneina VMware ESXi -palvelimilla. Suurin osa palvelimista toimii HP blade -palvelimilla, joiden tallennustilasta vastaa HP EVA -tallennusjärjestelmä. LabraNet-verkko muodostuu Juniper- ja Cisco-kytkimistä. Verkon osat toisistaan eristävänä palomuurina toimii Palo Alto.

3 Vaatimusmäärittely

Vaatimusmäärittelyn tarkoitus on asettaa lähtökohdat toteutukselle. Siinä käydään läpi kerättävät kohteet, halutut tapahtumatyypit ja lokijärjestelmältä vaadittavat ominaisuudet. Vaatimusmäärittely laadittiin yhdessä LabraNet-ympäristön ylläpitäjien kanssa.

3.1 KATAKRI

KATAKRI (Kansallinen turvallisuusauditointikriteeristö) on kriteeristö, jota käytetään viranomaisen suorittaessa yrityksessä tai muussa yhteisössä kohteen turvallisuustason todentavan tarkastuksen eli auditoinnin. Turvallisuusauditointikriteeristön toinen päätavoite on auttaa yrityksiä ja muita yhteisöjä sekä viranomaisia sidosryhmineen omassa sisäisessä turvallisuustyössään. Kriteeristön versio II julkaistiin vuonna 2011. (Kansallinen turvallisuusauditointikriteeristö (KATAKRI) 2013.)

KATAKRI sisältää neljä osa-aluetta: hallinnollinen turvallisuus, henkilöstöturvallisuus, fyysinen turvallisuus ja tietoturvallisuus. Vaatimukset on luokiteltu kolmeen turvallisuustasoon: perustaso, korotettu taso ja korkein taso. Lisäksi jokainen kohta sisältää viranomaisvaatimusten ulkopuoliset elinkeinoelämän suositukset.

Lokijärjestelmän turvallisuutta käsittelee kysymys I 504.0:

”Ovatko organisaation teknisten laitteiden ja palveluiden lokimenettelyt kunnossa?”.

Lisäkysymys: ”Kerätäänkö verkoista, laitteista ja järjestelmistä keskeiset lokitiedot ja käsitelläänkö niitä asianmukaisesti?”.

Viranomaisvaatimukset perustasolla (IV) ovat seuraavat:

- 1) Tallenteiden kattavuus on riittävä tietomurtojen tai niiden yritysten jälkikäteeseen todentamiseen.
- 2) Keskeisiä tallenteita säilytetään 6 kk tai erillisessä sopimuksessa määrätty aika.
- 3) Suojattavaa tietoa sisältävät lokitiedot on suojattu asianmukaisesti (pääsynvalvonta, käsittely, poisto).

(Kansallinen turvallisuusauditointikriteeristö 2011.)

Ensimmäinen viranomaisvaatimus koskee kerättävien lokiviestien kattavuutta, johon vaikuttaa pääasiassa laitteiden kyky tuottaa riittävästi informaatiota sisältäviä ja helposti tulkittavia lokiviestejä. Toisaalta liian suuri määrä viestejä hukuttaa helposti alleen tärkeät tapahtumat. Tietomurtojen havaitsemisessa on oleellista pystyä korreloimaan useista lähteistä vastaanotettuja viestejä yhteen, joka edellyttää viestien keräämistä riittävän monesta kohteesta ja ylläpitäjän tai lokijärjestelmän kykyä yhdistää toisiinsa liittyvät tapahtumat.

Tallenteiden riittävä kattavuus on pyritty varmistamaan kartoittamalla ympäristön kohteet, joista tapahtumia halutaan kerätä (verkkolaitteet, palvelimet ja työasemat). Tämän jälkeen selvitettiin millaisia tapahtumia ne tuottavat ja mitkä niistä ovat keräämisen arvoisia. Lokijärjestelmä ja laitteet konfiguroitiin näiden selvitysten perusteella.

Toinen vaatimus käsittelee lokiviestien säilytysaikaa. Lokiviestien säilytykseen vaikuttava tekijä on viestien kuluttama tallennustila suhteessa saatavilla olevaan tallennustilan määrään. Jos tallennustila ei riitä viestien säilyttämiseen 6 kuukauden ajan tulee tallennustilaa lisätä tai kerättävien viestien määrää vähentää. Viestien säilytysmuoto, kuten pakkaus, vaikuttaa suuresti niiden kuluttaman tilan määrään. Viestien säilytykseen on vaatimusmäärittelyssä omistettu oma luku.

Kolmas vaatimus koskee viestien suojausta. Suojauksessa tulee vaatimuksen mukaan ottaa huomioon pääsynhallinta sekä viestien turvallinen käsittely ja poisto. Suojaus käsitellään vaatimusmäärittelyn erillisessä luvussa.

3.2 Lokiviestien lähteet ja kerättävät tapahtumat

Tässä luvussa luetellaan LabraNet-ympäristön laitteet ja järjestelmät, joista lokiviestejä kerätään. Jokaisessa kohdassa on myös eritelty tallennettavat tapahtumatyyppit.

3.2.1 Audit

Kaikista lokiviestienlähteistä pyrittiin keräämään audit-lokit eli käyttäjien kirjautumiset ja suorittamat toiminnot. Audit-lokit ovat tietoturvan kannalta yksi tärkeimmistä tapahtumatyypeistä, koska niiden avulla voidaan muodostaa tapahtumaketju tietyn käyttäjän suorittamista toiminnoista eri järjestelmissä.

3.2.2 Verkkolaitteet

Lokitapahtumia haluttiin kerätä LabraNet-verkon toiminnan kannalta keskeisiltä verkkolaitteilta eli palomuurilta ja kytkimiltä. Palomuurina toimiva Palo Alto on tietoturvan kannalta keskeisessä asemassa, koska sen kautta kulkee liikenne internetin ja sisäverkon välillä. Palomuurilta tulisi kerätä seuraavan tyyppiset tapahtumat:

- Traffic. Sisältää sallitut ja estetyt yhteydet. Jokaisen palomuurisäännön kohdalle voidaan määrittää, kirjataanko lokiin yhteyden aloitus, lopetus vai molemmat.
- Threat. Sisältää palomuurin havaitsemat tietoturvauhat, joita voivat olla esimerkiksi havaitut virukset tai haittaohjelmat.
- System. Sisältää palomuurin toimintaan liittyvät tapahtumat, kuten ylläpitäjien kirjautumiset.
- Configuration. Sisältää järjestelmän konfiguraatioon tehdyt muutokset.

Kytkimiltä haluttiin kerätä lokiviestien kautta tietoa portteihin kytketyistä päätelaitteista sekä laitteisiin tehdyistä muutoksista. Kytkimiltä tulisi saada kerättyä seuraavat tapahtumat.

- MAC (Media Access Control) -port-kytkennät. Tapahtumatyyppejä sisältää tiedon kytkimen porttiin kytkeytyvän laitteen MAC-osoitteesta.
- DHCP (Dynamic Host Configuration Protocol) -bindings. Sisältää tiedon, mille MAC-osoitteelle on jaettu mikäkin IP-osoite DHCP-protokollan avulla.
- Konfiguraatiomuutokset. Kaikki laitteeseen tehdyt konfiguraatiomuutokset.
- Kaikki warning- ja emergency-vakavuustason tapahtumat.

3.2.3 Palvelimet

LabraNet-ympäristö sisältää useita palvelimia, joilta lokiviestejä halutaan kerätä. Verkon nimipalvelimilta tulisi kerätä seuraavat tapahtumat:

- Nimikyselyt (query). Sisältää päätelaitteiden tekemät nimikyselyt palvelimille.
- Zone transfer. Tapahtuma syntyy domain-kopion siirtyessä pääpalvelimelta orjapalvelimille.

LDAP (Lightweight Directory Access Protocol)-, Windows Active Directory - ja RADIUS (Remote Authentication Dial In User Service) -palvelimilta kerätään käyttäjien onnistuneet ja epäonnistuneet kirjautumiset. Verkon aikapalvelimelta halutaan tieto ajan synkronoinnin onnistumisista ja epäonnistumisista. VMware ESXi – ja Cisco UCS (Unified Computing System) -palvelimilta tulisi kerätä ylläpitäjien kirjautumiset.

3.3 Tarkastelu ja analysointi

Yksi lokijärjestelmän päävaatimuksista on mahdollistaa ylläpitäjille viestien helppo tarkastelu ja analysointi. Kaikkia viestejä tulisi pystyä tarkastelemaan yhden käyttöliittymän kautta ja tekemään hakuja niihin keskitetysti. Viestejä tulisi pystyä suodattamaan kenttien, esimerkiksi lähde IP-osoitteen tai käyttäjän, perusteella. Lisäksi viestien määrästä olisi hyvä saada esitettyä kaavioita.

Viestien tarkastelu ja analysointia varten työhön valittiin Graylog2-sovellus, joka on ilmainen ja perustuu avoimeen lähdekoodiin. Se sisältää vaadittavat ominaisuudet eli viestien helpon tarkastelun, haun ja suodatuksen. Lisäksi se mahdollistaa kaavioiden tuottamisen ja viestien jakamisen kenttien arvojen perusteella erillisiin osioihin. Graylog2 suurin heikkous on raportointiominaisuuksien puute, mikä ei kuitenkaan arvioitu olevan heikkous, joka estäisi sovelluksen käyttöönoton.

Toinen harkittu sovellus oli Splunk, joka on enterprise-tason kaupallinen datan keräämiseen, indeksointiin ja korrelaation erikoistunut ohjelma. Splunk sisältää monia Graylog2-sovellusta edistyneempiä ominaisuuksia, kuten raportoinnin, valvonnan ja automaattisen korrelaation (What is Splunk Enterprise? 2013.). Sitä ei kuitenkaan valittu työhön korkeiden lisenssikustannusten vuoksi (5000\$ + tukikustannukset) (Pricing 2013).

3.4 Säilytys ja poisto

Lokiviestien säilytys on jaettu kahteen osaan: lyhytaikaiseen ja pitkäaikaiseen säilytykseen. Graylog2 varastoi valmiiksi parsitut lokiviestit elasticsearch-tietokantaan, joka on optimaalinen nopeiden hakujen suoritusta varten. Tässä tietokannassa viestit säilytetään vain kuukauden ajan. Muokkaamattomat raakalokiviestit ohjataan erilliselle säilytyspalvelimelle pitkäaikaissäilytykseen, jossa ne säilytetään pakatussa muodossa. Pitkäaikaissäilytyksessä data pidetään vähintään KATAKRI-kriteeristön vaatiman kuusi kuukautta. Saatavilla olevan tallennustilan määrästä riippuen säilömisäikää voidaan pidentää.

Lokiviestien poistosta elasticsearch-tietokannasta huolehtii automaattisesti Graylog2-sovellus. Säilytyspalvelimen viestien poisto hoidetaan tehtävään erikoistuneella Logrotate-sovelluksella.

3.5 Suojaus

Lokijärjestelmän suojaus pyritään toteuttamaan järjestelmän jokaisessa osassa. Käytännössä tämä tarkoittaa lokiviestien lähteitä sekä keräys-, tarkastelu- ja säilytyspalvelimia. Suojaus suoritetaan asettamalla järjestelmiin pääsynhallinta, joka estää ulkopuolisten henkilöiden kirjautumisen. Pääsynhallinta pyritään liittämään keskitettyyn LDAP-käyttäjätietokantaan.

Lokiviestit liikkuvat erillisessä hallintaverkossa, joka on eristetty normaalista verkkoliikenteestä. Tällä pyritään estämään viestien kaappaaminen verkkoliikennettä kuuntelemalla. Tehokkaampien suojauskeinojen, kuten salauksen käyttöönottoa, ei koettu tarpeelliseksi, koska kerättävät lokiviestit eivät sisällä esimerkiksi käyttäjien henkilötietoja.

3.6 Skaalautuvuus ja luotettavuus

Tapahtumien määrä ympäristössä voi ajan kuluessa kasvaa käyttäjämäärien lisääntyessä ja uusien lokilähteiden lisäyksen myötä. Lokijärjestelmän tulee pystyä skaalautumaan lisääntyvän viestimäärän mukana. Järjestelmän toteutus virtuaalisena mahdollistaa resurssien helpon lisäämisen tarvitseville komponenteille. Useat työhön valituista sovelluksista tukevat kuorman jakamista klusteroinnin avulla ja pystyvät skaalautumaan uusia palvelimia klusteriin lisäämällä.

Lokijärjestelmän toiminnan luotettavuus on tärkeä ottaa huomioon rakenteen suunnittelussa ja käytettäviä tekniikoita valittaessa. Järjestelmässä viestien siirtoon palvelinten välillä käytetään AMQP-protokollaa, joka on suunniteltu luotettavaan viestien välitykseen. Kaikki viestit välitetään vaihteen kautta, joka pystyy tallentamaan viestejä puskuriin vastapään ollessa kykenemätön vastaanottamaan niitä. Tällöin esimerkiksi säilytyspalvelimen kaatuminen ei aiheuta välitöntä viestihävikkiä. Protokolla tukee lisäksi viestien kuittauksia, joilla varmistetaan niiden perillemeno.

3.7 Suorituskyky

Järjestelmän suorituskyvyn seuranta on tärkeää varsinkin alkuvaiheessa, jotta laiteresurssit pystytään mitoittamaan oikein. Esimerkiksi viestien kuluttama tallennustilan määrä on vaikea arvioida etukäteen, jolloin uhkana on tilan loppumien pidemmällä aikavälillä. Palvelinten resurssien kulutusta seuraamalla voidaan suorittaa ns. baseline-mittaus. Baseline-mittauksessa eri laiteresurssien, kuten muistin ja tallennustilan, kulutus mitataan tietyltä aikaväliltä. Tulosten perusteella voidaan arvioida resurssien riittävyys sekä havaita pullonkaulat järjestelmässä. Baseline-mittaus suoritettiin keräämällä tiedot palvelimilta SNMP (Simple Network Management Protocol) -protokollalla ja tuottamalla niistä kuviot Cacti-sovelluksella.

4 Lokit

4.1 Lokin määritelmä

”Lokitiedoksi kutsutaan dokumenttia jonkin tapahtuman toteutumisesta jonakin tietynä ajan hetkenä. Loki dokumentoi tapahtumia, jotka ovat tapahtuneet organisaation järjestelmissä, verkoissa tai muussa ympäristössä ja toiminnassa.” (Lokiohje 2009, 13.)

Lokitietoja voidaan kerätä erilaisista laitteista ja järjestelmistä, esimerkiksi verkkolaitteista, palvelimista, työasemista tai automaatiojärjestelmistä. Lokien keräystapoja on olemassa useita. Yleisimpinä mainittakoon SNMP- ja Syslog-protokollat sekä Windows-ympäristöissä käytössä oleva Event log.

Lokit voivat sisältää monenlaista tietoa tapahtumasta viestin tuottavasta sovelluksesta riippuen. Lokiviesti sisältää käytännössä aina aikaleiman, lähteen ja datan. Aikaleima kertoo milloin viesti on luotu ja on erityisen tärkeä aukottoman tapahtumaketjun todentamiseksi (Lokiohje 2009, 13). Lähde ilmaisee viestin tuottaneen järjestelmän ja on yleensä joko IP-osoite tai hostname. Data on lokiviestin ydin, joka sisältää tiedot tapahtumasta. Data voi sisältää esimerkiksi viestin tuottaman ohjelman nimen, käyttäjänimen, virhekoodin, tai suoritettua toimenpiteen. Aikaleiman, lähteen ja datan esitysmuodosta ei ole olemassa standardia formaattia, minkä vuoksi eri järjestelmät tuottavat erimuotoisia lokiviestejä. (Chuvakin, Schmidt & Crishtopher 2012, 6.)

Lokiformaatteja on olemassa useita, joista seuraavassa luettelossa on esitetty muutama yleisin:

- Syslog (RFC 3195, RFC 5424)
- Windows Event log
- W3C Common Log Format (CLF) ja Extended Log Format (ELF)
- Cisco SDEE/CIDEE

Edellä mainituista protokollista Syslogia käytetään lokiviestien tuottamisen lisäksi niiden verkon yli siirtämiseen. (Chuvakin, Schmidt & Crishtopher 2012, 36.)

Lokiviestien formaatit voidaan edelleen jakaa tyyppin mukaan avoimiin ja suljettuihin sekä teksti- ja binäärityyppisiin. Avoimista lokiformaateista on julkisesti saatavilla dokumentaatio joko standardina tai suosituksena. Esimerkki avoimesta formaatista on Syslog, joka on dokumentoitu IETF:n RFC-suosituksissa. Suljetut formaatit ovat usein heikosti dokumentoituja, jos ollenkaan, ja ovat yleensä käytössä vain tietyn valmistajan laitteissa. Suljettujen lokiformaattien lukuun ja käsittelyyn joudutaan usein käyttämään valmistajan omia sovelluksia, eikä voida olla varmoja, onko lukusovellusta saatavilla enää viiden tai kymmenen vuoden kuluttua. Esimerkki suljetusta lokiformaatista on Windows Event log. (Chuvakin, Schmidt & Crishtopher 2012, 40.)

Useimmat lokiformaatit ovat ASCII-tekstityypisiä. Tämän tyyppiset tiedostot ovat helposti luettavissa millä tahansa tekstieditorilla, ja kaikki lokienkäsittelyyn tarkoitettut sovellukset pystyvät tulkitsemaan niitä. Tekstityyppi voidaan edelleen jakaa erilaisiin merkintäkieliin, kuten XML (Extensible Markup Language), CVS (Comma-separated values) tai JSON (JavaScript Object Notation). Osa lokia tuottavista sovelluksista tallentaa lokitiedon binääriformaattiin, jota pystytään lukemaan ja käsittelemään vain siihen tarkoitetuilla sovelluksilla. Binääriformaatin etu on nopeampi käsittely ja pienempi tilankulutus. Esimerkiksi Snort IDS käyttää unified-binääriformaattia tapahtumien tallentamiseen suorituskykyisistä. Binääriformaatin huono puoli on heikko pakkautuvuus tekstitiedostoihin verrattuna. Muita esimerkkejä ovat verkkoliikenteestä kaapatut pcap-tiedostot ja windows event log. (Chuvakin, Schmidt & Crishtopher 2012, 38, 78.)

Lokitietoa voidaan hyödyntää järjestelmän eheyden varmistamiseen, häiriöiden havaitsemiseen ja korjaamiseen sekä luotettavan tapahtumaketjun muodostamiseen. Lokien avulla voidaan jäljittää järjestelmän tapahtumia (kuka, mitä, milloin), virheitä, väärinkäyttö- ja tietomurtotilanteita ja niiden yrityksiä. Lokeja voidaan myös käyttää todistusaineistona rikosprosessissa. (Lokiohje 2009, 14.)

Lokitietoa hyödyntämällä pyritään selvittämään tapahtuman osapuolet, kiistämättömyys ja kulku. Osapuolilla tarkoitetaan tapahtumaan liittyviä toimijoita. Osapuolet voivat olla tunnistettuja toimijoita tai esimerkiksi verkkolaitteita, joiden käyttäjän identiteettiä ei voida tunnistaa. Kiistämättömyydellä pyritään varmistamaan, että tapahtuman osapuolet eivät voi kiistää osallisuuttaan tapahtumaan. Kiistämättömyyden varmistamisen edellytys on tapahtuman osapuolten identiteetin todentaminen. Tapahtumien kulku voidaan dokumentoida lokijälkien kronologisella keräämisellä, jolloin saadaan muodostettua tapahtumien sarja. Järjestelmien kellonajan oikeellisuus on tärkeää tapahtumien kronologisen järjestyksen kannalta. (Lokiohje 2009, 15.)

Lokitietojen avulla pyritään lisäksi

- havaitsemaan tunkeutumiset ja poikkeamat
- havaitsemaan järjestelmien suorituskykyongelmat
- varmistamaan käyttäjien oikeusturva.

Tunkeutumisen havaitsemisella pyritään tunnistamaan poikkeavat ja ei-sallitut tapahtumat. Tällaisia tapahtumia voivat olla epäonnistuneet kirjautumisyriytykset resurssien valtuuttamaton käyttö tai normaalista poikkeava liikenne verkossa. Suorituskyvyn valvonnalla voidaan varmistaa järjestelmien oikea toiminta ja käytettävyys. Suorituskykyyn liittyvät lokitapahtumat voivat sisältää tietoa resurssien käytöstä, kuormituksesta tai virhetilanteista. Käyttäjien oikeusturvan varmistaminen liittyy tapahtumien osapuolten, kiistämättömyyden ja kulun todentamiseen. Näiden avulla voidaan luotettavasti osoittaa, kuka on tai ei ole tehnyt jotain tiettyä asiaa tietojärjestelmässä tai muussa ympäristössä, josta lokia kerätään. (Lokiohje 2009, 15.)

4.2 Lokin elinkaari

4.2.1 Elinkaaren osat

Lokien hallintaprosessin suunniteltaessa on tärkeää ymmärtää lokin koko elinkaari ja eri vaiheissa tapahtuvat toimenpiteet. Elinkaaren ymmärtämien auttaa kokonaisuuden hahmottamisessa, ja vaiheisiin jako mahdollistaa monimutkaisen järjestelmän paloittelun helpommin hallittaviksi osiksi. Jaon perusteella voidaan arvioida eri tuotteiden ominaisuuksia ja soveltuvuutta prosessiin.

Elinkaaren vaiheet voidaan jakaa lokipolitiikkaan, konfiguraatioon, keräämiseen, normalisointiin, indeksointiin, korrelaatioon, baselineen, hälytyksiin, raportointiin, säilytykseen ja poistoon sekä suojaukseen. Käytännön toteutuksissa vaiheet on usein yhdistetty suuremmiksi kokonaisuuksiksi, esimerkiksi baseline, hälytykset ja raportointi lokien analysointia suorittavaan sovellukseen. Elinkaari säilyy kuitenkin samana. (Grimes 2010a.)

4.2.2 Lokipolitiikka

Lokipolitiikan määrittämisessä otetaan kantaa, minkä tyyppisiä tapahtumia halutaan kerätä, mistä kohteista, miten lokeja säilytetään ja kuinka kauan sekä ketkä lokeja voivat käsitellä ja miten. Politiikka toimii lähtökohtana koko hallintaprosessin suunnittelulle, ja sen määrittely on syytä tehdä huolella. Vaatimuksia sisällölle voivat asettaa organisaation tietoturvalaki, standardit, asetukset, lainsäädäntö ja viranomaisten asettamat vaatimukset. Lisäksi organisaation asiakkaiden kanssa solmitut palvelusopimukset (SLA – Service Level Agreement) voivat asettaa vaatimuksia, jotka lokipolitiikassa tulee ottaa huomioon.

Lokipolitiikka kattaa ihmiset, prosessit ja teknologian. Teknologia sisältää käytettävät työkalut. Prosessit määrittävät, miten lokitieto kerätään, analysoidaan ja säilytetään. Prosessien dokumentointi on tärkeää, jotta voidaan myöhemmin todistaa lokienhallinnan toiminta esimerkiksi oikeusistuimessa. Dokumentoitu prosessi myös helpottaa uusien henkilöiden perehdyttämistä järjestelmään. Prosessin dokumentaatio tulee myös muistaa pitää ajan tasalla. Ihmisiä tarvitaan prosessin suorittamiseen ja läpivientiin. Lokienhallintaprosessissa työskenteleville henkilöille tulee määrittää selkeät roolit ja vastuualueet, esimerkiksi ketkä päättävät organisaation lokipolitiikan sisällöstä, kelle kuuluu järjestelmän ylläpito ja ketkä ylittäävät pääsevät lokitietoihin käsiksi. (Chuvakin, Schmidt & Crishtopher 2012, 24.)

Ensimmäisenä tulee määrittää, millaisia tapahtumia järjestelmistä halutaan kerätä. Organisaation tietojärjestelmien koosta riippuen tapahtumia voi syntyä suuri määrä, joten oleelliset tapahtumat on syytä rajata. Tällöin tärkeiden tapahtumien havaitseminen helpottuu, eikä oleellinen tieto huku massan sekaan.

Seuraavassa luettelossa on esitetty yleisiä tapahtumatyyppisiä:

- Audit
- Järjestelmien tilamuutokset
- Suorituskyky
- Vikahälytykset
- Tilastotiedot

- Tietoturva

Audit tapahtumalla tarkoitetaan käyttäjien onnistuneita ja epäonnistuneita kirjautumisyriä tietojärjestelmään sekä käyttäjien järjestelmissä suorittamia komentoja. Järjestelmän tilamuutoksilla voidaan tarkoittaa esimerkiksi laitteen lämpötilan raja-arvon ylittymistä, joka johtaa lokitapahtumaan. Suorituskyky tapahtumalla tarkoitetaan järjestelmän lähettämiä viestejä suorituskyvyn heikentymisestä. Vikahälytyksellä tarkoitetaan esimerkiksi järjestelmän komponentin vikaantumisesta syntyvää tapahtumaa. Tilastotapahtumassa järjestelmä lähettää säännöllistä tilastotietoa esimerkiksi siirrettävän tiedon määrästä. Tietoturva tapahtuma voi syntyä esimerkiksi havaitusta tunkeutumisesta järjestelmään.

Lokien keräyskohteita määritettäessä on kartoitettava käytössä oleva laitekanta ja eri järjestelmien tehtävät. Kartoituksessa voidaan hyödyntää esimerkiksi laiteluetteloa. Kartoituksella pyritään selvittämään millaisia tapahtumia järjestelmät tuottavat ja mitkä niistä ovat haluttuja tapahtumatyyppejä.

Seuraavassa luettelossa on esitetty yleisiä lokien keräyskohteita:

- Unix- ja Windows-järjestelmät
- Reitittimet
- Kytkimet
- Langattomat tukiasemat
- VPN-järjestelmät
- IDS/IPS järjestelmät
- Tulostimet

Keräyskohteet voidaan jaotella edelleen käyttöjärjestelmiin ja yksittäisiin sovelluksiin. (Chuvakin, Schmidt & Crishtopher 2012, 8, 9.)

Lokiviestit sisältävät usein arkaluontoista tietoa organisaation tietoverkosta ja järjestelmistä. Loki voi myös joissain tapauksissa sisältää tietoa asiakkaista tai tunniste ja henkilötietoja. Lokipolitiikkaa suunniteltaessa on tärkeä tunnistaa kerättävän lokitiedon sisältö ja sen pohjalta määrittää politiikkaan säännöt lokien käsittelylle. Säännöissä tulisi määrittää keillä tietoihin on oikeus päästä käsiksi ja miten arkaluontoisiksi luokiteltu tieto suojataan sekä käsitellään. Tiedon suojaus tulee ottaa huomioon teknisiä ratkaisuja suunniteltaessa, esimerkiksi pääsynhallinta- ja salauksenratkaisujen muodossa. Arkaluontoista tietoa käsiteltäessä tulee varmistaa että käsittely-ympäristö on tietoturvallinen, esimerkiksi kokonaan eristetty muista järjestelmistä.

Lainsäädäntö asettaa useita vaatimuksia lokien käsittelylle. Seuraavassa luettelossa on esitetty merkittävimmät lait, jotka asettavat vaatimuksia ja rajoitteita lokien käsittelylle:

- Henkilötietolaki
- Julkisuuslaki
- Laki yksityisyyden suojasta työelämässä
- Sähköisen viestinnän tietosuojalaki

Lainsäädäntö asettaa rajoituksia lokitiedon käsittelyyn erityisesti silloin, kun ne sisältävät tunniste- tai henkilötietoja. Lokitiedoista muodostuu henkilörekisteri, jos ne sisältävät henkilöä koskevaa tunnistettavaa tietoa. Tällöin tulee huomioida kaikki henkilörekisteriä koskevat vaatimukset tai arvioida uudelleen henkilötietojen tallentamisen tarve. (Lokiohje 2009, 20 - 21.)

Lainsäädännön lisäksi lokipolitiikassa voidaan joutua ottamaan huomioon standardien asettamat vaatimukset. Tietoturvastandardeista esimerkiksi ISO27001 ja ISO27002 sisältävät vaatimuksia lokien hallintaan. ISO27001 sisältää tietoturvallisuuden hallintajärjestelmän vaatimukset ja ISO27002 hyväksi havaittuja toimintatapoja (best practice) hallintajärjestelmän suunnitteluun, toteutukseen, ylläpitoon ja parantamiseen. Käytännössä ISO27001 määrittelee lokien osalta kontrolloitavat osa-alueet ja ISO27002 antaa ohjeet näiden kontrollien toteuttamiseen. (Lokiohje 2009, 27 - 28.)

4.2.3 Konfiguraatio

Lokipolitiikan pohjalta voidaan alkaa selvittämään, miten halutut tapahtumat saadaan tallennettua järjestelmistä. Useissa sovelluksissa on mahdollista määrittää millaiset tapahtumat kirjataan lokiin. Osa sovelluksista taas kirjoittaa automaattisesti kaikki tapahtumat lokiin. Käytännössä jokaisen järjestelmän kohdalla on selvitettävä dokumentaation avulla millaisia tapahtumia voidaan tallentaa ja miten konfiguraatio tulee toteuttaa.

4.2.4 Keräys

Keräyksellä tarkoitetaan lokiviestien siirtämistä tapahtuman lähteestä keskitettyyn pisteeseen prosessointia varten. Tämä piste on yleensä Unix- tai Windows-palvelin, joka suorittaa viestien parsimisen ja säilömisen tai lähettää ne edelleen toiselle palvelimelle. Keskitetyn lokipalvelimen etu on kaikkien viestien keskitetty käsittely, joka mahdollistaa tapahtumaketjujen havaitsemisen ja eri lähteistä vastaanotettujen viestien korreloimisen. Viestejä on helpompi hakea, kun ne säilytetään yhdessä paikassa. Lisäksi keskitetyn palvelimen suojaus ja varmuuskopiointi on helpompaa kuin usean hajautetun kohteen. (Chuvakin, Schmidt & Crishtopher 2012, 38.)

Lokiviestien lähetykseen etäpalvelimelle vaaditaan kuljetusprotokolla. Yleisimmin käytetty kuljetusprotokolla on lähes kaikista Unix-järjestelmistä löytyvä Syslog. Se perustuu asiakas-palvelin periaatteeseen, jossa lokinlähde toimii asiakaskomponenttina ja lähettää viestit lokipalvelimen palvelinkomponentille. Nykyisin suurin osa verkkolaitteista tukee Syslog-protokollaa viestien siirtoon. Toinen viestien kuljetukseen kykenevä sovellus on Windows Event log, jonka event collector -palvelu voi lähettää viestit toiseen Windows-järjestelmään. Windows-järjestelmille on myös olemassa useita Syslog-agentteja, jotka mahdollistavat Event logien lähetksen Syslog-protokollan avulla mihin tahansa sitä tukevaan järjestelmään. (Chuvakin, Schmidt & Crishtopher 2012, 38.)

Lokien verkon yli lähetyksessä tulee ottaa huomioon viestien luotettava toimitus ja tietoturva. Jotta viestien luotettava toimitus voidaan varmistaa, tulee lähetykseen käyttää TCP-protokollaa. Tietoturvan varmistamiseksi lokiviestien lähetykseen tulisi käyttää muusta liikenteestä eristettyä verkkoa. Lisäksi liikenne voidaan tarvittaessa salata esimerkiksi SSL-protokollalla.

4.2.5 Normalisointi ja indeksointi

Normalisoinnilla tarkoitetaan lokienhallinnan yhteydessä erimuotoisten lokiviestien muokkaamista yhdenmukaiseen formaattiin. Kuten aiemmin mainittua, kaikille lokityypeille yhteisiä kenttiä ovat käytännössä ainoastaan aikaleima, lähde ja data. Normalisoinnin tavoitteena on parsia erityyppisistä viesteistä oleelliset kentät ja muokata ne jäsenneltyyn formaattiin. Yleisiä normalisoitavia kenttiä ovat tapahtuman aikaleima, tyyppi ja vakavuus. Normalisointi helpottaa viestien myöhempää indeksointi ja analysointia. (Grimes 2010.)

Indeksi on tietorakenne, joka mahdollistaa datan haun tiettyjen kenttien sisällön perusteella. Indeksoinnin tavoitteena on nopeuttaa datan hakua, suodatusta ja raportointia. Indeksointi on helpompaa suorittaa valmiiksi parsitulle ja normalisoidulle datalle, jossa kentät on eroteltu toisistaan. (Grimes 2010.)

4.2.6 Korrelaatio ja baselining

Korrelaatiolla tarkoitetaan kykyä yhdistää erityyppiset tapahtumat yhdeksi tapahtumaksi. Esimerkiksi IDS-järjestelmä voi havaita porttiskannauksen verkkoon ja palvelin useita epäonnistuneita yrityksiä kirjautua SSH-yhteyden yli. Nämä tapahtumat yhdistämällä voidaan päätellä, että tunkeutuja etsii murrettavia kohteita verkosta. Korrelaation toteutus vaatii koko verkon lokiviestien analysointia yhdessä keskitetyssä pisteessä. Korrelaatio vaatii lokinhallintasovellukselta älykkyyttä muodostaa yksittäisistä tapahtumista suurempi kuva. Tällaisista sovelluksista käytetään nimitystä SIEM (Security Information and Event Manager). SIEM-sovellukset yhdistävät lokienhallinnan raportointiin ja tarjoavat reaaliaikaisen kuvan verkon tapahtumista. (Grimes 2010.)

Baselining tarkoittaa prosessia, jossa selvitetään millaisia tapahtumia ja kuinka paljon ympäristössä normaalisti esiintyy tietyllä aikavälillä. Esimerkiksi kuinka paljon järjestelmien kuorma on normaalitilanteessa tai kuinka monta epäonnistunutta kirjautumisyritystä tapahtuu vuorokaudessa. Baseline-tietoa voidaan käyttää hyväksi hälytysten raja-arvojen määrittämiseen. (Grimes 2010.)

4.2.7 Hälytykset ja raportointi

Kriittisistä tapahtumista, kuten järjestelmien vikailmoituksista tai murtautumisen havaitsemisesta tulisi lähettää ylläpitäjille hälytys, jotta niihin voidaan puuttua mahdollisimman nopeasti. Hälytys voidaan lähettää esimerkiksi SNMP trap -viestinä, sähköpostiviestinä tai tekstiviestinä. Hälytysten raja-arvot tulee määrittää baselinen pohjalta niin, että väärin hälytysten määrä minimoidaan. (Grimes 2010.)

Tapahtumien raportoinnin avulla voidaan kerätä tietoa erityyppisten tapahtumien määrästä pitkältä aikaväliltä. Raporteista voidaan havaita erilaisia trendejä ja järjestelmissä esiintyviä ongelmia. Pitkän aikavälin raportointi antaa ylläpitäjille tietoa esimerkiksi suorituskyvyn pullonkauloista tai järjestelmiä kohtaan tehdyistä murtoyrityksistä. (Grimes 2010.)

4.2.8 Säilytys ja poisto

Lokienhallinnan oleellinen osa on viestien säilytys. Lokiviestien säilytyksessä tulee ottaa huomioon säilöntäaika, tiedon varmistus ja tietoturva. Lokien säilöntäaikaan vaikuttaa käytettävissä oleva tallennustila, lainsäädäntö ja muut sopimukset. Käytettävissä oleva tallennustila asettaa rajoitteita säilytettävien viestien määrälle. Lisäksi ulkoiset vaatimukset, kuten lainsäädäntö, voi asettaa vaatimuksia lokien säilytysajalle.

Lokiviestit voidaan säilöä monessa muodossa. Helpoin ratkaisu on säilöä lokitiedostot suoraan tiedostojärjestelmään teksti- tai binäärimuodossa. Tiedostot tulisi nimetä selkeän kaavan mukaan viestien haun helpottamiseksi. Tiedostonimen tulisi sisältää ainakin lähteen nimi ja miltä aikaväliltä viestit on kerätty. Tiedostomuotoisen säilömisestä etuna on viestien helppo tarkastelu, siirrettävyys ja poisto.

Toinen tapa säilöä lokiviestit on tietokanta. Tietokantaan säilömisen etu on selkeä rakenne ja helppo haku SQL-kyselyiden avulla. Datan rakenteen vuoksi tietokantaan tallennetusta tiedosto on helppo muodostaa erilaisia raportteja. Lisäksi tietokannoista hakemiseen ja kirjoittamiseen on monissa ohjelmointikielissä sisäänrakennettu tuki. Huonona puolena erityisesti relaatiotietokannoissa on niiden skaalautuvuus suurien tietomäärien tallennukseen. Tietokantaan kirjoittaminen on hitaampaa kuin teksti- tai binääritiedostoihin ja ne kuluttavat myös enemmän tallennustilaa. Niiden siirto ja varmuuskopiointi on usein haasteellista. Perinteisten relaatiotietokantojen rinnalle onkin noussut uuden tyyppisiä NoSQL-tietokantoja, jotka on tarkoitettu erityisesti suurten tietomäärien tallennukseen. (Chuvakin, Schmidt & Crishtopher 2012, 78 – 79.)

Tiedon luotettavalla varmistuksella huolehditaan, että laiterikko tai muu vastaava tapahtuma ei aiheuta tiedon menetystä. Käytännössä varmistus voidaan toteuttaa säännöllisillä varmuuskopioinneilla. Lokiviestien säilytyksessä tulee varmistaa, että pääsynhallinta on toteutettu asianmukaisesti ja lokien eheys voidaan todentaa. Tallennettu tieto voidaan myös tarvittaessa suojata salaamalla.

Koska lokien säilytykseen on usein saatavilla rajallinen määrä tallennustilaa, joudutaan vanhoja viestejä ennen pitkää poistamaan. Viestien vähimmäis säilytysaika määritetään lokipolitiikassa. Vanhojen lokiviestien hävitys voidaan toteuttaa automaattisesti, kun luontipäivästä tulee kuluneeksi tietty aika. Tähän tarkoitukseen voidaan käyttää esimerkiksi Unix-järjestelmien Logrotate-sovellusta. Hävityksessä tulee myös huomioida varmuuskopioiden sisältämät viestit (Lokiohje 2009, 61).

4.2.9 Suojaus

Lokiviestit voivat sisältää tietojärjestelmien ja tietoverkon tietoturvan kannalta merkittävää tietoa sekä organisaation henkilöstöä ja asiakkaita koskevaa arkaluontoista tietoa. Tästä syystä viestien suojaamiseen tulee kiinnittää huomiota järjestelmän suunnittelusta lähtien. Lokitietoa suojattaessa on varmistettava tietoturvallisuuden kolme osa-aluetta: luottamuksellisuus, eheys ja saatavuus. Ne tulee ottaa huomioon elinkaaren jokaisessa vaiheessa. (Lokiohje 2009, 57.)

Luottamuksellisuudella tarkoitetaan, että lokitietoihin on pääsy ainoastaan valtuutetuilla henkilöillä. Luottamuksellisuus voidaan saavuttaa pääsynhallintajärjestelmällä ja tiedon salauksella. Pääsynhallinta tulee toteuttaa kaikkiin järjestelmiin, joissa lokitietoa käsitellään. Tämä koskee lokin koko elinkaarta aina tapahtuman lähteestä säilytykseen asti. Pääsynhallintajärjestelmää tulee valvoa keräämällä siitä lokitapahtumat. Tiedon salauksella pyritään varmistamaan luottamuksellisuusiinä tapauksessa, jos ulkopuolinen pääsynhallinnasta huolimatta onnistuu pääsemään lokitietoihin käsiksi. Salauksen toteutus on usein ongelmallista ja sitä käytetäänkin yleensä vain korkean suojaustason tiedon suojaamiseen.

Eheydellä pyritään varmistamaan, että tieto ei pääse muuttumaan tahatta tai tahallisesti. Eheys voidaan saavuttaa tarkistussummien avulla. Tarkistussumma perustuu tietoon että, sama data tuottaa aina saman tarkistussumman. Tällöin dataa muutettaessa, siitä laskettu tarkistussumma eroaa alkuperäisessä, jolloin tiedetään sen muuttuneen. Lokiviestien tarkistussummat tulee suojata niin, että niitä ei voi muokata tai poistaa. (Lokiohje 2009, 63.)

Saatavuudella tarkoitetaan, että tieto on käytettävissä kun sitä tarvitaan. Saatavuus voidaan varmistaa järjestelmien redundanttisuudella. Lokijärjestelmän toimintahäiriö voi johtaa tärkeiden viestien häviämiseen. Tämän vuoksi tärkeimmät komponentit olisi hyvä kahdentaa ja varmistaa jatkuva toiminta esimerkiksi klusteroinnilla. Lokijärjestelmän tilaa ja resurssien kulutusta tulisi jatkuvasti seurata, jotta voitaisiin reagoida nopeasti virhetilanteisiin ja havaita mahdolliset pullonkaulat.

4.3 Lokienhallinnan merkitys organisaation toiminnalle

Lokienhallinnalle voidaan nähdä kolme pääsyytä:

- Liiketoiminnan jatkuvuuden varmistaminen
- Toiminnan tehostaminen
- Tietoturvan parempi hallittavuus

(Sarjakivi 2013a.)

Liiketoiminnan jatkuvuudella tarkoitetaan liiketoiminnan keskeytyksetöntä harjoittamista. Sillä pyritään varmistamaan kaikkien ydinliiketoimintojen jatkuminen heti kriisin jälkeen sekä takaamaan liiketoimintojen asteittainen palautuminen kokonaisuutena ennalleen pitkäaikaisten ja vakavien häiriöiden yhteydessä. (Liiketoiminnan jatkuvuus 2013.)

Oikein toteutettu lokienhallintajärjestelmä mahdollistaa poikkeustilanteissa nopean reagoinnin tapahtumiin. Nopea pääsy oikeisiin lokitietoihin helpottaa ongelman pohjimmaisesta syystä (root cause) löytämistä ja liiketoiminnan palautumista. Nopea palvelutason palauttaminen on erityisen tärkeää tapauksissa, joissa asiakkaiden kanssa on solmittu SLA-sopimukset. Nämä sopimukset määrittävät usein tietyn saatavuuden palvelulle, josta poikkeaminen johtaa rahallisiin sanktioihin. Järjestelmien toiminnasta saatavien tietojen avulla mahdolliset ongelmat voidaan havaita sekä korjata ennakkoon ja näin välttyä liiketoiminnan häiriöiltä.

Ulkoiset tahot voivat velvoittaa organisaation lokienhallintajärjestelmän toteuttamiseen, jolla halutaan usein varmistaa liiketoiminnan jatkuminen ja riittävä tietoturvan taso. Lokienhallintaan velvoittavat mm. tietoturva-asetus, KATAKRI, PCI DSS, sosiaali- ja terveysministeriön KanTa sekä standardit, kuten ISO27001 ja Cobit (Sarjakivi 2013a).

Lokienhallinnasta kannattaa ottaa kaikki hyöty irti organisaation toiminnan tehostamisessa. Lokijärjestelmät tuottavat valtavan määrän tietoa, jota voidaan hyödyntää liiketoiminnassa. Tästä suuresta informaatiomäärästä käytetään usein ympäryöreää nimitystä ”big data”. Big data termillä tarkoitetaan massiivista jäsentelyä ja jäsentelemätöntä datamäärää, jota on vaikea prosessoida perinteisillä menetelmillä (Big data 2013). Termiä voidaan käyttää myös kuvaamaan työkaluja ja prosesseja tiedon käsittelyyn ja säilömiseen.

IBM:n julkaiseman raportin mukaan jopa 73 % big datasta on lokitietoa (Schroeck, Shockley, Smart, Romero-Morales & Tufano 2012). Tätä informaatiota voidaan hyödyntää käyttäjien profiloinnissa ja käyttötottumusten havainnoinnissa. Analysoidun tiedon perusteella tuotteita ja palveluja on mahdollista kehittää vastaamaan paremmin asiakkaan tarpeita. Tietoa voidaan myös hyödyntää raportoinnissa ja tukena järjestelmähankintoja suunniteltaessa.

Lokienhallinta on oleellinen osa tietoturvanhallintajärjestelmää. Tapahtumien lokiin kirjaus mahdollistaa tietoturvapoikkeamien havaitsemisen ja tapahtumien kulun selvittämisen. Esimerkiksi tietomurron yhteydessä lokitiedon avulla voidaan selvittää mihin järjestelmiin tunkeutuja on päässyt, millaisia toimia hän on tehnyt ja milloin murto on tapahtunut. Tätä tietoa voidaan käyttää myöhemmin mahdollisessa oikeusprosessissa todistusaineistona.

Lokienhallinta on yhdistetty tietoturvanhallintaan SIEM-järjestelmissä. Nimi viittaa järjestelmiin, joissa yhdistävät aiemmin erilliset SIM (Security Information Management) ja SEM (Security Event Management) -komponentit. SIEM-järjestelmät pyrkivät mahdollisimman reaaliaikaisen tapahtumien analysointiin, korrelaatioon ja hälytyksiin. Verkon laitteet syöttävät lokitietoa SIEM-järjestelmään, joka analysoi datan tiettyjen algoritmien perusteella ja suorittaa analyysin pohjalta tietyt toimenpiteet. Tavoitteena on nopeuttaa tietoturvapoikkeamiin reagointia. Lisäksi järjestelmät sisältävät työkaluja raportointiin ja visualisointiin. (Jamil 2009.)

4.4 Lokienhallinnan haasteet

Lokienhallintaan liittyy useita haasteita joita järjestelmää suunniteltaessa tulee ottaa huomioon. Haasteet voidaan jaotella karkeasti keräämiseen ja säilytykseen sekä suojaukseen ja analysointiin. Kaikkien osa-alueiden pohjimmaisena haasteena on järjestelmän skaalautuvuus jatkuvasti kasvavalle lokimäärälle. (Kent & Souppaya 2006, 2-8 – 2-10.)

Keräämisen ja säilytyksen haasteet liittyvät lähteiden sekä formaattien suureen määrään. Suurissa organisaatioissa lokia tuottavia lähteitä voi olla satoja ja ne voivat sijaita eri fyysisissä paikoissa, kuten sivukonttoreissa. Lokien kerääminen ja siirto luotettavasti kaikista kohteista keskitettyyn järjestelmään on todella haasteellista ja vaatii kohteiden välille hyvät tiedonsiirtoyhteydet. Lokienhallinta voikin olla järkevämpää hajauttaa useaan toimipisteeseen.

Eri järjestelmät voivat tallentaa hyvin erilaista tietoa lokiin. Osa tallentaa hyvin paljon tietoa tapahtumista, kun taas osa vain minimaalisen määrän. Usein sama tieto on vielä esitetty hieman eri tavalla, josta yleisin esimerkki ovat erityyppiset aika-
leimat. Lokilähteet voivat tuottaa tiedon eri formaatissa, kuten raakatekstinä tai XML-muotoiltuna. Lisäksi kentät voivat olla eroteltuna toisistaan eritavoin formaatista riippuen, esimerkiksi pilkulla tai hakasulkeilla. Epäyhtenäinen formaatti hankaloittaa analysointia ja korrelaation muodostamista tapahtumien välille. Käytännössä viestit täytyy parsia ja normalisoida ennen kuin niitä voidaan kunnolla hyödyntää. Lokien säilytykseen liittyvät haasteet on käyty läpi luvussa 1.2.5. (Kent & Souppaya 2006, 2-8, 2-10.)

Lokitietojen suojauksen haasteena on sen toteutus koko elinkaaren aikana, aina viestin tuottamisesta arkistointiin. Lokijärjestelmä koostuu erilaisista komponenteista, jotka voivat erota ominaisuuksiltaan huomattavasti toisistaan. Esimerkiksi eri laitevalmistajien tuki pääsynhallinta ja salausmenetelmille voi vaihdella, eivätkä vanhemmat laitteet välttämättä tue niitä ollenkaan. Valmistajien toteutukset voivat olla yhteensopimattomia keskenään. Lisäksi salauksen käyttö voi heikentää lokijärjestelmän suorituskykyä. Järjestelmää suunniteltaessa on syytä punnita suojauksen tarve siitä koituvia haittoja vastaan ja yrittää löytää hyvä kompromissi näiden välille.

Lokien analyysi on tärkeä osa lokienhallintaa, mutta se jää organisaatiossa usein pienelle huomiolle. Siihen tartutaan vasta ongelmien ilmaannuttua, sen sijaan että pyrittäisiin ennakoimaan ongelmien synty etukäteen analysoimalla päivittäisiä tapahtumia. Tehtävästä vastuussa ovat usein tietojärjestelmien ylläpitäjät, joilla ei usein ole aikaa perehtyä lokitietoihin muiden kiireellisimpien tehtävien vuoksi. Analysointiin ei myöskään aina ole saatavilla tehokkaita työkaluja. Ilman kunnollista prosessia lokien analysointiin, lokitiedon arvo laskee. (Kent & Souppaya 2006, 2-10.)

Lokien analysoinnin haasteena on oleellisten tapahtumien erottaminen muun tiedon joukosta. Analysointiin tarkoitetut työkalut pystyvät seuraamaan lokeja reaaliajassa sekä korreloimaan tapahtumia ja erottamaan lokeista erilaisia kuvioita. Nämä toiminnot ovat usein automatisoituja ja osaavat ilmoittavat havaitsemistaan poikkeamista ylläpidolle. (Kent & Souppaya 2006, 2-10.)

Suurin osa lokienhallintaan liittyvistä haasteista voidaan ratkaista muutamaa toimintatapaa noudattamalla:

- Lokienhallinnan priorisointi organisaatiossa. Lokienhallinta voidaan priorisoida organisaatiossa arvioimalla siihen liittyvät riskit ja tasapainottamalla ne toimintaan tarvittavien resurssien ja ajan kanssa.
- Poliitiikan ja toimintatapojen määrittäminen. Organisaation lokipolitiikan tulee sisältää lokienhallinnan päämäärän ja vaatimukset sekä toimintatavat, joilla niihin pyritään. Poliitiikkaa noudattamalla varmistetaan yhtenäiset toimintatavat ja organisaation sisäisten sekä ulkoisten vaatimusten täyttyminen.
- Järjestelmän jatkuva ylläpito. Lokienhallintajärjestelmän tilaa tulee seurata ja varmistaa resurssien riittävyys tiedon käsittelyyn. Ylläpidolla pyritään varmistamaan järjestelmän jatkuva häiriötön toiminta.

(Kent & Souppaya 2006, 2-10.)

5 Syslog

5.1 Yleistä

Syslog on protokolla lokiviestien hallintaan ja välitykseen. Se on standardoitu IETF:n RFC-dokumentissa 5424, joka korvasi vanhan RFC 3164 -standardin. Protokollasta on muodostunut de facto -standardi lokiviestien välitykseen.

Protokollan kehitti alun perin Eric Allman BSD (Berkley Software Distribution) -käyttöjärjestelmän sendmail-sovelluksen käyttöön. Protokolla yleistyi pian myös muissa käyttöjärjestelmissä ja laitteissa. Standardin puutteen vuoksi protokollasta on ollut käytössä useita, joskus yhteensopimattomia toteutuksia, jotka ovat pyrkineet parantamaan protokollan tietoturvaa ja joustavuutta. Vuonna 2001 IETF julkaisi RFC 3164 -dokumentin: "The BSD Syslog Protocol", joka oli ensimmäinen standardi Syslog-protokollasta. Toinen versio standardista julkaistiin RFC 5424 -dokumentissa, joka sisältää parannuksia protokollan tietoturvaan. (Eaton 2003, 7.)

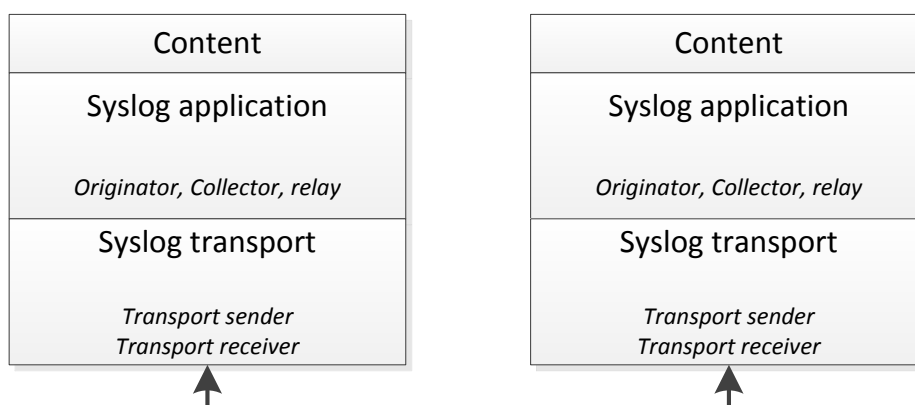
5.2 Protokollan rakenne

Syslog-protokolla perustuu kerroksittaiseen arkkitehtuuriin, jonka tarkoitus on erottaa viestien sisältö kuljetusmetodista ja samalla mahdollistaa helppo laajennettavuus eri kerroksilla. Standardi määrittää viestien formaatin, mutta ei ota kantaa viestien säilytykseen.

Protokolla koostuu kolmesta kerroksesta: viestien sisältö, sovellus ja kuljetus. Viestin sisältö on nimensä mukaisesti Syslog-viestien sisältämä informaatio. Sovelluskerros huolehtii viestien tuottamisesta, tulkinnasta, reitityksestä ja säilytyksestä. Kuljetustason tehtävä on viestien lähetys ja vastaanotto siirtotien yli. (Gerhards 2009, 4.)

Eri kerroksilla tapahtuu tiettyjä funktioita. Viestin lähde (originator) tuottaa sisällön viestiin. Kerääjä (collector) kokoaa viestit analysointia varten. Välittäjä (relay) vastaanottaa viestejä kerääjiltä tai muilta välittäjiltä ja lähettää ne edelleen toisille kerääjille tai välittäjille. Lähettäjä (transport sender) syöttää viestit kuljetusprotokollalle ja vastaanottaja (transport receiver) ottaa ne vastaan. Originator-, collector- ja relay-funktiot voivat kuulua samaan järjestelmään. (Gerhards 2009, 5.)

Kuviossa 1 on esitetty Syslog protokollan tasot ja niihin liittyvät funktiot.



Kuvio 1. Syslog-tasot ja funktiot

RFC 5424 määrittelee Syslog-viestien formaatin kuljetuskerroksen protokollasta riippumattomana. Viestien lähetys eri kuljetuskerroksen protokollien yli on määritetty omilla dokumenteillaan, esimerkiksi UDP RFC 5426 ja TCP RFC 6587. Kuljetusprotokollat eivät saa tarkoituksellisesti muokata viestejä ja mahdolliset lähetyksen aikaiset muutokset tulee palauttaa ennen viestin luovuttamista sovellustason protokollalle. (Gerhards 2009, 7.)

Syslog koostuu syslogd-palvelusta, joka käynnistetään yleensä järjestelmän käynnistyksen yhteydessä. Syslogd kommunikoi sovellusten kanssa Syslog-kirjaston kutsujen avulla ja vastaanottaa viestit Unix-sokettien välityksellä. Viestit voidaan vastaanottaa myös verkon yli UDP- tai TCP-protokollalla. (Chuvakin, Schmidt & Crishtopher 2012, 52.)

5.3 Viestiformaatti

Syslog viesti koostuu kolmesta osasta: PRI, HEADER ja MSG. Paketin maksimipituus voi olla RFC 3164 dokumentin mukaan 1028 tavua (Lonvick 2001, 8). Uudempi RFC 5424 määrittää koon riippuvaksi käytetystä kuljetusprotokollasta ja asettaa viestin minimipituudeksi 480 tavua (Gerhards 2009, 9).

5.3.1 PRI

PRI-kenttä määrittää viestin facility- ja severity-arvot. Se koostuu kolmesta, neljästä tai viidestä merkistä, joista ensimmäinen on < (suurempi kuin) ja viimeinen > (pienempi kuin). Näiden kahden merkin sisällä olevaa numeroa kutsutaan priority arvoksi (PRIVAL), joka sisältää sekä facility- että severity-arvot. Priority voi sisältää maksimissaan kolme desimaalinumeroa. Priority voidaan laskea facility- ja severity-arvoista kertomalla facility kahdeksalla ja lisäämällä tulokseen severity. Esimerkiksi facility-arvo 3 (System daemons) ja severity-arvo 6 (Informal) tuottavat priority-arvon 30. (Gerhards 2009, 9 – 10.)

Facility-arvo ilmaisee lokitapahtuman tuottaman sovelluksen tai prosessin tyyppin, esimerkiksi useille järjestelmän taustaprosesseille on olemassa oma facility. Severity-arvo ilmaisee tapahtuman vakavuuden. Facility- ja severity-arvojen käyttö ei ole pakollista, mutta hyvin yleistä. Taulukossa 1 on esitetty eri facility-arvot ja taulukossa 2 severity-arvot. (Gerhards 2009, 10 – 11.)

Taulukko 1. Syslog facility arvot

Koodi	Facility
0	Kernel messages
1	User-level messages
2	Mail system
3	System daemons
4	Securit/authorization messages
5	Messages generated internally by syslogd
6	Line printer subsystem
7	Network news subsystem
8	UUCP subsystem
9	Clock daemon
10	Security/authorization messages
11	FTP daemon
12	NTP subsystem
13	Log audit
14	log alert
15	Clock daemon
16	Local use 0
17	Local use 1
18	Local use 2
19	Local use 3
20	Local use 4
21	Local use 5
22	Local use 6
23	Local use 7

Taulukko 2. Syslog severity arvot

Koodi	Severity
0	Emergency: system is unusable
1	Alert: action must be taken immediately
2	Critical: critical conditions
3	Error: error conditions
4	Warning: warning conditions
5	Notice: normal but significant condition
6	Informational: informational messages
7	Debug: debug-level messages

5.3.2 HEADER

Syslog-viestin HEADER osan tulisi RFC 5424 -dokumentin mukaan sisältää VERSION-, TIMESTAMP-, HOSTNAME-, APP-NAME-, PROCID- ja MSGID-kentät. Kenttien formaatti kuitenkin vaihtelee Syslog-toteutuksesta riippuen.

VERSION-kenttä määrittää käytettävän Syslog-protokollan version. Versiota tulee kasvattaa aina, kun protokollan uusi versio sisältää muutoksia HEADER-kenttiin. (Gerhards 2009, 11.)

TIMESTAMP-kenttä sisältää tapahtuman aikaleiman. RFC 5424 määrittää aikaleiman formaatille tietyt rajoitteita: T- ja Z-merkkien tulee olla suuraakkosia, T merkin käyttö on pakollista ja karkaussekunteja ei sallita. Lisäksi aikaleima tulisi ilmoittaa lähteen kellon tarkkuuden salliessa sekunnin murto-osina tai NILLVALUE-arvona, jos kellonaikaa ei voida selvittää. Syslog-toteutusten aikaleimojen formaatit usein vaihtelevat, koska standardi ei määrittele sitä yksiselitteisesti. (Gerhards 2009, 11 – 12.)

HOSTNAME-kenttä kertoo viestin alun perin lähettäneen koneen nimen. Kentän tulee sisältää alkuperäisen lähettäjän FQDN (Fully Qualified Domain Name) -nimi eli hostname ja domain name. Jos Syslog-sovellus ei pysty selvittämään FQDN-nimeä, voidaan kentässä käyttää myös pelkkää hostnamea, IP-osoitetta tai NILLVALUE-arvoa. Samaa arvoa tulee käyttää yhtenäisesti eri viesteissä. (Gerhards 2009, 13.)

APP-NAME-kenttä identifioi viestin tuottaneen sovelluksen tai laitteen. NILLVALUE-arvoa käytetään, jos viestin tuottajaa ei voida selvittää. PROCID-kentän sisältö riippuu Syslog-toteutuksesta, mutta useimmiten sitä käytetään ilmaisemaan Syslog-prosessin id tai tapahtuman tuottaneen sovelluksen prosessi-id. Sitä voidaan myös käyttää tunnistamaan samaan ryhmään kuuluvat viestit. MSGID-kenttä kertoo viestin tyyppin. Esimerkiksi reititin voi käyttää MSGID-arvoa SNMP_TRAP_LINK_DOWN linkin katketessa. MSGID-kenttää voidaan käyttää suodattamaan lokiviesteistä tietyn tyyppiset tapahtumat. (Gerhards 2009, 14 – 15.)

5.3.3 MSG

Syslog-viestin MSG osa sisältää vapaamuotoisen kuvauksen tapahtumasta. Viestin sisältö tulisi olla UTF-8 koodattua. Muita koodauksia voidaan käyttää, jos Syslog-sovellus ei tue UTF-8 koodausta. (Gerhards 2009, 18 – 19.)

5.3.4 Esimerkit Syslog-viesteistä

Kuviossa 2 on esitetty esimerkki syslog viestistä.

Aug 31 14:25:20	localhost	kernel: e1000: eth1 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
Timestamp	Hostname	App. Message

Kuvio 2. Syslog-viesti esimerkki 1

Kyseinen viesti on peräisin CentOS Linux -käyttöjärjestelmästä. Ensimmäisenä kenttänä viestissä on aikaleima. Aikaleima on ilmaistu muodossa: kuukausi, päivä, tunnit, minuutit, sekunnit. Kyseinen muotoilu ei noudata RFC 5424 – dokumentin määrittämää muotoa. Seuraavat kentät ovat järjestelmän hostname (localhost) ja viestin tuottanut sovellus (kernel). Viimeisenä kenttänä on itse viesti.

Edellä nähty lokiviesti on rakenteeltaan selkeä ja noudattaa pääpiirteittäin RFC-standardia, jättäen osan kentistä pois. Kuviossa 3 nähdään toista ääripäätä edustava esimerkki.

```
<14>Oct 20 10:33:06 1,2009/10/20
10:33:06,0004C100661,TRAFFIC,end,1,2009/10/20
10:33:04,10.0.30.31,10.60.0.34,0.0.0.0,0.0.0.0,rule1,,,syslog,vsys1,trust,u
ntrust,ethernet1/2,ethernet1/1,af ALL to SYSLOG,2009/10/20
10:33:06,350,1,32773,514,0,0,udp,allow,127032,127032,127032,624,
2009/10/20 10:22:26,608,any,0
```

Kuvio 3. Syslog-viestiesimerkki 2

Kuvion 3 lokiviesti on peräisin Palo Alto Networks palomuurien PanOS-käyttöjärjestelmästä. Viesti sisältää paljon informaatiota eikä noudata RFC-standardin muotoilua. Sisällön ymmärtämiseksi täytyy turvautua valmistajan dokumentaatioon.

5.4 Tietoturva

Syslog-protokollaa käytettäessä tulee ottaa huomioon tiedon luottamuksellisuus, eheys ja saatavuus. Erityisesti nämä tietoturvan osa-alueet on huomioitava viestien etäpalvelimelle lähetyksessä, kun viestit kulkevat epäluotettavien verkkosegmenttien yli.

Protokollaan kohdistuvia tietoturvauhat ovat:

- Viestien lähteenä tai kohteena esittäytyminen. Hyökkääjä voi huijata siirron toista osa-puolta, joko lähettämällä vastaanottajalle omia viestejä tai huijaamalla lähde lähettämään viestit itselleen. Hyökkäyksen tarkoituksena on saada väärennetyillä viesteillä kohde luulemaan järjestelmissä tapahtuvan jotain ei oikeasti tapahdu, tai kerätä tietoa järjestelmistä lokiviestien avulla.
- Viestien muokkaus. Lähteen ja kohteen välissä toimiva hyökkääjä voi kaapata viestit, muokata niitä ja lähettää edelleen kohteelle. Vastaanottaja voi tulkita muokatun viestin väärin, joka voi johtaa häiriöön tai pahimmassa tapauksessa järjestelmän kaatumiseen. Hyökkäyksen tarkoitus on häiritä kohteen toimintaa.
- Tietojen urkinta. Luvaton henkilö voi päästä käsiksi tietoihin organisaation verkoista ja tietojärjestelmistä lokiviestejä tutkimalla. Syslog-viestit lähetetään selkokieლისenä ilman minkäänlaista salausta. Hyökkääjä voi lokitietoa hyödyntämällä kartoittaa verkon rakennetta, käytettäviä sovelluksia ja niiden versioita sekä käyttäjiä.
- Palvelunesto. Hyökkääjä voi suorittaa lokijärjestelmää kohtaan palvelunestohyökkäyksen esimerkiksi lähettämällä niin paljon dataa viestien kerääjälle, että sen toiminta häiriintyy. Tällä voidaan pyrkiä peittelemään muuta haitallista toimintaa, kuten järjestelmiin murtautumista.

(Miao, Ma & Salowey 2009.)

Yllä mainittujen uhkien torjumiseksi Syslog-viestit voidaan siirtää TLS (Transport Layer Security) -protokollan yli. TLS mahdollistaa sisällön luottamuksellisuuden vahvalla salauksella sekä eheyden ja osapuolten todentamisen Message Authentication Coden (MAC) avulla.

Viestien lähde toimii TLS-protokollan asiakkaana (client) ja vastaanottaja palvelimena (server). Asiakas ja palvelin suorittavat kättelyn, jossa ne sopivat käytettävät salaiset avaimet, salausmenetelmät ja todentavat toistensa henkilöllisyyden. Asiakas ja palvelin suorittavat todennuksen käyttäen sertifikaatteja, jolloin molemmat varmistavat vastapuolen sertifikaatin oikeellisuuden ja että vastapuolella on oikea yksityinen avain. Kättelyn päättyessä salattu sessio on muodostettu ja viestejä voidaan alkaa lähettää. (Miao, Ma & Salowey 2009.)

TLS avulla voidaan varmistaa viestien siirron turvallisuus, mutta myös lähde- ja kohdejärjestelmien tietoturvallisuus tulee ottaa huomioon. Järjestelmiin pääsy tulee rajata ainoastaan ylläpidon käyttöön ja hallinta tulee suorittaa suojattujen yhteyksien yli muusta liikenteestä eristetyistä verkkosegmentistä. Järjestelmät tulisi koventaa ottamalla kaikki ylimääräiset palvelut pois käytöstä ja suorittamalla päivitykset säännöllisesti.

5.5 Syslog-toteutukset

5.5.1 Syslog-ng

Syslog-ng on BalaBit IT Securityn kehittämä Unix-tyyppisille käyttöjärjestelmille suunniteltu Syslog toteutus. Tuotteesta on olemassa ilmainen avoimen lähdekoodin versio (OSE – Open Source Edition) ja maksullinen lisäominaisuuksia sisältävä versio (PE – Premium Edition). (The Foundation of Log Management 2013.)

Syslog-ng tukee IETF:n määrittämien Syslog-standardien RFC 3164 ja RFC 5424 lisäksi useita kehittyneitä ominaisuuksia. Se mahdollistaa viestien lähetyksen turvallisesti ja luotettavasti TLS- ja TCP-protokollien avulla kerääjältä etäpalvelimelle. Palvelin voi lähettää viestit myös AMQP-vaihteeseen. Viestejä on mahdollista lajitella ja suodattaa palvelimella sisällön perusteella ja tallentaa omiin tiedostoihin tai tauluihin tietokannassa. Syslog-ng pystyy vertaamaan lokiviestejä ennalta määritettyihin malleihin ja niiden perusteella luokittelemaan, merkitsemään ja korreloimaan viestejä. (Product features and benefits 2013.)

Syslog-ng on saatavilla useille järjestelmäarkkitehtuureille (x86, x86_64, SUN Sparc, PowerPC, Alpha, ARM, MIPS) ja käyttöjärjestelmille (Linux, BSD, Solaris, IBM AIX, HP-UX). Lisäksi esimerkiksi Amazon Kindle sekä F5 ja Endian verkkolaitteet hyödyntävät sovellusta. (Platform support 2013.)

5.5.2 Rsyslog

Rsyslog on laajasti käytetty avoimen lähdekoodin Syslog-toteutus Unix-tyyppisille käyttöjärjestelmille. Projekti sai alkunsa vuonna 2004 kilpailijana Syslog-ng -sovellukselle. Rsyslog on saatavilla ilmaiseksi, mutta sille saa myös maksullista tukea.

Rsyslog sisältää hyvin samoja ominaisuuksia kuin Syslog-ng, esimerkiksi TCP ja TLS tuen, viestien suodatuksen ja parsimisen sekä tietokantaan tallennuksen. Viestejä voidaan myös puskuroida paikallisesti, mikäli vastaanottaja ei ole valmis. Eräs uusi ominaisuus on integraatio Hadoop-ohjelmistoalustan kanssa. Hadoop on suunniteltu sovellusten ajamiseen suurissa klustereissa. (Features 2013.)

Rsyslog on laajasti käytössä Unix-pohjaisissa käyttöjärjestelmissä, esimerkiksi Debian, Red Hat, Ubuntu ja openSUSE käyttävät sitä oletuksena. Lisäksi se on saatavilla muun muassa BSD-, Solaris ja AIX-alustoille. (Platforms 2013.)

6 SNMP

6.1 Yleistä

SNMP (Simple Network Management Protocol) on IP-verkkojen yli tapahtuvaan laitteiden hallintaan suunniteltu protokolla. IETF julkaisi ensimmäisen version protokollasta vuonna 1988 ja nykyisen version kolme vuonna 2004. SNMP-protokollasta on muodostunut standardi tapa laitteiden hallintaan, jota lähes kaikki verkkoon liittyvät laitteet tukevat.

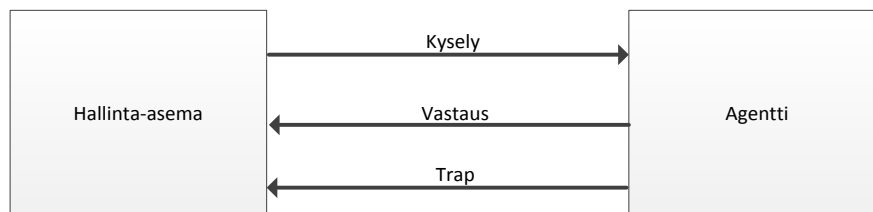
Vaikka SNMP:n varsinainen tarkoitus ei ole toimia lokijärjestelmänä, sen tuottamat trap-viestit voidaan luokitella lokiviesteiksi. Vanhemmat laitteet eivät välttämättä tue lokiviestien lähetystä Syslog-protokollalla, jolloin SNMP trap -viestejä voidaan hyödyntää tapahtumatietojen keräämiseen. Trap-viestit voivat joissain tapauksissa sisältää tietoja, joita Syslog-protokollalla ei voida kerätä. (Chuvakin, Schmidt & Crishtopher 2012, 59.)

6.2 Arkkitehtuuri

SNMP-arkkitehtuuri koostuu hallinta-asemista ja agenteista. Hallinta-asema on palvelin, joka sisältää verkonhallintaan tarkoitetun sovelluksen. Sen tehtävä on tietojen kysely ja trap-viestien vastaanotto agenteilta. Kyselyjen lisäksi hallinta-asema voi myös asettaa arvoja hallittavaan laitteeseen. Hallinta-asemasta käytetään usein nimitystä NSM (Network Management Station). Agenti on hallittavissa laitteissa sijaitseva sovellus, joka vastaa hallinta-aseman pyyntöihin ja lähettää sille trap-viestejä tapahtumista. Sen tehtävä on välittää hallinta-asemalle tietoa järjestelmässä tapahtuvista toiminnoista ja muutoksista, esimerkiksi havaituista virheistä. Agenti voi olla laitteen käyttöjärjestelmään sisäänrakennettu ominaisuus tai erikseen asennettava palvelu. (Mauro & Schmidh 2005, 3.)

Hallinta-asema voi lähettää kyselyn agentille koskien valvottavan järjestelmän tiettyä osa-aluetta, esimerkiksi prosessorikuormaa. Agentti vastaa kyselyyn pyydettyllä tiedolla, jonka perusteella hallinta-asema voi suorittaa tarvittavat toimenpiteet. Agentti voi lähettää määrätystä tapahtumasta automaattisesti tiedon hallinta-asemalle trap-viestin avulla, jolloin hallinta-aseman ei tarvitse erikseen sitä kysellä. (Mauro, Schmidh 2005, 4)

Kuviossa 4 on esitetty SNMP hallinta-aseman ja agentin välinen toiminta.



Kuvio 4. SNMP hallinta-aseman ja agentin suhde

6.3 SMI ja MIB

SMI (Structure of Management Information) käytetään määrittämään objektit ja niiden käyttäytyminen. Esimerkki kyseisistä objekteista on verkkolaitteen kotelon lämpötila, joka voi olla numeerinen arvo. Hallinta-asema käyttää objektien tietoa hyväkseen agentin kautta, joka hallinnoi listaa valvottavista objekteista järjestelmässä. (Mauro & Schmidh 2005, 4.)

MIB (Management Information Base) on tietokantamainen rakenne järjestelmän hallittavista objekteista. SMI tarjoaa keinon objektien määrittämiseen, kun taas MIB sisältää määrittämisen itse objekteista. MIB on järjestetty puumaiseen rakenteeseen, jonka lehtiä ovat objektit. OID (Object Identifier) -arvot ovat ASN.1-standardin mukaan määritettyjä arvoja, joita käytetään yksilöimään objektit MIB-rakenteessa. Agentti voi käyttää useita MIB-tietokantoja, joista MIB-II on standardin määrittämä ja pakollinen kaikille agenteille. Laittevalmistajat lisäävät usein laiteisiinsa omat MIB-muuttujat, joita MIB-II ei sisällä. (Mauro & Schmidh 2005, 4 – 5.)

6.4 Versiot

SNMP versio 1 (RFC 1157) oli ensimmäinen versio protokollasta, joka määrittä sen arkkitehtuurin ja toimintaperiaatteen. Ensimmäisen version tietoturva perustuu community-merkkijonoon, jonka avulla määritetään pääsy laitteen hallintatietoihin. Community-arvoja on tyypillisesti kolme: read-only, read-write ja trap. Community lähetetään SNMP-viestien mukana selkokielenä. Versio 2 (RFC 3416-3418) pyrki parantamaan edellisen version heikkouksia muun muassa tietoturvan osalta esittelemällä uuden tietoturvamallin. SNMPv2c (RFC 1901-1903) käyttää perinteistä community-perusteista tietoturvamallia ja siitä on muodostunut de facto SNMPv2-standardi. Versio 3 (RFC 3411-3418) on uusin versio protokollasta. Se sisältää osapuolten todennuksen, vahvan salauksen ja viestien eheyden varmistuksen. Nykyisen monet laitteet tukevat kaikkia kolmea standardia. (Mauro & Schmidh 2005, 2.)

6.5 SNMP lokitapahtumien keräämiseen

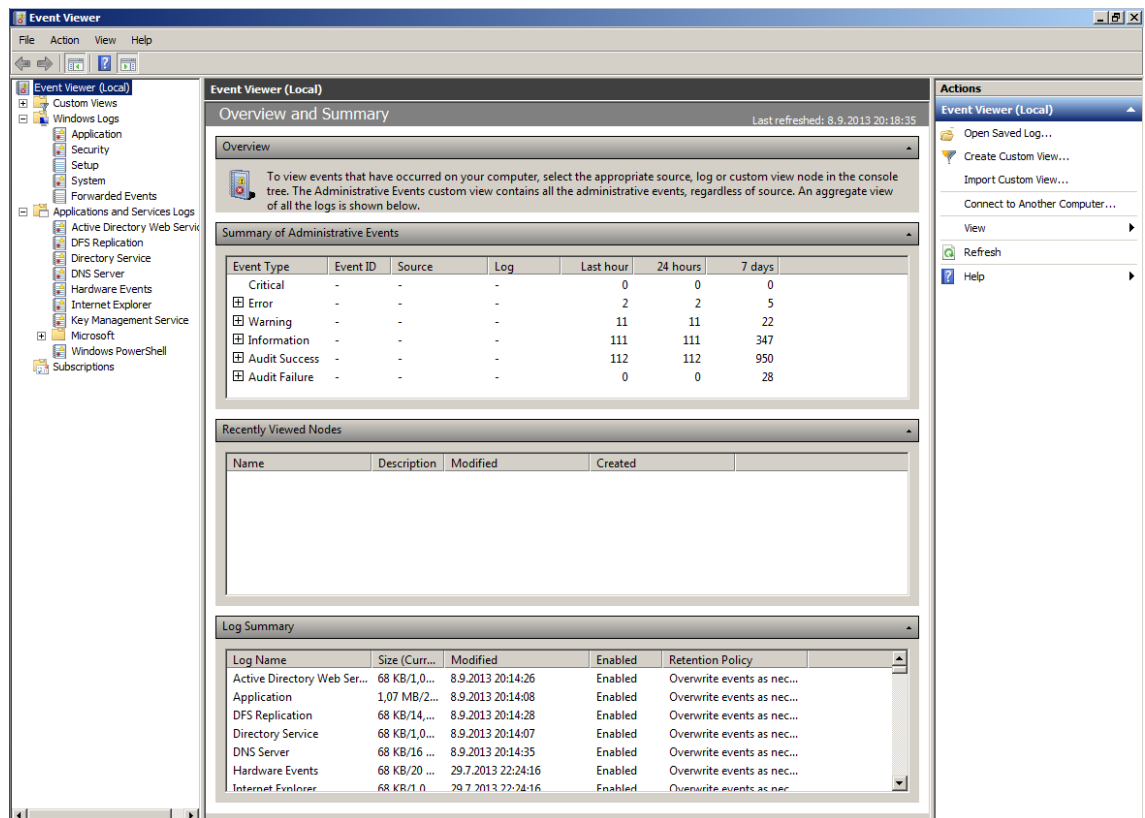
SNMP määrittää laitteen tukemat trap-viestit MIB-tietokannoissa. Vaikka MIB-tietokannat mahdollistavat hyvin muotoiltujen ja ymmärrettävien viestien tuottamisen, laitevalmistajat eivät usein noudata sitä omissa toteutuksissaan. Esimerkiksi MIB-arvo ei välttämättä vastaa laitteen lähettämää trap-viestiä tai valmistaja voi tehdä muutoksia SNMP-toteutukseen päivittämättä MIB-tietokantaa. Tästä seuraa, että SNMP trap -viestien formaatti vaihtelee valmistajan toteutuksesta riippuen, mikä tekee parsinnasta yhtä hankalaa kuin Syslog-viestien kohdalla. (Chuvakin, Schmidt & Crishtopher 2012, 60 – 62.)

SNMP ei ole ideaalein ratkaisu lokiviestien keräämiseen, eikä tarkoitusta varten suunniteltu. Sitä voidaan kuitenkin hyödyntää Syslog-protokollan rinnalla keräämään viestejä laitteista, jotka eivät tue Syslogia.

7 Windows Event log

Microsoft Windows -käyttöjärjestelmät sisältävät oman lokien tuottamiseen ja keräämiseen tarkoitetun järjestelmän nimeltä Event log. Se on ollut sisäänrakennettuna Windows-käyttöjärjestelmässä ensimmäisestä NT-versiosta lähtien. Event log tallentaa lokiviestit järjestelmään binäärimuodossa, eikä niitä voi tarkastella ilman erillistä sovellusta. Windows-käyttöjärjestelmät sisältävät Event Viewer nimisen sovelluksen lokiviestien tarkasteluun ja hallintaan. Event logien hallintaan on saatavilla myös useita kolmansien osapuolten kehittämiä työkaluja.

Kuviossa 5 nähdään Event Viewer -työkalun pääikkuna.



Kuvio 5. Event Viewer pääikkuna

7.1 Tapahtumatyypit

Windows-järjestelmistä voidaan kerätä pääasiassa kahden tyyppisiä tapahtumia: windows-lokit sekä sovellus- ja palvelulokit. Windows-tyypin lokien tarkoitus on kerätä viestejä legacy-sovelluksista ja koko järjestelmää koskevista tapahtumista. Ne on jaettu edelleen application-, security-, setup-, system- ja forwardedevent-kategorioihin. Application-lokit sisältävät sovellusten tuottamia lokiviestejä. Security-loki sisältää pääsynhallintaan ja resurssien käyttöön liittyvät lokit, kuten käyttäjien kirjautumiset ja tiedostojen muokkaukset. Järjestelmän ylläpitäjä voi valita, mitä security-lokeja kerätään. Setup-loki kattaa nimensä mukaisesti sovellusten asennukseen liittyvät tapahtumat. System-loki sisältää Windows-järjestelmän komponenttien tuottamat viestit, esimerkiksi tietyn järjestelmäpalvelun käynnistykseen epäonnistumisen. Forwardedeventlogs-kategoriaan kuuluvat muista järjestelmistä vastaanotetut lokiviestit. (Event Logs 2013.)

Sovellus- ja palvelulokit keräävät viestejä yksittäisistä sovelluksista tai komponenteista. Ne on jaettu neljään kategoriaan: admin, operational, analytic ja debug. Admin-kategorian lokit ovat tarkoitettu pääasiassa järjestelmän ylläpitäjille. Ne sisältävät ongelman ja hyvin määritellyn ratkaisun, jonka perusteella ylläpitäjä voi toimia. Admin-kategoriaan tapahtumat ovat hyvin dokumentoituja tai sisältävät suoraan ohjeen ongelman ratkaisuun. Operational-kategoriaan kuuluvia lokeja käytetään ongelmien diagnosointiin ja analysointiin. Analytic-kategorian tapahtumat kuvaavat sovellusten operaatioita ja osoittavat ongelmia, joihin ei voi vaikuttaa käyttäjän toimilla. Debug-kategoria on tarkoitettu sovelluskehittäjille ongelmienratkintaan. (Event Logs 2013.)

7.2 Tapahtumien rakenne

Event log -järjestelmä kirjoitettiin kokonaan uudestaan Windows Vistassa. Uudella järjestelmällä pyrittiin parantamaan tapahtumien seuranta, skaalautuvuutta suuriin viestimääriin, suorituskykyä ja tietoturvallisuutta. Yksi keskeisistä parannuksista on helpottaa kiinnostavien viestien suodatusta muiden tapahtumien joukosta. Tämä perustuu viestien tallentamiseen XML-skeema rakenteeseen. XML-rakenne mahdollistaa tapahtumien hakemisen ja suodatuksen määritettyjen kenttien perusteella. (Menn 2006.)

Kuviossa 6 on esitetty XML-muotoinen lokitapahtuma.

```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
- <System>
  <Provider Name="Microsoft-Windows-Security-Auditing" Guid="{54849625-5478-4994-A5BA-3E3B0328C30D}" />
  <EventID>4672</EventID>
  <Version>0</Version>
  <Level>0</Level>
  <Task>12548</Task>
  <Opcode>0</Opcode>
  <Keywords>0x8020000000000000</Keywords>
  <TimeCreated SystemTime="2013-09-08T17:14:28.619188600Z" />
  <EventRecordID>7799</EventRecordID>
  <Correlation />
  <Execution ProcessID="460" ThreadID="560" />
  <Channel>Security</Channel>
  <Computer>WIN-EP30PFLQ1J.ad.testnet.local</Computer>
  <Security />
</System>
- <EventData>
  <Data Name="SubjectUserSid">S-1-5-18</Data>
  <Data Name="SubjectUserName">SYSTEM</Data>
  <Data Name="SubjectDomainName">NT AUTHORITY</Data>
  <Data Name="SubjectLogonId">0x3e7</Data>
  <Data Name="PrivilegeList">SeAssignPrimaryTokenPrivilege SeTcbPrivilege SeSecurityPrivilege
    SeTakeOwnershipPrivilege SeLoadDriverPrivilege SeBackupPrivilege SeRestorePrivilege SeDebugPrivilege
    SeAuditPrivilege SeSystemEnvironmentPrivilege SeImpersonatePrivilege</Data>
</EventData>
</Event>
```

Kuvio 6. Event Viewer XML esimerkki

Kaikki tapahtumat sisältävät kaksi osaa: System ja EventData. Ne sisältävät kyseistä tapahtumaa kuvaavat kentät, esimerkiksi käyttäjän kirjautumisesta kertova tapahtuma sisältää käyttäjänimen ilmaisevan SubjectUserName-kentän. Jokaista lokitiedostoa käsitellään sarjana näitä kenttiä. System-osa sisältää saman tyyppin tapahtumille yhteiset tiedot ja tapahtuman julkaisun yhteydessä kerättyjä järjestelmäparametreja. Sen tärkeimmät kentät ovat EventID ja Version, jotka yhdessä uniikisti määrittävät tapahtuman. Level määrittää viestin vakavuustason tai laajuuden. Level-kenttä arvot 1-5 ovat ennalta määriteltyjä, mutta viestin tuottaja voi myös käyttää omia arvoja lukuun 255 asti. Task-kenttä sisältää numeerisen arvon, joko tapahtuman tuottajan toiminnallisuuden tai sovelluksen alikomponentin. Opcode-arvo kuvaa tyypillisesti sovelluksen tiettyä toimintoa tai sen osaa. Keyword on 56-lippua sisältävä kenttä, jota käytetään samankaltaisten tapahtumien ryhmittelemiseen. Tapahtumat voivat sisältää useita lippuja ja näin kuulua useaan ryhmään. Muita System-kenttiä ovat muun muassa tapahtuman aikaleima, prosessi-id, thread-id ja tietokoneen nimi (hostname). EventData-osa sisältää tapahtuman tuottaneen sovelluksen asettamat tiedot. EventData-osaa on mahdollista laajentaa. (Menn 2006.)

7.3 Haut ja suodatus

Event log -järjestelmästä on mahdollista hakea viestejä XPath-hakukielen avulla. XPath on W3C-standardointijärjestön kehittämä hakukieli kenttien valitsemiseen XML-dokumenteista. Sen avulla lokiviestejä voidaan hakea minkä tahansa kentän, alikentän tai attribuutin perusteella.

Esimerkki yksinkertaisesta XPath lausekkeesta:

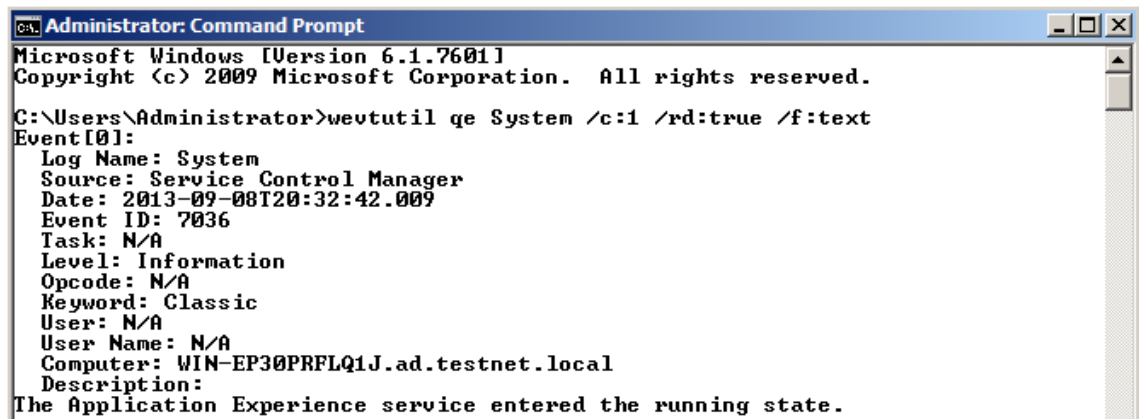
```
*/System[Provider/@Name='Microsoft-Windows-Security-Auditing' and Level <= 3]
```

Kyseinen lauseke valitsee kaikki kentät (tähti) viesteistä, joissa System-osan Provider-arvo on "Microsoft-Windows-Security-Auditing" ja joiden vakavuustaso on pienempi kuin 3.

Windows Event Viewer mahdollistaa käyttäjän määrittelemien näkymien luomisen, joihin voidaan sisällyttää viestit kenttien perusteella. Näkymään kuuluvat viestit voidaan määrittää helposti graafisen käyttöliittymän kautta (Filter), tai XPath-lausekkeen avulla (XML). Vastaavalla tavalla voidaan valmiiksi asetetuista kategorioista suodattaa vain halutut viestit. (Menn 2006.)

Event Viewer -työkalusta on olemassa myös komentorivipohjainen versio wevtutil. Wevtutil mahdollistaa lokiviestien tarkastelun Windows Server Core -palvelimilla, joissa ei ole lainkaan graafista käyttöliittymää. Sitä voidaan myös hyödyntää esimerkiksi skriptauksessa. (Tulloch 2010.)

Kuviossa 7 on esitetty wevtutil-työkalulla tehty haku.



```

Administrator: Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>wevtutil qe System /c:l /rd:true /f:text
Event[0]:
  Log Name: System
  Source: Service Control Manager
  Date: 2013-09-08T20:32:42.009
  Event ID: 7036
  Task: N/A
  Level: Information
  Opcode: N/A
  Keyword: Classic
  User: N/A
  User Name: N/A
  Computer: WIN-EP30PRFLQ1J.ad.testnet.local
  Description:
The Application Experience service entered the running state.

```

Kuvio 7. Wevtutil haku

8 Common Log Format ja Extended Log Format

Common Log Format (CLF) on W3C-standardointiorganisaation kehittämä lo-giformaatti, jota käytetään verkkopalvelinten lokiviesteissä esimerkiksi kirjaamaan lokiin asiakaslaitteiden tekemät pyynnöt palvelimelle. Alla on esimerkki CLF-tyypin lokiviestistä Apache-verkkopalvelimesta.

```
192.168.1.136 - - [13/Sep/2013:19:12:26 +0300] "GET /test.php HTTP/1.1" 200
61095
```

Kyseinen viesti sisältää asiakaslaitteen IP-osoitteen, aikaleiman, HTTP-pyyynnön, palvelimen lähettämän statuskoodin (200) ja asiakkaalle lähetettävän vastauksen koon (61095).

Extended Log Format (ELF) kehitettiin laajentamaan yleisesti verkkopalvelinten lokeissa käytettyä Common Log Format (CLF) -lokityyppiä, joka mahdollistaa vain tiettyjen tietojen tallentamisen tapahtumasta. ELF mahdollistaa kustomoitujen lo-kiviestien tuottamisen yleisten analysointisovellusten ymmärtämässä muodossa. (Hallam-Baker & Behlendorf N.d.)

Kuviossa 8 nähdään esimerkki ELF-lokiviestistä.

```

#Version: 1.0
#GMT-Offset: -0800
#Software: Oracle9iAS Web Cache/2.0.0.2.0
#Start-Date: 2001-10-31 00:00:18
#Fields: c-ip c-dns c-auth-id date time cs-method cs-uri sc-status
bytes cs(Cookie) cs(Referrer) time-taken cs(User-Agent)
#Date: 2001-10-31 00:00:18
64.103.37.2 client_joaz7 DMS.user 2001-10-31 00:00:18
GET /admin/images/oc_bottomleft.gif 200 350 "BIGipServerwww_
webcache_pool=1443321748.19460.0000;ORA_UCM_
AGID=%2fMP%2f8M7%3etSHPV%40%2fS%3f%3fDh3VHO"
"http://www.oracle.com/nl/partner/content.html" 370879 "Mozilla/4.5
[en] (WinNT; I)"

```

Kuvio 8. ELF-lokiviesti

9 Logstash

9.1 Yleistä

Logstash on lokienhallintaan tarkoitettu työkalu, jolla voidaan kerätä, parsia, lähettää ja tallentaa useista lähteistä vastaanotettuja lokiviestejä. Se on saatavilla ilmaiseksi Apache 2 -lisenssin alaisena. Logstash on kirjoitettu Jruby-ohjelmointikielellä ja paketoitu yhdeksi JAR-tyypin tiedostoksi. Tästä syystä Logstash vaatii toimiakseen ainoastaan Javan. Javan alustariippumattomuus mahdollistaa Logstashin toiminnan useilla eri käyttöjärjestelmillä. (Home 2013.)

9.2 Rakenne

Logstash-konfiguraatio koostuu kolmesta osasta: inputs, filters ja outputs. Näistä inputs ja outputs ovat pakollisia ja filters valinnainen. Kaikki kolme sisältävät lisäosia, joilla viestien käsittely tapahtuu. Lisäosat ovat modulaarisia eli niitä voi olla käytössä tarpeesta riippuen haluttu määrä yhtäaikaan. Lisäosien monipuolisuus ja määrä tekee Logstashista todellisen lokienhallinnan monitoimisosovelluksen. (Docs 2013.)

Inputs eli sisääntulot määrittävät mistä ja miten lokiviestejä vastaanotetaan. Logstash sisältää suuren määrän input-lisäosia, joista seuraavassa luettelossa on esitelty muutamia.

- Stdin. Lukee tapahtumia standardisyötteestä (standard input). Käytännössä logstash lukee kaiken konsoliin syötetyn tekstin.

- Syslog. Vastaanottaa Syslog-protokollan viestejä verkon yli. Syslog-input lisäosa tukee RFC 3164 mukaisia viestejä, jotka voidaan vastaanottaa UDP- tai TCP-protokollan avulla. Syslog on ehkä eniten käytetyin sisään-tulo, koska sillä voidaan vastaanottaa viestit laitteilta, jotka tukevat ainoastaan Syslog-protokollaa lähetykseen.
- Snmptrap. Lisäosa mahdollistaa lokitapahtumien vastaanottamisen SNMP-protokollan avulla.
- Eventlog. Kerää tapahtumia Windows Event log -järjestelmästä. Lisäosan avulla voidaan kerätä lokitapahtumat Windows-järjestelmästä ja lähettää eteenpäin keskitetylle palvelimelle.
- Rabbitmq. Lisäosan avulla voidaan vastaanottaa viestejä RabbitMQ-vaihteesta. Se mahdollistaa halutun jonon määrittämisen sekä autentikoinnin ja SSL-salauksen käytön.

(Docs 2013.)

Filters eli suodattimia käytetään viestien parsimiseen ja normalisointiin. Niillä voidaan erottaa halutut kentät viesteistä, poistaa turhat tai muokata niistä halutun muotoisiksi. Seuraavassa luettelossa on esitetty hyödyllisimmät suodattimet.

- Grok. Grok-suodattimen avulla voidaan muokata jäsentymättömistä loki-viesteistä kenttien avulla jäsenneityjä, joita on myöhemmin helppo indeksoida ja hakea. Logstash sisältää suuren määrän malleja, joita voidaan käyttää grok-suodattimen kanssa viestien parsimiseen. Malleja on myös helppo luoda itse lisää.
- Mutate. Mutate-suodattimen avulla voidaan muokata viestien kenttiä, kuten uudelleennimetä, poistaa, korvata ja muuttaa.
- Date. Date-suodatinta käytetään viestien aikaleiman parsimiseen.
- Syslog_pri. Suodatin mahdollistaa Syslog-viestien PRI-kentän parsimisen erillisiin facility- ja priority-kenttiin.
- Clone. Suodattimen avulla tietyt viestit voidaan monistaa ja suorittaa niille esimerkiksi erilainen parsinta.

(Docs 2013.)

Outputs eli ulostulot määrittävät miten tapahtumat tallennetaan tai lähetetään eteenpäin. Ulostulot sisältävät monia samoja lisäosia kuin sisääntulot, esimerkiksi syslog ja rabbitmq. Seuraavassa luettelossa on esitetty muutama ulostulo.

- File. Tallentaa tapahtumat tiedostojärjestelmään. Viestit voidaan pakata automaattisesti Gzip-pakkauksella ja viestin kenttiä voidaan käyttää nimeämään tallennettava tiedosto.
- Elasticsearch. Tallentaa tapahtumat Elasticsearch-tietokantaan, joka on tarkoitettu dokumenttien nopeaan reaaliaikaiseen hakuun.
- Email. Ulostulo mahdollistaa sähköpostin lähettämisen tapahtumista. Sähköpostiviestiin voidaan lisätä myös liitteitä.
- XMPP. Ulostulon avulla tapahtumat voidaan lähettää XMPP-palvelimelle haluttuun huoneeseen.

(Docs 2013.)

Edellämainittujen ulostulojen lisäksi Logstash tukee tapahtumien viemistä useisiin graafeja tuottaviin sovelluksiin, kuten Graphite, Ganglia ja Opentsdb. Tapahtumat voidaan myös ohjata verkonvalvotasovelluksiin, kuten Nagios ja Zabbix. (Docs 2013.)

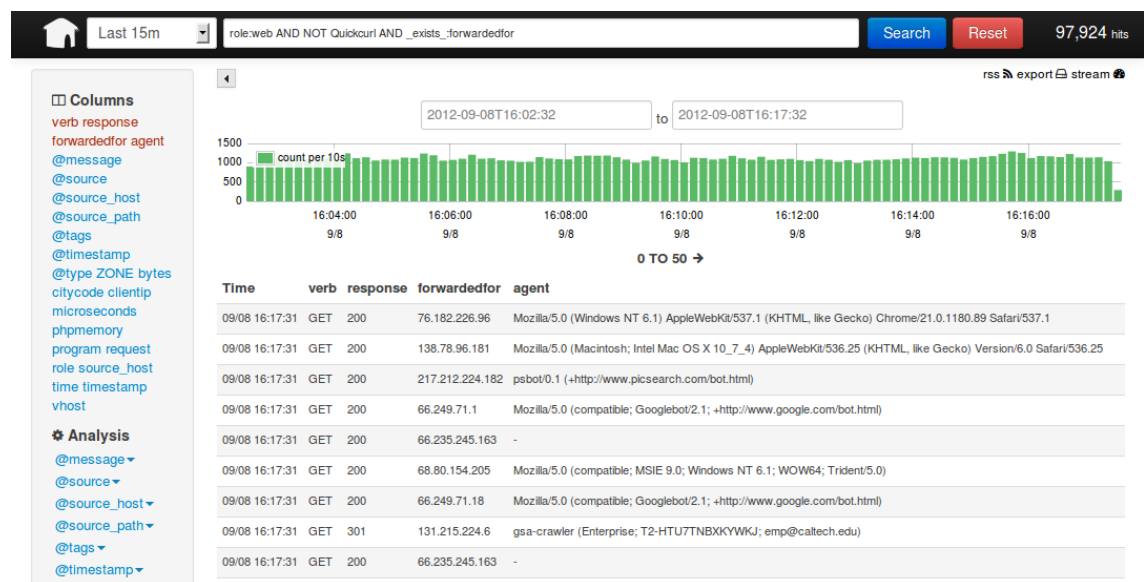
9.3 Web-käyttöliittymä

Logstash sisältää sisäänrakennetun Kibana-nimisen verkkokäyttöliittymän. Kibana hyödyntää Elasticsearch-tietokantaa viestien hakemiseen. Itse sovellus on kirjoitettu Ruby-ohjelmointikielellä ja hyödyntää Sinatra-webohjelmistokehystä. (Home 2013.)

Kibanan avulla lokiviestejä voidaan hakea kenttien nimen tai sisällön perusteella ja tuottaa tuloksena saatujen viestien määrästä graafeja. Haku on mahdollista rajata tietylle aikavälille tai seurata viestejä reaaliajassa. Kenttien sisällöstä voidaan analysoida erilaisia trendejä, kuten laskea yleisimpien arvojen lukumäärä ja prosentti kokonaismäärästä. Tätä ominaisuutta voidaan käyttää esimerkiksi hakemaan IP-osoitteet, joista vastaanotetaan eniten tapahtumia. Kibana osaa myös näyttää tietyllä aikavälillä tapahtuneet muutokset viestien määrässä. (About 2013a.)

Kibanan käyttöönotto on melko helppoa. Ensin tulee varmistaa, että Elasticsearch on asennettu ja toiminnassa. Seuraavaksi Logstash asetetaan ohjaamaan viestit Elasticsearch-tietokantaan lisäosan avulla. Lopuksi Logstash-prosessi käynnistetään web-option kanssa, jolloin myös Kibana-prosessi käynnistyy. Tämän jälkeen verkkokäyttöliittymään päästään avaamalla selain osoitteeseen <http://localhost:9292>, jossa 9292 on Kibanan käyttämä portti.

Kuviossa 9 nähdään kuva Kibanan käyttöliittymästä.



Kuvio 9. Kibanan käyttöliittymä

10 Elasticsearch

10.1 Yleistä

Elasticsearch on Shay Banonin vuonna 2010 kehittämä avoimen lähdekoodin hakupalvelin. Käytännössä se on tietokantapalvelin, joka tallentaa datan hakuoperaatioihin optimoituun muotoon. Elasticsearch soveltuu erityisen hyvin käyttötarkoituksiin, joissa suureen datamäärään halutaan tehdä nopeita hakuja.

Elasticsearch on kirjoitettu Java-ohelmointikielellä ja perustuu Apache Lucene -sovellukseen. Elasticsearch käyttää samoja algoritmeja tiedon hakuun ja tallennukseen, mutta tarjoaa paremman ohjelmointirajapinnan ja skaalautuvuuden. Lisäksi se mahdollistaa tiedon jatkuvan saatavuuden klusteroinnin ja replikaation avulla. Klusteroinnilla suuri määrä tietoa voidaan hajauttaa usealla palvelimelle ja tasata kuorma niiden välille. (Cholakian 2013.)

Elasticsearch on suunniteltu palauttamaan likimääräinen vastaus hakuihin. Tämä tarkoittaa, että tuloksen ei tarvitse täsmätä täydellisesti hakulausekkeeseen. Tätä ominaisuutta voidaan käyttää hakemaan lokiviesteistä tietty arvo riippumatta sen muotoilusta. Elasticsearch soveltuu myös tarkkoihin hakuihin. (Cholakian 2013.)

Elasticsearch-tietokantaa ei ole suunniteltu korvaamaan relaatiotietokantoja. Sillä ei esimerkiksi voi suorittaa matemaattisia laskutoimituksia datalle. Elasticsearch ei myöskään takaa tiedon uniikkiutta tietueiden välillä. (Cholakian 2013.)

10.2 Rakenne

Pienin datayksikkö Elasticsearch-tietokannassa on kenttä. Kenttälle on määritetty base unit of storage ja se sisältää yhden tai usemman kyseisen tyyppin arvon, esimerkiksi numeron tai merkkijonon. Dokumentit muodostavat tietokannan perustalennusyksikön sisältäen kokoelman kenttiä. Dokumentteja voidaan verrata relaatiotietokantojen riveihin. (Cholakian 2013.)

Dokumentit sisältävät käyttäjän asettaman tyyppikartoituksen, joka vastaa relaatiotietokantojen skeemoja. Tyyppikartoitus määrittää dokumentin kenttien tyytit ja kuinka ne indeksoidaan. (Cholakian 2013.)

Elasticsearch-tietokannan perustyytit on esitetty taulukossa 3.

Taulukko 3. Elasticsearch perustyytit

Tyyppi	Määritelmä
string	teksti
integer	32 bittinen kokonaisluku
long	64 bittinen kokonaisluku
float	liukuluku
double	64-bittinen liukuluku
boolean	totuusarvo
date	UTC päivämäärä/aika
null	null-arvo

Elasticsearch-tietokanta käyttää tiedon muotoiluun pääasiassa JSON-formaattia. JSON on ohjelmointikieli riippumaton tiedonsiirtomuoto, joka on suunniteltu helposti tulkittavaksi ihmisille sekä helposti tuotettavaksi ja parsittavaksi tietokoneelle. Se koostuu kokoelmasta nimi/arvo-pareja ja järjestetystä listasta arvoja. (Cholakian 2013.)

Kuviossa 10 Nähdään esimerkki JSON-muotoilusta.

```
{
  "Hostname" : "homepc.local",
  "Operating_system" : "Windows_7_x64",
  "Software" : [ Microsoft_Office, Mozilla_Firefox, Mozilla_Thunderbird ],
  "Hardware" : { "CPU" : "Intel_i5", "Memory" : "16G_DDR3" "Mainboard" : "Gigabyte_Z77" }
}
```

Kuvio 10. JSON esimerkki

Elasticsearch varaa osan kentistä omaan käyttöön, esimerkiksi `_id` yksilöi dokumentin. Id-arvo vastaa relaattietokantojen perusavainta. Elasticsearch muuntaa JSON-muotoilun Lucene-ohjelmointirajapinnan yksinkertaisiksi avain/arvo-pareiksi. (Cholakian 2013.)

Elasticsearch-tietokannan suurin datayksikkö on indeksi. Indeksit ovat itsenäisiä dokumentit sisältäviä osioita Elasticsearch-palvelimella. Dokumentit ja dokumentityypit ovat uniikkeja niiden sisällä. Indeksit vastaavat relaattietokantojen yksittäisiä tietokantoja. (Cholakian 2013.)

Monet Elasticsearch-tietokannan saatavuuteen liittyvistä asetuksista, kuten klusterointi ja replikaatio tapahtuvat indeksitasolla. Elasticsearch mahdollistaa tiedonhaun yhdestä tai useammasta indeksistä. Relaatiotietokannoista poiketen, index-tietorakenteisiin tallennetut arvot ovat optimoitu nopeisiin ja tehokkaisiin hakuihin. (Cholakian 2013.)

10.3 Datan tallennus ja haku

Kommunikointi Elasticsearch-tietokannan kanssa perustuu HTTP-protokollaan ja REST-ohjelmistorajapintaan. REST-rajapinnan avulla voidaan tallentaa ja hakea dataa, hallita indeksejä sekä muuttaa tietokannan parametreja. Rajapintaa voidaan käyttää suoraan verkkoselaimesta tai komentoriviltä Curl-työkalulla.

Objekteihin viitataan tiedostojärjestelmistä tutulla polulla, esimerkiksi `"/users/1"` viittaa käyttäjäindeksin käyttäjään, jonka id-arvo on yksi. REST-ohjelmistorajapinta tukee HTTP-protokollan komentoja, kuten GET, POST ja PUT. Näillä komennoilla suoritetaan Elasticsearch-tietokantaan kohdistuvat operaatiot, esimerkiksi GET hakee tietoa ja POST sekä PUT tallentavat.

Kuvion 11 esimerkissä GET-komennolla noudetaan Elasticsearch-klusterin palvelinten tiedot.

```
[root@CentOS ~]# curl -XGET http://localhost:9200/_cluster/nodes?pretty=true
{
  "ok" : true,
  "cluster_name" : "elasticsearch",
  "nodes" : {
    "NBiSLXGOR6uedWcKk0-_Xw" : {
      "name" : "Belasco",
      "transport_address" : "inet[/192.168.1.125:9300]",
      "version" : "0.90.5",
      "http_address" : "inet[/192.168.1.125:9200]"
    }
  }
}
```

Kuvio 11. Elasticsearch GET esimerkki

Tietokantaan voidaan lisätä dataa POST-komennolla. Curl-työkalua käytettäessä, lisättävä data tulee kirjoittaa `d`-option jälkeen ja sen tulee olla JSON-muotoiltua.

```
[root@CentOS ~]# curl -XPOST 'http://localhost:9200/users/admins' -d '{
"name" : "jani",
"city" : "jyvaskyla",
"school" : "jamk",
"status" : "student"
}'
{"ok":true,"_index":"users","_type":"admins","_id":"p0esvxA4S-uvRUtINXL52w","_version":1}
```

Kuvio 12. Elasticsearch POST esimerkki

Kuvion 12 esimerkissä tietokantaan luotiin indeksi "users", jonka tyyppi on "admins". Kuvion alimmalla rivillä nähdään palvelimen palauttama vastaus, joka kertoo operaation statuksen ja mihin dokumentti luotiin. Elasticsearch lisäsi dokumentille automaattisesti uniikin tunnisteen (_id).

Tietokannasta hakuihin käytetään "_search"-optiota. Tällöin haettava kenttä ja arvo määritetään querystring-parametriin. Kuvion 13 esimerkissä haetaan kaikki dokumentit, joissa nimi-kentän arvo on "jani".

```
[root@CentOS ~]# curl -XGET 'http://localhost:9200/users/_search?q=name:jani&pretty=true'
{
  "took" : 3,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "failed" : 0
  },
  "hits" : {
    "total" : 1,
    "max_score" : 0.30685282,
    "hits" : [ {
      "_index" : "users",
      "_type" : "admins",
      "_id" : "p0esvxA4S-uvRUtINXL52w",
      "_score" : 0.30685282, "_source" : {
        "name" : "jani",
        "city" : "jyvaskyla",
        "school" : "jamk",
        "status" : "student"
      }
    } ]
  }
}
```

Kuvio 13. Elasticsearch haku

10.4 Klusterointi ja replikaatio

Elasticsearch voi toimia itsenäisenä palvelimena tai useasta palvelimesta koostuvana klusterina. Klusterin mahdollistaa suurten datamäärien käsittelyn hajauttamisen ja vikasietoisuuden. Elasticsearch indeksien data voidaan hajauttaa sirpaleisiin (shard), joista jokainen on erillinen Apache Lucene -indeksi. Sirpaleet jaetaan tämän jälkeen klusterin palvelimille. Käyttäjän hakiessa tietoa useasta sirpaleesta koostuvasta indeksistä, Elasticsearch lähettää haun eri sirpaleille ja kokoaa tuloksen yhteen ennen käyttäjälle palauttamista. (Kuč & Rogoziński 2013.)

Elasticsearch mahdollistaa replikoinnin, jossa data peilataan useille palvelimille. Dataa replikoimalla voidaan tehostaa hakuja sekä saavuttaa korkea vikasietoisuus. Replikaatio koostuu ensisijaisesta sirpaleesta ja halutusta määrästä replika-sirpaleita. Ensisijaiseen sirpaleeseen ohjataan kaikki indeksin datan muutokset. Replika-sirpaleet ovat täydellisiä kopioita ensisijaisesta sirpaleesta. Jos ensisijainen sirpale menetetään, replika-sirpaleista valitaan automaattisesti uusi ensisijainen sirpale. Elasticsearch osaa klusterin palvelimen vikaantuessa automaattisesti järjestellä datan uudelleen jäljellä oleville palvelimille. (Kuč & Rogoziński 2013.)

11 AMQP

11.1 Yleistä

AMQP (Advanced Message Queuing Protocol) on viestien välittämiseen suunniteltu verkkoprotokolla. Protokollan kehitti vuonna 2003 JPMorgan Chase pankissa työskennellyt Jon O'Hara. Nykyisin kehityksestä vastaa AMQP-työryhmä, johon kuuluu useita suuria pankkeja ja IT-alan yrityksiä. AMQP on avoin standardi, jonka pohjalta kuka tahansa voi kehittää oman toteutuksen. Yksi standardin tavoitteista on yhteensopivuuden varmistaminen eri toteutuksien välillä. Esimerkkejä eri toteutuksista ovat SwiftMQ, Apache ActiveMQ, StormMQ ja RabbitMQ. AMQP-asiakasohjelmia on saatavilla lähes kaikille yleisille ohjelmointikielille, kuten Java, C, Python, Perl ja Ruby.

11.2 Protokollan toiminta

AMQP-protokolla mahdollistaa asiakassovellusten kommunikoinnin viestikeskusten kanssa. Viestikeskuksesta (broker) voidaan käyttää myös nimitystä vaihde. Ne vastaanottavat viestit tuottajilta (producer), jotka ovat viestien lähteitä. Vaihde reitittää viestit niitä prosessoivalla asiakalle (consumer). AMQP on suunniteltu toimimaan verkon yli, joten tuottajat, vaihteet ja asiakkaat voivat sijaita eri järjestelmissä. (Klishin 2013.)

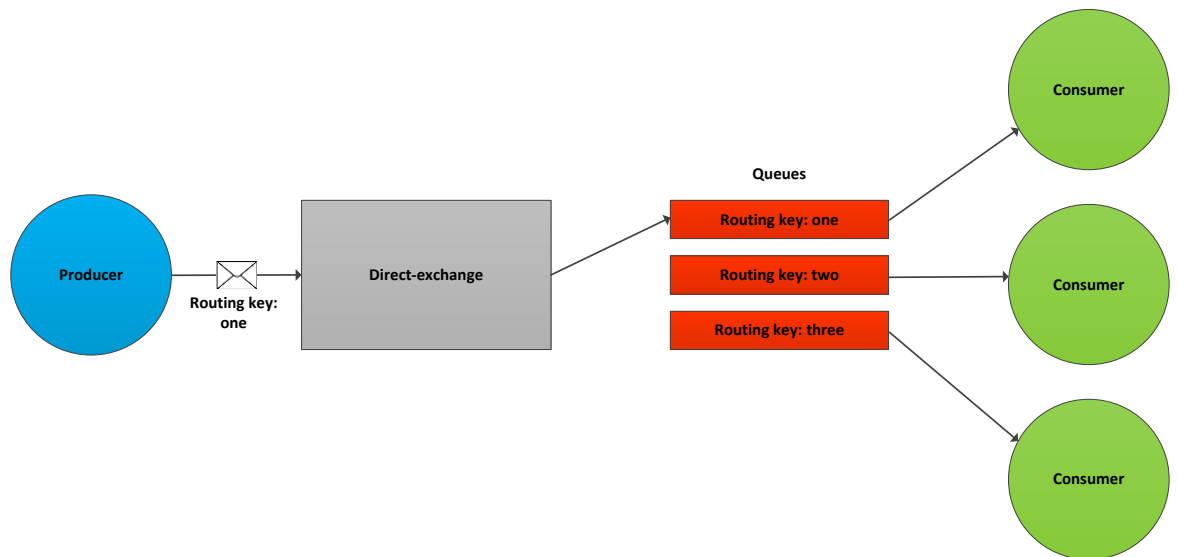
Tuottajat lähettävät viestit vaihteeseen, joka lisää sen jonoihin sidossääntöjen (bindings) perusteella. Vaihde lähettää viestit asiakkaille, jotka ovat rekisteröityneet jonoon. Vaihtoehtoisesti asiakkaat voivat itse pyytää viestejä jonosta tarvittaessa. Tuottajat voivat lisätä viesteihin attribuutteja, joista osa on tarkoitettu vaihteen käyttöön ja osa asiakkaan. AMQP-protokolla mahdollistaa viestien kuittaukset, joilla asiakas voi ilmoittaa vaihteelle vastaanottaneensa viestin. Kuittauksia käytettäessä vaihde poistaa viestin jonosta vasta saatuaan kuittauksen perillemenosta. Kuittauksilla pyritään takaamaan viestien luotettava kuljetus. (Klishin 2013.)

11.3 Vaihteet

AMQP-vaihteet toimivat protokollassa viestien välittäjinä tuottajien ja asiakkaiden välissä. Viestien reitittämiseen käytettävä algoritmi riippuu vaihteen tyypistä ja sidossäännöistä. Vaihteita on neljää eri tyyppiä: Direct, Fanout, Topic ja Headers. (Klishin)

Direct-vaihde välittää viestit jonoihin reititysavaimen perusteella ja soveltuu hyvin viestien unicast-tyyppiseen reititykseen. Direct-vaihteessa jonot sidotaan vaihteeseen reititysavaimen perusteella, jolloin viestin saapuessa tietyllä reititysavaimella, ohjataan se kyseiseen avaimen sidottuun jonoon. (Klishin 2013.)

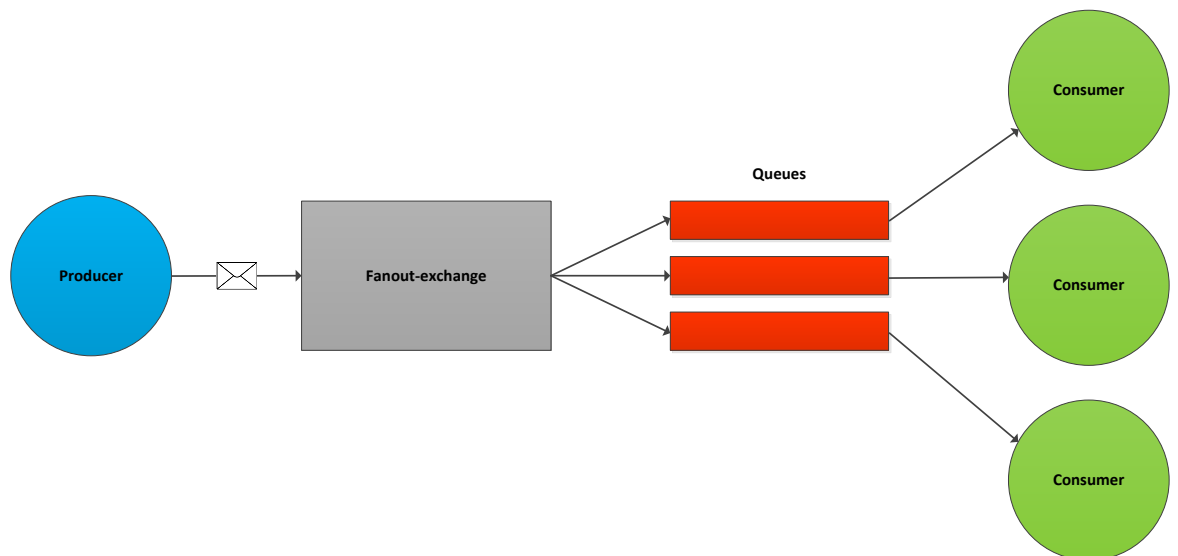
Kuviossa 14 on esitetty Direct-vaihteen toiminta.



Kuvio 14. Direct-exchange

Fanout-vaihde toimii broadcast-periaatteella, eli välittää viestit kaikkiin siihen sidottuihin jonoihin reititysavaimesta välittämättä. Fanout-vaihteeseen vastaanotettu viesti kopioidaan kaikkiin siihen sidottuihin jonoihin. Se soveltuu hyvin tilanteisiin, joissa tietty viesti halutaan lähettää kaikille osapuolille, kuten chat-tyyppisissä sovelluksissa. (Klishin 2013.)

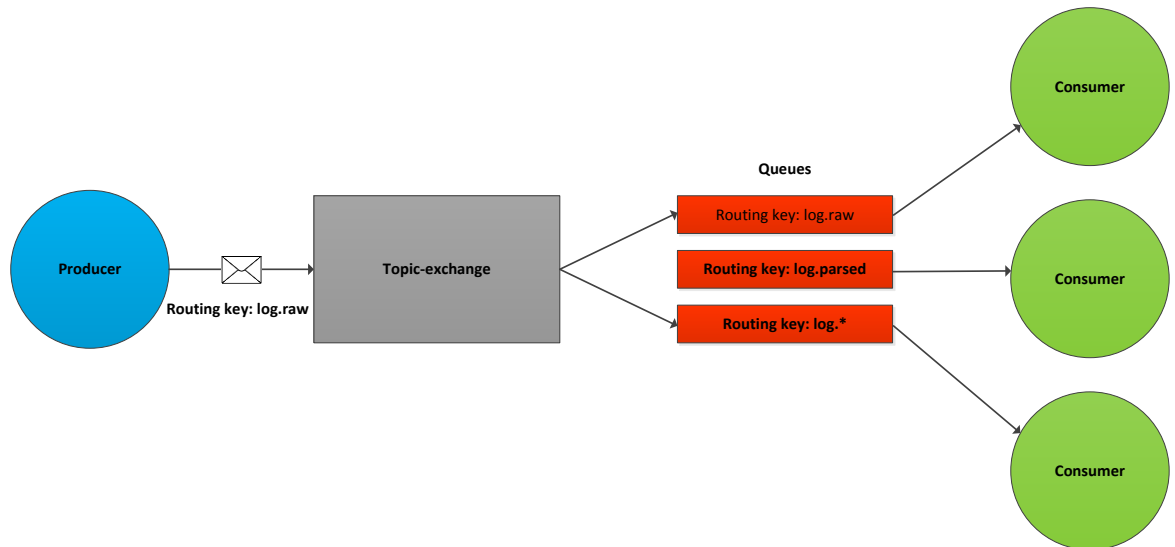
Kuviossa 15 on esitetty fanout-vaihteen toiminta.



Kuvio 15. Fanout-exchange

Topic-vaihte välittää viestit yhteen tai useampaan jonoon reititysavaimen ja jonon vaihteeseen sitomisessa käytetyn säännön perusteella. Se soveltuu viestien unicast-reitityksen lisäksi multicast-reititykseen. Topic-vaihte soveltuu parhaiten tilanteisiin, joissa viestejä lähetetään useille asiakkaille, jotka haluavat valita vastaanottamansa viestit. (Klishin 2013.)

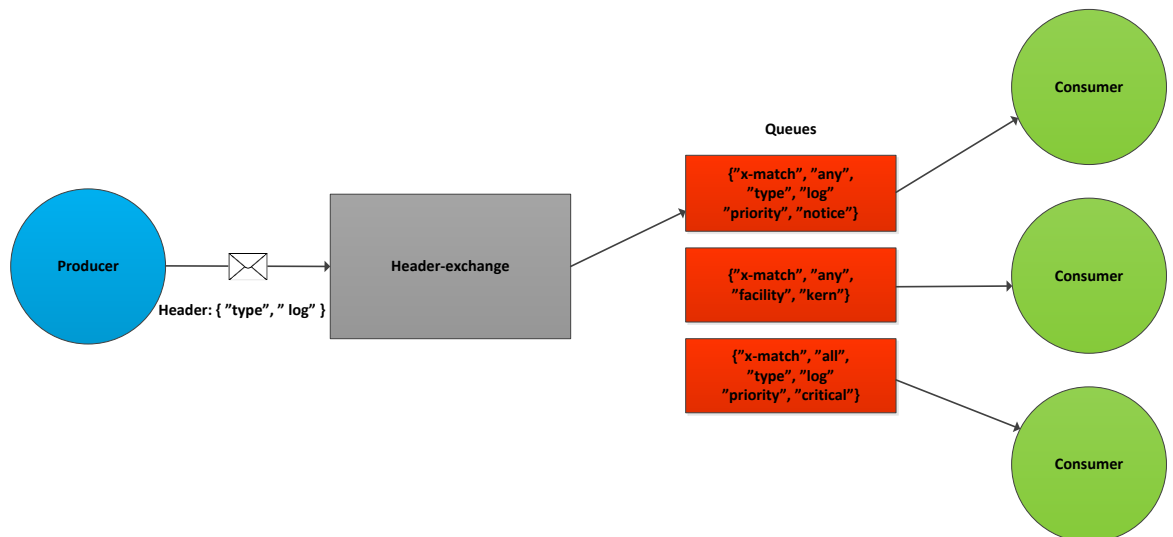
Kuviossa 16 on esitetty topic-vaihteen toiminta.



Kuvio 16. Topic-exchange

Headers-vaihte välittää viestit otsikoiden attribuuttien perusteella, eikä huomio reititysavainta. Otsikon attribuutin arvon tulee täsmätä jonon vaihteeseen sidonnassa käytettyyn arvoon. Vaihteeseen voidaan määrittää tuleeko kaikkien attribuuttien täsmätä vai ainoastaan tiettyjen. (Klishin 2013.)

Kuviossa 17 on esitetty headers-vaihteen toiminta.



Kuvio 17. Headers-exchange

AMQP-vaihde luo aina direct-tyyppisen oletusvaihteen, johon kaikki jonot liitetään niiden nimeä vastaavalla reititysavaimella. Tuottaja voi tällöin lähettää viestit haluamaansa jonoon asettamalla viestin reititysavaimeksi jonon nimen. (Klishin 2013.)

Tyyppin lisäksi vaihteet sisältävät monia muita attribuutteja, kuten nimen, kestävyysden, automaattipoiston ja vaihdetyypistä riippuvia attribuutteja. Kestävyydellä tarkoitetaan vaihteen kykyä selvittää uudelleenkäynnistyksestä ilman, että sitä tarvitsee luoda uudestaan. Automaattipoiston ollessa päällä vaihde poistuu automaattisesti, kun siihen ei ole sidottuna yhtään jonoa. (Klishin 2013.)

11.4 Jonot ja sidokset

AMQP-vaihteen vastaanottamat viestit siirretään jonoihin. Ne sisältävät samoja ominaisuuksia kuin vaihteet, esimerkiksi nimi, kestävyys ja automaattipoisto. Kestävät jonot tallennetaan kiintolevyllä, joten ne eivät häviä vaihteen uudelleenkäynnistyksen yhteydessä. (Klishin 2013.)

Jono tulee määrittää ennen kuin sitä voidaan käyttää. Vaihteeseen liittyvät sovellukset määrittävät siihen kuuluvat jonot. Sovellus voi asettaa jonon nimen itse tai pyytää vaihdetta generoimaan nimen. Vaihde luo lisäksi omia amqp-alkuisia jonoja sisäiseen käyttöön. (Klishin 2013.)

Sidokset ovat sääntöjä, joita vaihteet käyttävät viestien reitittämiseen eri jonoihin. Jonot liitetään sidosten avulla vaihteisiin. Osa vaihdetyypeistä käyttää reititysavainta sidoksissa määrittämään jonot, joihin viestit ohjataan. Sidokset mahdollistavat monipuolisten reitityssääntöjen luomisen eri viestityypeille. Jos viestiä ei voida reitittää mihinkään jonoon, se joko pudotetaan tai lähetetään takaisin tuottajalle. (Klishin 2013.)

11.5 AMQP-viestit

AMQP-viestit sisältävät useita attribuutteja, joista osa on protokollan määrittämiä. Viestiattribuutteja ovat esimerkiksi sisällön tyyppi ja koodaus, reititysavain, toimitustapa, prioriteetti ja aikaleima. Attribuuteista osa on tarkoitettu vaihteiden käyttöön, mutta suurin osa jää vastaanottavan sovelluksen tulkittavaksi. Viestien attribuutit asettaa tuottava sovellus. (Klishin 2013.)

Atribuuttien lisäksi viestit sisältävät hyötykuorman, joka on viestin sisältämä data. Data muutetaan usein serialisoituun muotoon, kuten JSON-formaattiin ennen sen lisäämistä hyötykuorman. Vaihteet eivät käsittele viestien hyötykuormaa. AMQP-viestit voidaan määrittää toimitustapa-attribuutin avulla kestäviksi, jolloin vaihteet tallentaa ne kiintolevyille. Tällöin kestävät viestit säilyvät vaihteen uudelleenkäynnistyksessä. Toisaalta tämä vaikuttaa vaihteen suorituskykyyn. (Klishin 2013.)

11.6 Yhteydet ja kanavat

AMQP on sovellustason protokolla, joka käyttää TCP-protokollaa viestien luotettavaan kuljetukseen. Yhteyden osapuolet voidaan todentaa ja yhteys voidaan salata käyttäen TLS-protokollaa. AMQP-yhteydet multipleksoidaan kanaviksi, joiden avulla monta yhteyttä vaihteeseen voidaan toteuttaa yhden TCP-yhteyden kautta. Tämä on hyödyllistä tapauksissa, joissa yksi sovellus tarvitsee usean yhteyden vaihteeseen. Kanavat ovat täysin eristettyjä toisistaan ja jokainen AMQP-metodi kuljettaa mukanaan kanavanumeroa, jolla asiakassovellus erottaa mihin kanavaan metodi kuuluu. (Klishin 2013.)

11.7 RabbitMQ

RabbitMQ on yksi yleisimmin käytetyistä AMQP-vaihdetoteutuksista. Sen kehitti alunperin Rabbit Technologies Ltd, mutta nykyisin kehityksestä vastaa VMwareen kuuluva SpringSource. RabbitMQ on kirjoitettu Erlang-ohjelmointikielellä ja julkaistaan avoimenlähdekoodin Mozilla Public license -ohjelmistolisenssin alla.

RabbitMQ mahdollistaa viestien luotettavan siirron kuittausten sekä kestävien vaihteiden ja jonojen avulla. Se sisältää monia laajennuksia AMQP-protokollaan, kuten kuittaukset viestin tuottajalle, vaihteiden välisen reitityksen sekä jono- ja viestikohtaiset TTL-arvot. RabbitMQ mahdollistaa vaihteiden klusteroinnin avulla korkean saatavuuden. Jonot voidaan peilata klusterin palvelinten välille, jolloin laiterikko ei aiheuta vaihteen viestien häviämistä. Klusterointia voidaan myös käyttää kuorman jakamiseen useiden vaihteiden kesken. (What can RabbitMQ do for you? 2013.)

RabbitMQ tukee useita liitännäisiä, joilla vaihteen ominaisuuksia voidaan laajentaa. Näitä ovat esimerkiksi vaihteen hallintaan tarkoitettut työkalut sekä ulkoisen autentikoinnin mahdollistavat liitännäiset. RabbitMQ etu on myös sen helppo asennus ja käyttöjärjestelmäriippumattomuus. Se on saatavilla lähes kaikille Unix-tyyppisille käyttöjärjestelmille sekä Microsoft Windows ja Apple Mac OS X -alustoille. (Plugins 2013.)

12 Graylog2

12.1 Yleistä

Graylog2 on avoimeen lähdekoodiin (GPLv3) perustuva lokienhallintajärjestelmä. Graylog2 koostuu Java-pohjaisesta palvelimesta ja verkkokäyttöliittymästä. Palvelin vastaanottaa viestit ja tallentaa ne Elasticsearch-tietokantaan. Viestit voidaan vastaanottaa verkosta UDP-, TCP- tai AMQP-protokollan avulla. Lokiformaattina voi toimia Syslog tai GELF (Graylog Extended Log Format). Palvelin pystyy tekijöiden mukaan käsittelemään tuhansia viestejä sekunnissa ja säilömään dataa useiden teratavujen edestä. Sovellusta voidaan myös laajentaa kolmansien osapuolten liitännäisillä. Graylog2-lokijärjestelmälle on nykyisin saatavilla kaupallisia tuki- ja ylläpitopalveluja. (About 2013b.)

12.2 Käyttöliittymä ja ominaisuudet

Graylog2-palvelimen tallentamia viestejä voidaan tarkastella verkkokäyttöliittymän avulla, joka toimii Ruby on Rails -verkko-ohjelmistokehyksen päällä. Verkkokäyttöliittymä messages-sivu esittää uusimmat vastaanotetut viestit järjestyksessä. Viestin valittaessa avautuu oikealle valikko, josta nähdään selkeästi sen sisältämät kentät. Kentän arvoa painamalla sovellus näyttää kaikki viestit, jotka sisältävät saman arvon. Messages-sivun hakukentän avulla voidaan suorittaa hakuja Elasticsearch-tietokantaan. Viestejä voidaan hakea vapailla hakusanoilla sekä tiettyjen kenttien arvojen perusteella. Hakuja voidaan ketjuttaa loogisten operaattorien avulla. Quickfilter-suodatin mahdollistaa tarkkojen, kenttien arvoihin perustuvien hakujen teon. (About 2013b.)

Streams-sivulla viestit voidaan jakaa virtoihin, jotka ovat käytännössä tallennettuja hakuja. Virroille määritetään säännöt, joiden perusteella viestit jaetaan. Säännöissä voidaan määrittää viestit täsmäämään tietyn kentän arvon perusteella, esimerkiksi kaikki viestit tietyistä osoitteesta tai tietyllä vakavuusasteella. Virtoihin voidaan myös asettaa viestien määrälle tietyt raja-arvot, joiden ylittyessä lähetetään automaattinen hälytys ylläpitäjille. (About 2013b.)

Graylog2 tallentaa viestit Elasticsearch-tietokannan lisäksi MongoDB-tietokantaan graafeja ja statistiikkoja varten. Verkkokäyttöliittymästä pystytään tarkastelemaan graafia viestien kokonaismäärästä tai vain tiettyyn hakuun täsmäävien viestien määrästä. Graafeista voidaan valita aikaväli, jonka aikana vastaanotetut viestit näytetään.

12.3 GELF

GELF on Graylog2-lokijärjestelmän tekijöiden kehittämä lokiformaatti. Sen tarkoituksena on korjata Syslog-protokollan heikkoudet, joita ovat 1024-tavun pituusraja sekä jäsentymättömyys. GELF-lokiformaatista löytyy ohjelmointikirjasta monille kielille, kuten Python, Perl, PHP, Ruby ja Java.

GELF-viesti itsessään on GZIP- tai ZLIB-pakattu JSON-muotoiltu merkkijono. Graylog2 tukee ns. chunked GELF-muotoa, jossa pitkät viestit pilkotaan pienemmiksi paloiksi ja vastaanottaja kokoaa ne yhteen. Tällöin viestien otsikon tulee sisältää GELF ID-, Message ID-, järjestetysnumero- ja osien määrä -kentät. (Graylog Extended Log Format 2013.)

Kuviossa 18 nähdään esimerkki GELF-viestistä.

```
{
  "version": "1.0",
  "host": "www1",
  "short_message": "Short message",
  "full_message": "Backtrace here\n\nmore stuff",
  "timestamp": 1291899928.412,
  "level": 1,
  "facility": "payment-backend",
  "file": "/var/www/somefile.rb",
  "line": 356,
  "_user_id": 42,
  "_something_else": "foo"
}
```

Kuvio 18. GELF-viesti

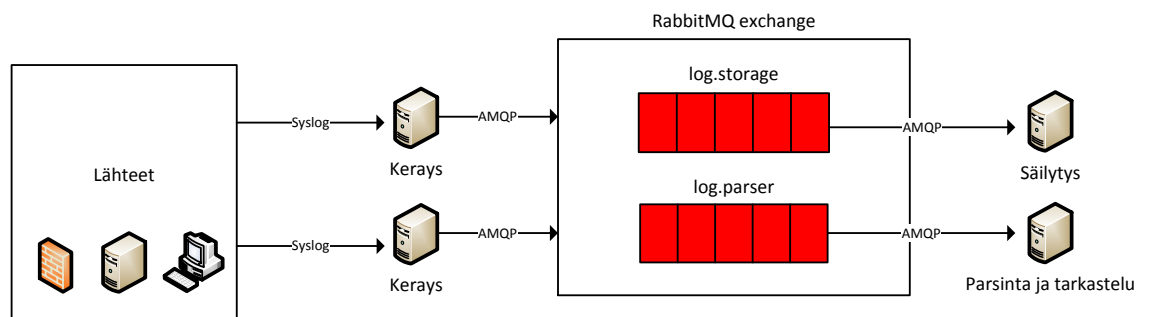
13 Käytännön toteutus

13.1 Lokienhallintajärjestelmän rakenne

Toteutettava lokijärjestelmä koostuu viidestä osasta: lokiviestien lähteet, keräyspalvelimet, AMQP-vaihde, parsinta- ja tarkastelupalvelin sekä säilytyspalvelin.

Lokiviestien lähteet lähettävät viestit Syslog-protokollan avulla keräyspalvelimille, jotka lähettävät ne edelleen AMQP-vaihteeseen log.raw-reititysavaimella. Parsinta- ja tarkastelupalvelin vastaanottaa viestit vaihteesta, suorittaa niille parsinnan ja normalisoinnin, jonka jälkeen lähettää ne Graylog2-sovellukselle. Graylog2 mahdollistaa viestien tarkastelujen ja haun. Säilytyspalvelin vastaanottaa muokkamattomat viestit vaihteesta ja varastoi ne pitkäaikaissäilytystä varten.

Kuviossa 19 on kuvattu järjestelmän toiminta.



Kuvio 19. Lokijärjestelmän toiminta

Lokienhallintajärjestelmä toteutettiin virtuaalisena VMware ESXi -ympäristöön. Toteutus virtuaaliympäristöön mahdollistaa hyvän skaalautuvuuden, koska virtuaalikoneisiin voidaan lisätä helposti resursseja ja uusia koneita voidaan ottaa käyttöön nopeasti valmiista pohjista kloonamalla. Virtuaalikoneiden hallinta onnistuu helposti yhden käyttöliittymän kautta. Niiden varmuuskopiointi on myös helpompaa kuin erillisten fyysisten koneiden.

Järjestelmän palvelimien käyttöjärjestelmäksi valittiin CentOS Linux uusim versio 6.4. Se on ilmainen avoimen lähdekoodin käyttöjärjestelmä, joka perustuu Red Hat Enterprise Linux -käyttöjärjestelmään. CentOS valittiin sen luotettavuuden ja laajan tuen perusteella. Lisäksi CentOS-koneita on paljon käytössä LabraNet-ympäristössä, joten sen hallinta on ylläpidolle ennestään tuttua.

Palvelimet liitettiin hallintaliikenteelle tarkoitettuun verkkoon, jonka kautta ne pysyvät kommunikoimaan toistensa ja lokilähteiden kanssa. Palvelimille asetettiin taulukon 4 mukaiset IP-osoitteet ja nimet:

Taulukko 4. Lokipalvelinten tiedot

Virtuaalikone	Hostname	IP-osoite	Aliverkon peite	Yhdyskäytävä
Collector1	collector1	192.168.x.x	255.255.255.0	192.168.x.x
Collector2	collector2	192.168.x.x	255.255.255.0	192.168.x.x
Log_storage	log-storage	192.168.x.x	255.255.255.0	192.168.x.x
AMQP-broker	amqp	192.168.x.x	255.255.255.0	192.168.x.x
Graylog2	graylog	192.168.x.x	255.255.255.0	192.168.x.x

13.2 Lokiviestien lähteet

Lokiviestien lähteillä tarkoitetaan laitteita, joilta viestit kerätään. Niitä ovat muun muassa palvelimet, työasemat, kytkimet, reitittimet ja palomuurit. Laitteiden sisällä lähteet voidaan jakaa käyttöjärjestelmiin ja erillisiin sovelluksiin.

Lokiviestien lähetykseen keräyspalvelimille hyödynnettiin Syslog-protokollaa. Syynä protokollan valintaan oli sen de-facto -status sekä laaja tuki eri valmistajien laitteissa ja sovelluksissa. Syslog-protokollaa tukemattomissa järjestelmissä, kuten Windows-palvelimilla, käytettiin erillistä sovellusta viestien lähetykseen keräyspalvelimelle.

13.3 Keräyspalvelimet

13.3.1 Kuvaus ja resurssit

Keräyspalvelimet vastaanottavat viestit laitteilta ja lähettävät ne AMQP-protokollalla vaihteeseen. Keräyspalvelimia on käytössä kuormantasaamiseksi kaksi kappaletta, joista ensimmäinen huolehtii verkkolaitteiden viesteistä ja toinen palvelinten. Keräyspalvelimia voidaan helposti lisätä uusia viestien määrän lisääntyessä.

Virtuaaliympäristöön luotiin Collector1- ja Collector2-nimiset virtuaalikoneet, joille asetettiin seuraavat laitteistoresurssit:

- CPU: 1 kpl
- muisti: 2 GB
- tallennustila: 16 GB.

Keräyspalvelimiin asennettiin Logstash-sovellus, joka huolehtii viestien vastaanotamisesta ja välityksestä vaihteeseen.

13.3.2 Logstash-asennus

Logstash-sovellus asennettiin /opt/logstash-kansion alle. Uusin versio (1.2.1) ladattiin osoitteesta <http://logstash.net/>.

```
wget https://logstash.objects.dreamhost.com/release/logstash-1.2.1-flatjar.jar
```

Logstash vaatii toimiakseen Javan, josta asennettiin avoimen lähdekoodin open-JDK-versio.

```
yum install java-1.7.0-openjdk
```

Jotta Logstash-sovellusta olisi mahdollista ajaa järjestelmän taustapalveluna, sille tarvitaan init-skripti. Logstash cookbook tarjoaa valmista skriptiä Red Hat - pohjaisille järjestelmille. Kyseinen skripti ladattiin /etc/init.d/-kansioon.

```
wget http://cookbook.logstash.net/recipes/using-init/logstash.sh -P /etc/init.d/
```

Tiedosto nimettiin uudelleen ja sille asetettiin suoritusoikeus.

```
mv /etc/init.d/logstash.sh /etc/init.d/logstash  
chmod +x /etc/init.d/logstash
```

Init-skriptissä määritetään muun muassa konfiguraatio-, Java-binääri- ja lokitiedoston sijainti. Binääri-tiedoston nimeksi asetettiin JARNAME-parametrissa aiemmin ladatun JAR-tiedoston nimi ja muut parametrit jätettiin oletusarvoiksi.

```
JARNAME= logstash-1.2.1-flatjar.jar
```

13.3.3 Logstash-asetukset

Logstash-sovelluksen asetustiedosto "logstash.conf" luotiin /opt/logstash-hakemiston alle. Logstash-konfiguraatio koostuu kolmesta osasta: input, filter ja output. Input-osioon määritetään sisääntulot, filter-osioon viestien suodatus ja output-osioon ulostulot. Ainoastaan Input- ja output-osiot ovat pakollisia. Eri osioiden alle määritetään liitännäiset, jotka suorittavat viestien käsittelyn.

Eri osiot erotellaan toisistaan ohjelmointikielistä tutuilla kaarisulkeilla. Osioiden alle asetetaan halutut liitännäiset ja niiden parametrit. Parametrit ovat yksinkertaisia avain/arvo pareja. Arvoja on olemassa usean tyyppisiä, esimerkiksi merkkijono, numero, totuusarvo ja taulukko. Liitännäisten parametrit on dokumentoitu sovelluksen verkkosivuilla.

Kuviossa 20 on esitetty konfiguraation rakenne.

```

input {
  stdin {
  }
}

filter {
  mutate {
    add_tag => [ "test" ]
  }
}

output {
  file {
    path => "/var/log/logstash"
  }
}

```

Kuvio 20. Logstash konfiguraatioesimerkki

Kuvion konfiguraatiossa viestit vastaanotetaan standardisyötteestä (stdin), jonka jälkeen mutate-filter lisää viesteihin tagin "test". Tägeja voidaan käyttää viestien merkkaimiseen myöhempää käsittelyä varten sekä helpottamaan hakuja. Lopuksi file-ulostulo kirjoittaa viestit /var/log/logstash-tiedostoon. Jos filter-osio sisältää useita liitännäisiä, suoritetaan ne järjestyksessä ylhäältä alas.

Collector1-palvelimelle määritettiin udp-sisääntulo, jolla Syslog-viestit vastaanotetaan. Udp-sisääntulon ainoa pakollinen parametri on sovelluksen kuunteleva portti, jonka arvoksi asetettiin virallinen IANA-järjestön määrittämä Syslog-portti 514. Muita valinnaisia parametreja ovat esimerkiksi kuunneltava IP-osoite ja puskurin koko.

```

input {
  udp {
    port => 514
  }
}

```


Ulostuloksi asetettiin rabbitmq, jolla viestit välitetään vaihteeseen. Rabbitmq-ulostulo sisältää useita parametreja, joista vaihteen osoite, nimi ja tyyppi ovat pakollisia. Vaihteen osoitteeksi asetettiin 192.168.x.x, nimeksi "Labranet-loki" ja tyyppi "topic". Valinnaisista parametreista reititysavaimen arvoksi asetettiin "log.raw". Avaimen avulla vaihde osaa ohjata viestit jonoihin. "Durable"-parametri määrittää vaihteen kestäväksi ja "persistent"-parametri ohjeistaa vaihdetta säilöämään viestit levyille.

```
output {
  rabbitmq {
    host => "192.168.x.x"
    exchange => "Labranet-loki"
    exchange_type => "topic"
    key => "log.raw"
    durable => true
    persistent => true
  }
}
```

Collector2-palvelimelle asetettiin vastaavat asetukset, mutta sisääntuloihin lisättiin tuki GELF-viesteille. GELF-viestit vastaanotetaan Windows-palvelimilta.

```
input {
  udp {
    port => 514
  }
  gelf {
    port => 12201
  }
}
```

13.4 AMQP-vaihde

13.5 Kuvaus ja resurssit

AMQP-vaihteen tehtävä on toimia välikätenä keräyspalvelinten ja viestien käsitteijöiden välissä. Se vastaanottaa viestit keräyspalvelimilta ja asettaa jonoihin reititysavaimen perusteella. AMQP-protokolla mahdollistaa luotettavan viestien kuljetuksen sekä korkean saatavuuden. Lisäksi AMQP-vaihde mahdollistaa lokijärjestelmän skaalautuvuuden, koska siihen voidaan helposti liittää haluttu määrä viestien lähteitä ja kuluttajia.

Vaihdetta varten luotiin "AMQP-broker"-niminen virtuaalikone seuraavilla resursseilla:

- CPU: 1 kpl
- muisti: 4 GB
- tallennustila: 25 GB.

Palvelimelle asennettiin RabbitMQ-sovellus. Ensin asennettiin sovelluksen riippuvuus, Erlang-ohjelmointikieli.

```
yum install erlang
```

RabbitMQ asennuspaketti ladattiin ja asennettiin sovelluksen virallisilta sivuilta:

```
wget http://www.rabbitmq.com/releases/rabbitmq-server/v3.1.5/rabbitmq-server-3.1.5-1.noarch.rpm  
rpm -ivh rabbitmq-server-3.1.5-1.noarch.rpm
```

Seuraavaksi asennettiin rabbitmq-management lisäosa, jolla vaihdetta voidaan hallita verkkokäyttöliittymän ja komentorivityökalun avulla. Lisäosa asennettiin komennolla:

```
rabbitmq-plugins enable rabbitmq_management
```

Vaihte käynnistettiin ja palvelu otettiin käyttöön komennoilla:

```
service rabbitmq-server start  
chkconfig rabbitmq-server on
```

RabbitMQ-vaihteen verkkokäyttöliittymään päästään menemällä selaimella osoitteeseen <http://<Palvelimen-IP>:15672>. Oletuskäyttäjätunnuksena/salasanana toimii guest/guest.

Kuviossa 21 nähdään RabbitMQ-verkkokäyttöliittymän pääsivu.



Overview

▼ Totals

Queued messages (chart: last minute) (?)

Currently idle

Message rates (chart: last minute) (?)

Currently idle

Global counts (?)

Connections: 0 Channels: 0 Exchanges: 8 Queues: 0 Consumers: 0

▼ Nodes

Name	File descriptors (?)	Socket descriptors (?)	Erlang processes	Memory	Disk space	Uptime	Type
rabbit@amqp	26 1024 available	1 829 available	195 1048576 available	30.2MB 1.5GB high watermark	17.7GB 953.7MB low watermark	3h 37m	Disc Stats

▼ Ports and contexts

Listening ports

Protocol	Bound to	Port
amqp	::	5672

Web contexts

Context	Bound to	Port	SSL	Path
RabbitMQ Management	0.0.0.0	15672	o	/
Redirect to port 15672	0.0.0.0	55672	o	/

Kuvio 21. RabbitMQ hallintapaneeli

Hallintapaneelin pääsivu sisältää yleisnäkymän vaihteen toimintaan, kuten yhteyksien, jonojen ja asiakkaiden lukumäärään. Sivulta nähdään myös palvelimen tila ja resurssien kulutus.

Exchanges-valikosta nähdään RabbitMQ-vaihteet. Oletuksena sivu sisältää kahdeksan palvelimen sisäiseen käyttöön tarkoitettua vaihdetta. ”Add a new exchange”-painikkeesta voidaan luoda uusi vaihde. Uudelle vaihteelle tulee määrittää nimi, tyyppi ja kestävyys. ”Auto delete”-valinta määrittää poistetaanko vaihde kun siihen ei ole liittynenä yhtään jonoa. ”Internal”-valinta määrittää onko vaihde vain palvelimen sisäiseen käyttöön. ”Alternate Exchange”-kohdassa voidaan määrittää vaihde, johon reitittämättömät viestit ohjataan. ”Arguments”-kohdassa voidaan asettaa ylimääräisiä muuttujia vaihteeseen.

Lokijärjestelmää varten luotiin ”Labranet-loki”-niminen topic-tyypin vaihde, kuten kuviossa 22.

Exchanges

▼ All exchanges

 Filter:

Name	Type	Policy	Parameters	Message rate in	Message rate out
(AMQP default)	direct		D		
amq.direct	direct		D		
amq.fanout	fanout		D		
amq.headers	headers		D		
amq.match	headers		D		
amq.rabbitmq.log	topic		D		
amq.rabbitmq.trace	topic		D		
amq.topic	topic		D		

▼ Add a new exchange

Name: *
 Type: ▼
 Durability: ▼
 Auto delete: (?) ▼
 Internal: (?) ▼
 Alternate exchange: (?)
 Arguments: = ▼

Kuvio 22. Vaihteen luonti

Vaihteeseen ei tarvitse määrittää jonoja, vaan ne voidaan tehdä viestien vastaanottajien (Logstash) asetuksissa. Vastaanottajan asetukseen määritetään haluttu jono ja siihen liitettävä reititysavain, jolloin vaihde osaa automaattisesti luoda jonon ja liitokset vastaanottajan avatessa siihen yhteyden. Jonot ja liitokset voidaan tarvittaessa määrittää myös käsin vaihteeseen.

Kuviossa 23 nähdään Labranet-loki -vaihteen liitokset. Vaihde välittää log.raw-reititysavaimella saapuvat viestit log.parser- ja log.storage-jonoihin.

▼ Bindings

This exchange

⇓

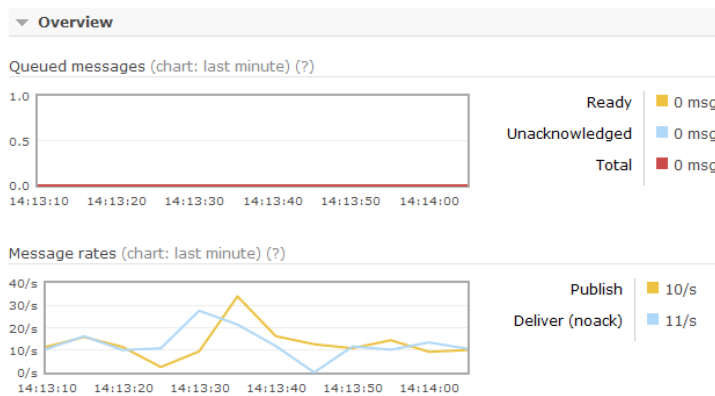
To	Routing key	Arguments	
log.parser	log.raw		Unbind
log.storage	log.raw		Unbind

Kuvio 23. Labranet-loki liitokset

Queues-valikon kautta päästään tarkastelemaan vaihteeseen liitettyjen jonojen tietoja. Jokaisen jonon kohdalla nähdään reaaliajassa päivittyvä kuvio kyseiseen jonoon julkaistuista ja vastaanottajille toimitetuista viesteistä. Queued messages kertoo kuinka monta viestiä odottaa jonossa toimitusta vastaanottajille.

Kuviossa 24. nähdään log.parser-jonon viestien määrä viimeisen minuutin ajalta.

Queue log.parser



Kuvio 24. Jonon viestimäärät

Message rates breakdown -kohdasta nähdään vastaanotettujen viestien lähde ja mihin ne toimitetaan. Consumers-kohdassa nähdään jonoon liittyneet viestien vastaanottajat.

Kuviossa 25 nähdään log.parser-jonon viestien lähteet ja asiakkaat.

Message rates breakdown			
Incoming			Deliveries
Exchange	publish	confirm	
Labranet-loki	32/s		
Channel	deliver / get	ack	
192.168. :37653 (1)	28/s		
Consumers			
Channel	Consumer tag	Ack required	Exclusive
192.168. :37653 (1)	amq.ctag-kHFsrnxJlNq9vbxlm_jpoQ	o	o

Kuvio 25. Jonon viestien lähteet ja asiakkaat

Connections-valikko näyttää palvelimeen yhteydessä olevien asiakkaiden tiedot, kuten IP-osoitteen, tilan ja liikennemäärän.

Kuviossa 26 nähdään AMQP-palvelimen yhteydet.

Connections

Filter:

Network						Overview		
Name	Protocol	Client	From client	To client	Timeout	Channels	User name	State
192.168. :38344	AMQP 0-9-1	RabbitMQ / Java 3.1.3	3.3kB/s (876.5MB total)	0B/s (1.3kB total)	600s	1		running
192.168. :46537	AMQP 0-9-1	RabbitMQ / Java 3.1.3	7.9kB/s (297.3MB total)	0B/s (1.3kB total)	600s	1		running
192.168. :35916	AMQP 0-9-1	RabbitMQ / Java 3.1.3	0B/s (1.7kB total)	16.0kB/s (1.0GB total)	600s	1		running
192.168. :36854	AMQP 0-9-1	RabbitMQ / Java 3.1.3	0B/s (2.3kB total)	16.0kB/s (1.2GB total)	600s	1		running

Kuvio 26. AMQP-palvelimen yhteydet

Users-valikon kautta voidaan hallita palvelimen käyttäjiä ja asettaa heille oikeudet vain tiettyihin resursseihin.

Kuviossa 27 nähdään AMQP-palvelimen käyttäjienhallintavalikko.

Users

▼ All users

Filter:

Name	Tags	Can access virtual hosts	Has password
guest	administrator	/	•

(?)

▼ Add a user

Username:

Password: (confirm)

Tags: (?)
[Admin] [Monitoring] [Management] [None]

Kuvio 27. AMQP-palvelimen käyttäjienhallinta

Palvelimelle luotiin admin-käyttäjä, jota ylläpitäjät käyttävät RabbitMQ management -verkkokäyttöliittymän hallintaan. Samalla quest-käyttäjältä otettiin administrator oikeudet pois. Parsinta- ja säilytyspalvelinten Logstash-sovellukset käyttävät guest-käyttäjää oletuksena ottaessaan yhteyden vaihteeseen.

13.6 Parsinta- ja tarkastelupalvelin

13.6.1 Kuvaus ja resurssit

Lokiviestien parsinta ja lyhytaikainen säilytys viestien tarkastelua varten on yhdistetty yhteen virtuaalikoneeseen. Logstash-sovellus noutaa viestit log.parser-jonosta ja suorittaa niille tarvittavan parsinnan, kuten sisällön erittelyn kenttiin. Logstash ohjaa parsitut viestit Graylog2-sovellukselle, joka tallentaa viestit Elasticsearch-tietokantaan. Graylog2-verkkokäyttöliittymän avulla vastaanotettuja viestejä voidaan tarkastella reaaliajassa sekä hakea ja suodattaa viestejä käyttäjän määrittämien ehtojen perusteella.

Palvelinta varten luotiin ”Graylog2”-niminen virtuaalikone seuraavilla resursseilla:

- CPU: 4 kpl
- muisti: 4 GB
- tallennustila: 125 GB.

Tallennustila jaettiin kahteen kiintolevyyn, joista järjestelmälevylle asetettiin 15 gigatavua tallennustilaa ja datalevylle 110 gigatavua. Järjestelmälevy sisältää käyttöjärjestelmän tiedostot. Datalevy sisältää Elasticsearch- ja MongoDB-tietokantojen tiedostot sekä käyttöjärjestelmän lokiviestit. Levyjen osiointi suoritettiin LVM (Logical Volume Manager) avulla, jossa levyt jaetaan tallenneryhmiin ja niiden sisältää loogisiin osiin. LVM:n käyttö mahdollistaa yksittäisten osioiden kasvattamisen myöhemmin lisäämällä uuden virtuaalikiintolevyn tallenneryhmään.

13.6.2 Graylog2-asennus

Graylog2-palvelin vaatii toimiakseen Elasticsearch-tietokannan version 0.20.4 ja uudehkon version MongoDB-tietokannasta. Ensimmäisenä asennettiin Elasticsearch-tietokannan vaatima Java.

```
yum install java-1.7.0-openjdk
```

Elasticsearch-tietokannan lataus ja purku suoritettiin seuraavilla komennoilla:

```
wget https://download.elasticsearch.org/elasticsearch/elasticsearch/elasticsearch-0.20.4.tar.gz
tar xzvf elasticsearch-0.20.4.tar.gz
```

Seuraavaksi asennettiin service wrapper, jolla tietokantaa hallitaan.

```
curl -k -L http://github.com/elasticsearch/elasticsearch-
servicewrapper/tarball/master | tar -xz
mv *servicewrapper*/service /opt/elasticsearch-0.20.4/bin/
rm -Rf *servicewrapper*
/opt/elasticsearch-0.20.4/bin/service/elasticsearch install
```

Elasticsearch.yml-tiedostoon määritettiin klusterin- ja noden-nimi sekä sirpaleiden (shard) ja kopioiden (replica) määrä. Sirpalaiden määräksi asetettiin oletusarvo 5. Kopioiden määrä asetettiin levytilan säästämiseksi nolnaan.

```
cluster.name: graylog2
node.name: "Graylog2_server"
```

```
index.number_of_shards: 5
index.number_of_replicas: 0
```


Palvelu käynnistettiin ja otettiin käyttöön komendoilla:

```
service elasticsearch start
chkconfig elasticsearch on
```

Mongodb asennusta varten palvelimeen lisättiin sovelluksen virallinen pakettivarausto.

```
cat << EOF >> /etc/yum.repos.d/10gen.repo
[10gen]
name=10gen Repository
baseurl=http://downloads-distro.mongodb.org/repo/redhat/os/x86_64
gpgcheck=0
enabled=1
EOF
```

Mongodb asennettiin komennolla:

```
yum install mongo-10gen-server
```

Palvelu käynnistettiin ja otettiin käyttöön komendoilla:

```
service mongod start
chkconfig mongod on
```

Graylog2-palvelinta varten luotiin Mongodb-tietokanta ja käyttäjä.

```
mongo

use graylog2
db.addUser('graylog', <salasana>)
db.auth('graylog', <salasana>)
```

Seuraavaksi asennettiin itse Graylog2-palvelin. Paketin lataamisen ja purkamisen jälkeen asetustiedostot kopioitiin oikeaan paikkaan.

```
cp /opt/graylog2-server-0.12.0/graylog2.conf.example /opt/graylog2-server-0.12.0/graylog2.conf
cp /opt/graylog2-server-0.12.0/elasticsearch.yml.example /opt/graylog2-server-0.12.0/graylog2-elasticsearch.yml
```

Palvelimen asetukset määritettiin graylog2.conf-tiedostoon. Elasticsearch-asetustiedoston polku muutettiin oikeaksi.

```
elasticsearch_config_file = /opt/graylog2-server-0.12.0/graylog2-elasticsearch.yml
```

Mongodb-asetukset määritettiin vastaamaan aiemmin luotua tietokantaa.

```
mongodb_useauth = true  
mongodb_user = graylog  
mongodb_password = <salasana>  
mongodb_host = 127.0.0.1  
mongodb_database = graylog2  
mongodb_port = 27017
```

Graylog2-palvelin asetettiin vastaanottamaan Logstash-sovelluksen lähettämät GELF-viestit.

```
use_gelf = true  
gelf_listen_address = 127.0.0.1  
gelf_listen_port = 12201
```

Graylog2-elasticsearch.yml -tiedosto sisältää Elasticsearch-tietokantaa koskevat asetukset. Klusterin nimeksi määritettiin sama arvo, kuin tietokannan luonnin yhteydessä.

```
cluster.name: graylog2
```

Tietokannan osoitteeksi asetettiin palvelimen paikallinen osoite.

```
network.host: 127.0.0.1
```

Graylog2 asetettiin käyttämään unicast ping -viestejä Elasticsearch-klusterin löytämiseen.

```
discovery.zen.ping.multicast.enabled: false  
discovery.zen.ping.unicast.hosts: ["127.0.0.1:9300"]
```

Palvelun suorittamista varten ladattiin init-skripti, jolle asetettiin tarvittavat oikeudet.

```
wget -O /etc/init.d/graylog2-server
https://gist.github.com/marzacchi/1659948/raw/fa197ce9f82f17006d9960edef8f7ef0e2c853da/etc-init.d-graylog2.sh
chmod +x /etc/init.d/graylog2-server
```

Init-skriptiin asetettiin oikeat polut palvelimen asetustiedostoon.

```
GL_HOME=/opt/graylog2-server-0.12.0/
CONFIG_FILE=/opt/graylog2-server-0.12.0/graylog2.conf
```

Lopuksi palvelu käynnistettiin ja otettiin käyttöön komennoilla:

```
service graylog2 start
chkconfig graylog2-server on
```

Seuraava vaihe oli verkkokäyttöliittymän asennus. Verkkokäyttöliittymä vaatii toimiaukseen Ruby-ohjelmointikielen, jonka uusin vakaa versio asennettiin RVM-skriptin avulla. RVM-skriptiä käytettiin koska CentOS-järjestelmän pakettivarastot eivät sisällä riittävän tuoretta versiota ohjelmointikielestä.

```
curl -L https://get.rvm.io | bash -s stable --ruby
source /usr/local/rvm/scripts/rvm
rvm install ruby-dev
```

Ruby-kirjastojen asentamista varten asennettiin bundler-paketti ilman dokumentaatioita sekä muut vaadittavat paketit.

```
gem install bundler --no-rdoc --no-ri
gem install bson_ext mongo json
```

Loput riippuvuudet asennettiin pakettihallinnan kautta.

```
yum install httpd httpd-devel libcurl-devel gd gd-devel
```

Verkkokäyttöliittymän tiedostot ladattiin ja purettiin.

```
wget -O graylog2-web-interface.tar.gz https://github.com/Graylog2/graylog2-web-interface/releases/download/0.12.0/graylog2-web-interface-0.12.0.tar.gz
tar xzvf graylog2-web-interface.tar.gz
mv graylog2-web-interface-0.12.0/ graylog2-web-interface
```

Asetustiedostot sijaitsevat config-kansion alla. Mongoid.yml-tiedostoon asetettiin Mongoddb-tietokannan tiedot.

```
production:
  host: localhost
  port: 27017
  username: graylog
  password: <salasana>
  database: graylog2
```

Seuraavaksi asennettiin tarvittavat kirjastot bundler-sovelluksen avulla.

```
cd /opt/graylog2-web-interface/
bundle install --without=development
```

Verkkokäyttöliittymä vaatii toimiakseen salaisen avaimen luonnin. Avain luotiin rake secret -komennolla ja lisättiin config/initializers/secret_token.rb-tiedostoon.

Rails-verkkosovelluksia on mahdollista suorittaa Apache-verkkopalvelimella Phusion Passenger -sovelluksella. Passenger asennettiin komennoilla:

```
gem install passenger --no-rdoc --no-ri
passenger-install-apache2-module
```

Apache-palvelimen httpd.conf-tiedostoon lisättiin Passenger-moduulin määrittäykset.

```
LoadModule passenger_module /usr/local/rvm/gems/ruby-2.0.0-
p247/gems/passenger-4.0.19/buildout/apache2/mod_passenger.so
PassengerRoot /usr/local/rvm/gems/ruby-2.0.0-p247/gems/passenger-4.0.19
PassengerDefaultRuby /usr/local/rvm/wrappers/ruby-2.0.0-p247/ruby
```

Graylog2-verkkokäyttöliittymää varten luotiin /etc/httpd/conf.d/-kansioon graylog2.conf-tiedosto, joka sisältää sivuston virtual-host -määrittäksen.

```
<VirtualHost *:80>
  DocumentRoot /opt/graylog2-web-interface/public

  RailsEnv 'production'

  <Directory /opt/graylog2-web-interface/public>
    Allow from all
```

```
Options -MultiViews
</Directory>
```

```
ErrorLog /var/log/apache2/error.log
LogLevel warn
CustomLog /var/log/apache2/access.log combined
</VirtualHost>
```

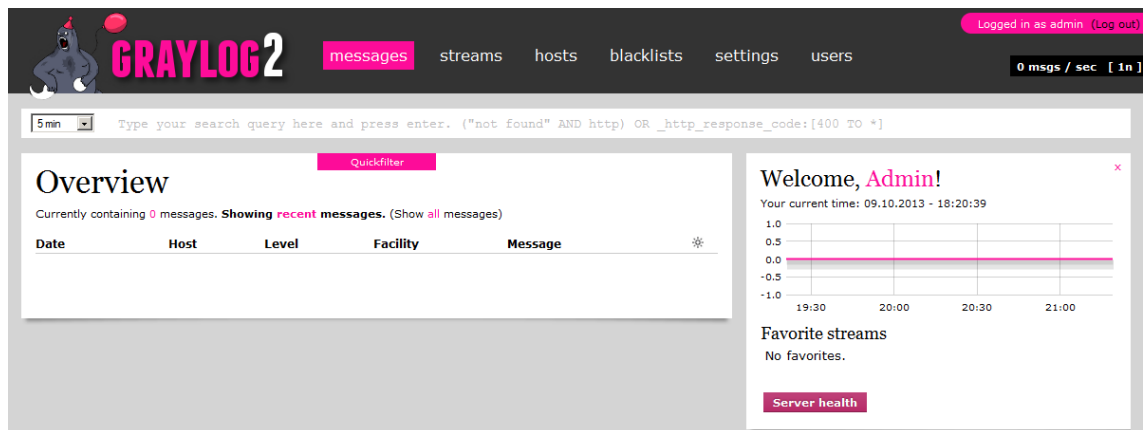
Apache-käyttäjä asetettiin verkkokäyttöliittymän tiedostojen omistajaksi.

```
chown apache:apache /opt/graylog2-web-interface/
```

Lopuksi apache-verkkopalvelin käynnistettiin ja otettiin käyttöön.

```
service httpd start
chkconfig httpd on
```

Verkkokäyttöliittymän toiminta testattiin ottamalla yhteys selaimella palvelimen IP-osoitteeseen. Ensimmäistä kertaa yhdistettäessä Graylog2 pyytää luomaan käyttäjätunnuksen. Hallintaa varten luotiin admin-niminen käyttäjä. Käyttäjän luomisen ja kirjautumisen jälkeen avautuu pääsivu, joka nähdään kuviossa 28.



Kuvio 28. Graylog2 pääsivu

Graylog2-verkkokäyttöliittymän autentikaatio on mahdollista liittää LDAP-tietokantaan. LDAP-asetukset määritettiin /opt/graylog2-web-interface/config/ldap.yml tiedostoon:

```
enabled: true
host: ldap.xx.xx
port: 636
tls_enabled: true
```

```

displayname_attribute: cn
mail_attribute: mail
search_base_dn: <base_dn>
search_filter: (uid=%s)
search_bind_dn: <search_bind_dn>
search_bind_password: <salasana>

```

Verkkopalvelimen uudelleenkäynnistyksen jälkeen verkkokäyttöliittymään kirjautuminen onnistuu LDAP-tietokannan käyttäjätunnuksilla.

13.6.3 Logstash-asetukset

Logstash-asetustiedostoja varten palvelimelle luotiin logstash.d-kansio. Kansion alle asetettiin input- ja output-määrittysten ohella jokaiselle viestityypille oma suodatussäännöt sisältävä tiedosto. Logstash yhdistää tiedostot yhdeksi kokonaisuudeksi nimen perusteella aakkosjärjestyksessä. Tästä syystä tiedostot nimettiin muodossa "numero_nimi.conf", jolloin järjestys voidaan määrätä numeron perusteella.

Ensimmäisenä määritettiin rabbitmq-sisääntulo viestien vaihteesta noutamiseksi. Parametreina asetettiin vaihteen IP-osoite ja nimi, jonon nimi, reititysavain sekä jonon kestävyys.

```

input {
  rabbitmq {
    host => "192.168.x.x"
    exchange => "Labranet-loki"
    queue => "log.parser"
    key => "log.raw"
    durable => true
    auto_delete => false
    exclusive => false
    ack => false
  }
}

```

Tiedostonimeksi asetettiin "00_input.conf", jotta Logstash suorittaa sen ensimmäisenä.

Viestien Graylog2-sovellukselle välitystä varten asetettiin GELF-ulostulo, johon kohdeosoitteeksi määritettiin palvelimen paikallinen osoite (127.0.0.1) ja portiksi 12201.

```
output {
  gelf {
    host => "127.0.0.1"
    port => 12201
  }
}
```

Tiedostonimeksi asetettiin "99_output.conf", jotta Logstash suorittaa sen viimeisenä.

Lokilähteiden suodatussäännöt määritettiin erillisiin tiedostoihin. Suodatussääntöjen alkuun asetettiin ehtolause, jolla varmistetaan suodatuksen suoritus vain kyseisen lähteen viesteille. Esimerkiksi Juniper-laitteille luodun suodattimen alkuun lisättiin ehtolause, jonka perusteella suodatin suoritetaan ainoastaan kyseisten laitteiden IP-osoitteista vastaanotetuille viesteille.

```
filter {
  if [host] in "192.168.x.x 192.168.x.x 192.168.x.x 192.168.x.x 192.168.x.x" {
```

Lokilähteiden suodatussäännöt on esitetty erillisissä kappaleissa.

13.7 Kytkimet

13.7.1 Yleistä

LabraNet-verkko sisältää neljätoista access-kytkin, joihin verkon työasemat kytkeytyvät. Osa kytkimistä on yhdistetty virtual chassis -tekniikan avulla loogiseksi kokonaisuudeksi, jolloin ne toimivat yhdessä kuin yksi kytkin. Lisäksi verkossa on keskuskytkin, johon kaikki access-kytkimet liittyvät. Kytkimet on lueteltu taulukossa 5.

Taulukko 5. LabraNet-kytkimet

Nimi	Valmistaja	Malli	IP-osoite
Kerros1-Access	Juniper	EX	192.168.x.x
Kerros3-Access	Juniper	EX	192.168.x.x
Kerros3-Access2	Juniper	EX	192.168.x.x
Kerros4-access	Juniper	EX	192.168.x.x
Kerros4-Cisco	Cisco	Catalyst	192.168.x.x
Kerros5-Cisco	Cisco	Catalyst	192.168.x.x

Cisco Nexus	Cisco	Nexus	192.168.x.x
-------------	-------	-------	-------------

Lisäksi Spidernet-ympäristö sisältää edustakytkimiä, joilta halutaan kerätä lokitietoja.

Taulukko 6. Spidernet-kytkimet

Nimi	Valmistaja	Malli	IP-osoite
Spidernet1	Juniper	EX	192.168.x.x
Spidernet2	Cisco	Catalyst	192.168.x.x
Spidernet3	Cisco	Catalyst	192.168.x.x

13.7.2 Juniper-kytkimet

Juniper EX -sarjan kytkinten JUNOS-käyttöjärjestelmä sallii lokiviestien tulostamiseen käyttäjien konsoliin, tallentamisen tiedostoon tai lähetyksen etäpalvelimelle Syslog-protokollalla. Kerättävät viestit voidaan määrittää facility- tai severity-arvon perusteella.

Juniper-kytkimet asetettiin lähettämään lokiviestit Collector1-keräyspalvelimelle. Ensimmäisenä määritettiin authorization-facility, joka sisältää käyttäjien autentikointiin ja valtuutukseen liittyvät tapahtumat. Vakavuustasoksi asetettiin info.

```
set system syslog host 192.168.x.x authorization info
```

Change-log -facility sisältää muutokset laitteen konfiguraatioihin.

```
set system syslog host 192.168.x.x change-log info
```

Interactive-commands -facility sisältää käyttäjien syöttämät komennot.

```
set system syslog host 192.168.x.x interactive-commands info
```

Kerättäväksi määritettiin myös kaikki error-tasosta vakavammat tapahtumat.

```
set system syslog host 192.168.x.x any error
```

Aikaleiman formaatti asetettiin sisältämään vuoden ja millisekunnit.

```
set system syslog time-format year millisecond
```


Vaatimusmäärittelyssä asetettuja MAC-port ja DHCP-binding -viestejä ei pystynyt lähettämään kytkimeltä etäpalvelimelle.

Juniper kytkinten tuottamien lokiviestien formaatti on seuraava:

```
<Syslog-priv> aikaleima hostname prosessi[pid]: %-viestikoodi: viesti
```

Kytken suodatussäännöt määritettiin "05-juniper.conf"-asetustiedosto. Ensimmäisenä määritettiin grok-suodatin, jolla viestien sisältö voidaan parsia kenttiin. Grok-suodattimen match-parametriä käytetään tiettyjä malleja vastaavien merkkijonojen löytämiseen tekstistä. Match-parametrin syntaksissa määritetään haluttu kenttä ja siihen täsmäävät mallit.

Logstash sisältää yli 120 valmista mallia ja niitä voi myös luoda itse regular-expression-lausekkeiden avulla. Malli määritetään seuraavasti: `{SYNTAX:SEMANTIC}`, jossa SYNTAX on mallin nimi ja SEMANTIC täsmääväle tekstille annettava tunnus. Esimerkiksi `{IP:ip_address}` täsmää IP-mallin mukaiseen tekstiin ja antaa sille tunnuksen `ip_address`.

Juniper kytkinten lokiviesteille luotiin grok-suodatin, jolla viesteistä erotetaan yleiset Syslog-kentät, kuten `pri` ja `hostname`:

```
grok {
    patterns_dir => [ "/opt/logstash/patterns/" ]
    match => [ "message",
"<{%POSINT:syslog_pri}>{%JUNIPER_TIMESTAMP}
{%SYSLOGHOST:hostname} {%DATA:program}(?:\{%POSINT:pid}\})?: %-
{%DATA:juniper_message_code}: {%GREEDYDATA}" ]
}
```

`JUNIPER_TIMESTAMP`-malli määritettiin `/opt/logstash/patterns/juniper-tiedostoon`:

```
JUNIPER_TIMESTAMP {%MONTH} +{%MONTHDAY} {%TIME} {%YEAR}
```

`Patterns_dir`-parametrissa asetetaan polku mallit sisältävään kansioon.

Seuraavaksi määritettiin grok-suodattimet, joilla viesteistä saadaan poimittua käyttäjä ja syötetty komento:

```

grok {
    match => [ "message", "%{DATA} User
%{WORD:juniper_user}%{GREEDYDATA}" ]
}

grok {
    match => [ "message", "%{DATA} command
%{DATA:juniper_command} %{GREEDYDATA}" ]
}

```

Syslog_pri-suodatin erottaa Syslog-viestien pri-kentän koodin facility- ja severity-arvoiksi:

```

syslog_pri {
}

```

Mutate-suodattimella poistetaan ylimääräiset tags- ja version-kentät:

```

mutate {
    remove_field => [ "tags", "@version" ]
}

```

Suodatuksen tuloksena saadaan JSON-formaatin viesti, jossa viestin osat on erotettu omiin kenttiin:

```

{"message":"<190>Oct 10 17:46:23.777 2013 xxx mgd[39822]: %-
UI_CMDLINE_READ_LINE: User 'test', command 'exit '",
"@timestamp":"2013-10-10T14:46:22.695Z",
"host":"192.168.x.x",
"syslog_pri":"190",
"hostname":"xxx",
"program":"mgd",
"pid":"39822",
"juniper_message_code":"UI_CMDLINE_READ_LINE",
"juniper_user":"test",
"juniper_command":"exit",
"syslog_severity_code":6,
"syslog_facility_code":23,
"syslog_facility":"local7",
"syslog_severity":"informational"}

```

13.7.3 Cisco-kytkimet

Cisco-kytkimet mahdollistavat lokiviestien tallentamisen laitteen muistiin tai ulkoiselle Syslog-palvelimelle. Valitettavasti Cisco Catalyst -mallin kytkimiltä ei onnistuttu saamaan keräämisen arvoista lokitietoa. Syynä tähän on kytkinten vanhan IOS-käyttöjärjestelmäversion puutteelliset lokiominaisuudet.

Cisco Nexus kytkin sisältää NX-OS -käyttöjärjestelmän, joka mahdollistaa paremman lokienhallinnan kuin Catalyst -kytkimet. Nexus on jaettu useaan VDC (Virtual Device Context) -instanssiin, jotka toimivat kuin erilliset fyysiset kytkimet, sisältäen omat kytkentätaulut ja asetukset. Lokiviestit määritettiin kerättäväksi LabraNettiin kuuluvasta instanssista.

Ensimmäisenä asetettiin lokiviestit lähetettäväksi keräyspalvelimelle käyttäen VRF (Virtual Routing and Forwarding Instance) -instanssia.

```
logging server 192.168.x.x use-vrf xxx
```

Kerättäviksi lokityypeiksi asetettiin AAA (Authentication, Authorization, Accounting), confcheck ja security.

```
logging level aaa 6
logging level confcheck 6
logging level securityd 6
```

Cisco Nexus lokiviestit ovat muotoa:

```
<pri>: aikaleima: %facility-severity-mnemonic: viesti
```

Viestejä varten luotiin seuraava grok-suodatin:

```
grok {
    match => [ "message", "<{%POSINT:syslog_pri}>: {%DATA}:"
%%{%WORD:cisco_facility}-{%INT:cisco_severity}-{%WORD:cisco_msg_type}:"
 {%GREEDYDATA}" ]
}
```

Grok-suodattimen lisäksi asetettiin pri- ja mutate-suodattimet:

```

syslog_pri {
}

mutate {
    remove_field => [ "tags", "@version" ]
}

```

Esimerkki suodatuksen tuloksena saatavasta viestistä:

```

{"message": "<190>: 2013 Oct 13 17:58:43.566 EEST: %AAA-6-
AAA_ACCOUNTING_MESSAGE: start:192.168.x.x@pts/2:confbackup.",
"@timestamp": "2013-10 13T15:05:49.397Z",
"host": "192.168.x.x",
"syslog_pri": "190",
"cisco_facility": "AAA",
"cisco_severity": "6",
"cisco_msg_type": "AAA_ACCOUNTING_MESSAGE",
"syslog_severity_code": 6,
"syslog_facility_code": 23,
"syslog_facility": "local7",
"syslog_severity": "informational"}

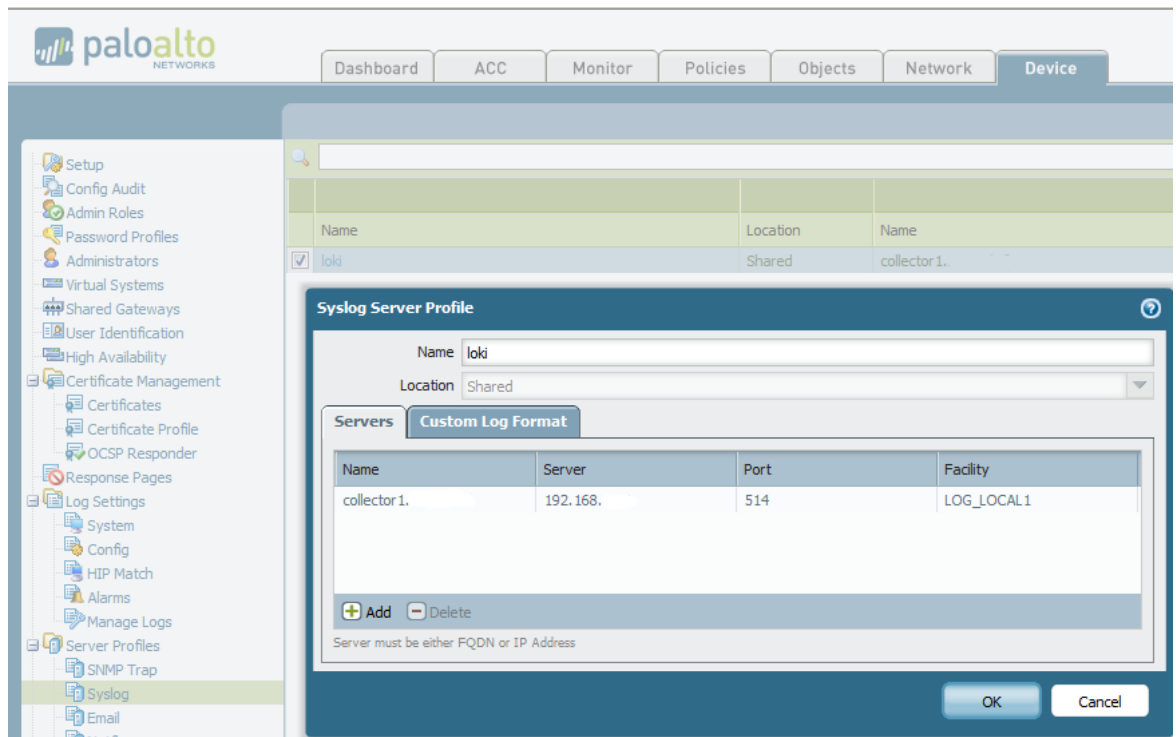
```

13.8 Palomuri

LabraNet-verkon palomuurina toimiva Paloalto mahdollistaa verkkoliikenteen, havaittujen tietoturvahkien, asetusmuutosten ja järjestelmän tuottamien tapahtumien tallentamisen. Tapahtumat voidaan lähettää etäkohteeseen Syslog-protokollan, sähköpostin tai SNMP trap -viestien avulla.

Laitteen Syslog-asetukset määritetään device-välilehdellä service profiles -valikon Syslog-kohdassa. Keräyspalvelinta varten luotiin profiili, johon asetettiin Collector1-keräyspalvelimen IP-osoite ja portti. Facility-arvoksi asetettiin LOG_LOCAL1. Log-settings valikon System-kohtaan asetettiin lokiprofiili käyttöön informal-, low-, medium-, high-, ja critical-tasoilla.

Kuviossa 29 nähdään palomuriin tehty Syslog Server Profile.



Kuvio 29. Paloalto Syslog-asetukset

Palomuuri tuottaa lokiviestit CSV-muodossa, jossa arvot erotetaan toisistaan pilkulla. Traffic-typin lokiviestien formaatti on seuraava:

FUTURE_USE, Receive Time, Serial Number, Type, Subtype, FUTURE_USE, Generated Time, Source IP, Destination IP, NAT Source IP, NAT Destination IP, Rule Name, Source User, Destination User, Application, Virtual System, Source Zone, Destination Zone, Ingress Interface, Egress Interface, Log Forwarding Profile, FUTURE_USE, Session ID, Repeat Count, Source Port, Destination Port, NAT Source Port, NAT Destination, Port, Flags, Protocol, Action, Bytes, Bytes Sent, Bytes Received, Packets, Start Time, Elapsed Time, Category, FUTURE_USE, Sequence Number, Action Flags, Source Location, Destination Location, FUTURE_USE, Packets Sent, Packets Received

Threat-typin viestien formaatti on seuraava:

FUTURE_USE, Receive Time, Serial Number, Type, Subtype, FUTURE_USE, Generated Time, Source IP, Destination IP, NAT Source IP, NAT Destination IP, Rule Name, Source User, Destination User, Application, Virtual System, Source Zone, Destination Zone, Ingress Interface, Egress Interface, Log Forwarding Profile, FUTURE_USE, Session ID, Repeat Count, Source Port, Destination Port, NAT Source Port, NAT Destination Port, Flags, Protocol, Action, Miscellaneous, Threat ID, Category, Severity, Direction, Sequence Number, Action Flags, Source Location, Destination Location, FUTURE_USE, Content Type

Configuration-typin viestien formaatti on seuraava:

FUTURE_USE, Receive Time, Serial Number, Type, Subtype, FUTURE_USE, FUTURE_USE, Host, Virtual System, Command, Admin, Client, Result, Configuration Path, Sequence Number, Action Flags

System-tyypin viestien formaatti on seuraava:

FUTURE_USE, Receive Time, Serial Number, Type, Subtype, FUTURE_USE, FUTURE_USE, Virtual System, Event ID, Object, FUTURE_USE, FUTURE_USE, Module, Severity, Description, Sequence Number, Action Flags

Lisäksi viestien alkuun lisätään Syslog pri-arvo ja aikaleima.

Palomuurin suodatussäännöt määritettiin "03_paloalto.conf"-tiedostoon. Ensimmäisenä määritettiin grok-suodatin, joka poimii viestien Syslog pri -kentän ja tyyppin. Tyyppi-kenttää käytetään erottamaan traffic-, threat-, config- ja system-tyypin viestit, jotta niihin voidaan soveltaa erillisiä suodattimia.

```
grok {
    match => [ "message",
"<{%POSINT:syslog_pri}>{%SYSLOGTIMESTAMP}
{%INT},{%DATA},{%DATA},{%WORD:pa_type},{%GREEDYDATA}" ]
}
```

Viestityyppien erotus tapahtuu if-ehtolauseen avulla, esimerkiksi traffic-tyypin viestejä varten luotiin ehtolause:

```
if [pa_type] == "TRAFFIC" {
```

Lokiviestien kenttien erotteluun käytetään csv-suodatinta, joka osaa poimia pilkulla erotetut arvot kentiksi. Columns-parametri lisää kentille käyttäjän määrittämät nimet. Viesteistä poistettiin tarpeettomat kentät remove_field -parametrilla.

```
csv {
    columns => [
"timestamp","receive_time","serial","pa_type","subtype","future","generation_time",
"source_ip","destination_ip","nat_source_ip","nat_destination_ip","rule","source_u",
"ser","destination_user","application","vsys","source_zone","destination_zone","ing",
"ress_interface","egress_interface","log_profile","timestamp2","session_id","repeat",
"_count","source_port","destination_port","nat_src_port","nat_dst_port","flags","prot",
"ocol","action","bytes","bytes_send","bytes_received","packets","start_time","elaps",
"ed","category","future","sequence","action_flags","source_location","destination_lo",
"cation","future","packets_send","packets_received" ]
```

```

                                remove_field => [
"@version", "timestamp", "future", "receive_time", "serial", "subtype", "future", "log_profile", "timestamp2", "repeat_count", "flags", "bytes", "bytes_send", "bytes_received", "packets", "start_time", "elapsed", "category", "sequence", "action_flags", "packets_send", "packets_received" ]
                                }
        }

```

Muille viestityypeille luotiin vastaavat suodattimet.

```

if [pa_type] == "THREAT" {

                                csv {

                                        columns => [
"timestamp", "receive_time", "serial", "pa_type", "subtype", "future", "generation_time", "source_ip", "destination_ip", "nat_source_ip", "nat_destination_ip", "rule", "source_user", "destination_user", "application", "vsys", "source_zone", "destination_zone", "ingress_interface", "egress_interface", "log_profile", "timestamp2", "session_id", "repeat_count", "source_port", "destination_port", "nat_src_port", "nat_dst_port", "flags", "protocol", "action", "misc", "threat_id", "category", "severity", "direction", "sequence", "action_flags", "source_location", "destination_location", "future", "content_type" ]

                                        remove_field => [
"@version", "timestamp", "future", "receive_time", "serial", "subtype", "future", "generation_time", "log_profile", "timestamp2", "repeat_count", "flags", "category", "sequence", "action_flags" ]
                                        }
                                }

if [pa_type] == "CONFIG" {

                                csv {

                                        columns => [
"timestamp", "receive_time", "serial", "pa_type", "subtype", "future", "future", "src_host", "vsys", "command", "admin", "client", "result", "config_path", "sequence", "action_flags" ]

                                        remove_field => [
"@version", "timestamp", "receive_time", "serial", "future", "sequence", "action_flags" ]
                                        }
                                }

if [pa_type] == "SYSTEM" {

                                csv {

```

```

        columns => [
"timestamp", "receive_time", "serial", "pa_type", "subtype", "future", "future", "vsys", "ev
ent_id", "object", "future", "future", "module", "severity", "description", "sequence", "acti
on_flags" ]

        remove_field => [
"@version", "timestamp", "receive_time", "serial", "future", "sequence", "action_flags" ]
    }
}

```

Esimerkki suodatetusta traffic-tyypin viestistä:

```

{"message":["<142>Oct 18 14:26:08 1,2013/10/18
14:26:08,xxx, TRAFFIC,end, 1,2013/10/18
14:26:08,192.168.xx.xx,190.93.246.58,195.xx.xx.xx,190.93.246.58,Permit,,web-
browsing,vsysx,To-XXX,To-XXX,ae1.xx,ethernet1/xx,Lokille,2013/10/18
14:26:08,303301,1,35645,80,4482,80,0x404000,tcp,allow,52704,2207,50497,53,2
013/10/18 14:24:20,106,any,0,824955099,0x0,192.168.0.0-
192.168.255.255,Costa Rica,0,13,40\u0000"],
"@timestamp":"2013-1018T11:26:06.959Z",
"host":"192.168.x.x",
"pa_type":"TRAFFIC",
"generation_time":"2013/10/18 14:26:08",
"source_ip":"192.168.x.x",
"destination_ip":"190.93.246.58",
"nat_source_ip":"195.xx.xx.xx",
"nat_destination_ip":"190.93.246.58",
"rule":"Permit",
"source_user":null,
"destination_user":null,
"application":"web-browsing",
"vsys":"vsysx",
"source_zone":"To-XXX",
"destination_zone":"To-XXX",
"ingress_interface":"ae1.xx",
"egress_interface":"ethernet1/xx",
"session_id":"303301",
"source_port":"35645",
"destination_port":"80",
"nat_src_port":"4482",
"nat_dst_port":"80",
"protocol":"tcp",
"action":"allow",
"source_location":"192.168.0.0-192.168.255.255",
"destination_location":"Costa Rica"}

```


13.9 Linux-palvelimet

Linux-palvelimet tuottavat Syslog-formaatin lokiviestejä, joiden yleisten kenttien parsimiseen määritettiin yhteinen sääntötiedosto "10_linux_syslog.conf". Lokilähdekohtaiset parsintasäännöt määritettiin erillisiin tiedostoihin. Sääntö asetettiin kaikille infrastruktuurin tärkeille Linux-palvelimille.

Ensimmäisenä määritettiin grok-suodatin, jolla parsitaan Syslog-formaatin yleiset kentät:

```
grok {
    match => [ "message",
"<{%POSINT:syslog_pri}>{%SYSLOGTIMESTAMP} {%HOSTNAME:hostname}
{%SYSLOGPROG}: {%GREEDYDATA}" ]
}
```

Seuraavaksi asetettiin grok-suodattimet, jotka parsivat viesteistä autentikaatiotyy-
pin, rhost-osoitteen ja käyttäjänimen. Näitä kenttiä voidaan käyttää hyväksi palvelinten pääsynvalvonnassa.

```
grok {
    match => [ "message", "%{DATA} authentication
{%WORD:pam_auth_type};{%GREEDYDATA}" ]
}

grok {
    match => [ "message", "%{DATA} rhost=%{DATA:pam_rhost}
user=%{WORD:pam_source_user};{%GREEDYDATA}" ]
}
```

Loppuun asetettiin vielä syslog_pri- ja mutate-suodattimet:

```
syslog_pri {
}

mutate {
    remove_field => [ "tags", "@version" ]
}
```

13.10 Nimipalvelimet

Verkko sisältää kahdenlaisia nimipalvelimia: resolvereja ja autoritäärisiä-nimipalvelimia. Resolverien tehtävä on selvittää asiakaslaitteiden tekemät nimikyselyt. Autoritääriset-nimipalvelimet vastaavat niiden hallinnoivia zoneja koskeviin nimikyselyihin. LabraNet-ympäristö sisältää kaksi resolveria ja neljä autoritääristä nimipalvelinta.

Kyseiset palvelimet ovat CentOS-palvelimia, joissa nimipalveluna toimii Bind ja lokienhallinnasta huolehtii Rsyslog. Rsyslog asetettiin lähettämään lokiviestit Collector2-keräyspalvelimelle lisäämällä /etc/rsyslog.conf-tiedostoon rivi:

```
daemon.info;authpriv.*; *.alert @192.168.x.x:514
```

Lähetettäviksi viesteiksi määritettiin daemon-facilityn info-vakavuustason viestit sekä kaikki authpriv-facilityn viestit. Daemon-facility sisältää Bind-nimipalvelun viestit ja authpriv-facility palvelimen autentikaatioviestit. Lisäksi haluttiin kerätä kaikki alert-vakavuustason ja vakavemmat viestit.

Yleisten Syslog-kenttien parsinta suoritetaan ylempänä määritetyssä 10_linux_syslog.conf-tiedostossa, joten nimipalvelinten sääntötiedostoon tarvitsee asettaa suodatussäännöt ainoastaan Bind-nimipalvelun viesteille.

Ensimmäisenä määritettiin suodatin query-tyypin viesteille, eli asiakaslaitteiden tekemille nimikyselyille. Query-viesteistä parsittiin asiakaslaitteen IP-osoite ja portti sekä kyseltävä domain-nimi. Viesteihin lisättiin tyyppikenttä kertomaan kyseessä olevan query-viesti.

```
grok {
    match => [ "message", "%{DATA}: queries: %{WORD}: client
%{IP:dns_client_ip}#%{INT:dns_client_port}: query: %{DATA:dns_target_domain}
%{GREEDYDATA}" ]
    add_field => [ "dns_type", "query" ]
}
```

Error-tyypin viesteille asetettiin grok-suodatin, joka lisää viestiin tyyppikentän:

```
grok {
    match => [ "message", "%{DATA}: error %{GREEDYDATA}" ]
```

```

    add_field => [ "dns_type", "error" ]
  }

```

Zone-tyyppin viesteille asetettiin grok-suodatin, joka parsii zonen-nimen ja lisää viestiin tyyppikentän:

```

grok {
    match => [ "message", "%{DATA}: zone %{DATA:dns_zone}:
%{GREEDYDATA}" ]
    add_field => [ "dns_type", "zone" ]
  }

```

Loppuun lisättiin vielä mutate-suodatin:

```

mutate {
    remove_field => [ "tags", "@version" ]
  }

```

Esimerkki parsitusta query-tyyppin lokiviestistä:

```

{"message":"<30>Oct 26 12:10:45 xx named[3437]: queries: info: client
192.168.x.x#60954: query: x.x.168.192.in-addr.arpa IN PTR + (195.x.x.x)",
"@timestamp":"2013-10-26T09:10:52.642Z",
"host":"195.x.x.x",
"syslog_pri":"30",
"syslog_severity_code":6"
,"syslog_facility_code":3"
,"syslog_facility":"daemon",
"syslog_severity":"informational"
,"hostname":"xxx",
"program":"named",
"pid":"3437",
"dns_client_ip":"192.168.x.x",
"dns_client_port":"60954",
"dns_target_domain":"x.x.168.192.in-addr.arpa",
"dns_type":"query"}

```

13.11 NTP-palvelin

Lokitapahtumien analysoinnin kannalta on tärkeää että kaikki lokilähteet ovat samassa ajassa. LabraNet-verkon laitteet synkronoivat kellonsa NTP-aikapalvelimen kanssa, jonka käyttöjärjestelmänä toimii CentOS ja lokienhallinnasta vastaa Rsyslog.

Rsyslog-asetukset määritettiin samoin kuin nimipalvelinten kohdalla:

```
daemon.info;authpriv.info; *.alert @192.168.x.x:514
```

NTP-palvelimen viesteille ei tarvitse määrittää erillisiä suodatussääntöjä, vaan voidaan käyttää pelkästään 10_linux_syslog.conf-tiedoston sääntöjä.

Esimerkki NTP-palvelimen synkronointiviestistä:

```
{"message":"<30>Nov 10 14:05:13 ntp ntpd[6963]: synchronized to  
194.100.2.198, stratum 1",  
"@timestamp":"2013-11-10T12:05:21.991Z",  
"host":"192.168.x.x",  
"syslog_pri":"30",  
"hostname":"ntp",  
"program":"ntpd",  
"pid":"6963",  
"syslog_severity_code":6,  
"syslog_facility_code":3,  
"syslog_facility":"daemon",  
"syslog_severity":"informational"}
```

13.12 RADIUS-palvelin

RADIUS-palvelinta käytetään verkkolaitteille tapahtuvassa käyttäjien autentikoinnissa. Palvelimen käyttöjärjestelmänä toimii CentOS ja lokienhallinnasta vastaa Rsyslog.

Freeradius-palvelun asetuksiin määritettiin lokiviestien kohteeksi Syslog, facility-arvoksi daemon ja asetettiin autentikaatio pyynnöt kirjoitettavaksi lokiin.

```
destination = syslog  
syslog_facility = daemon  
auth = yes
```

Rsyslog asetuksiin määritettiin daemon-facilityn notice-vakavuustason, authpriv-facilityn info-vakavuustason ja alert-vakavuustason viestit lähetettäväksi keräyspalvelimelle.

```
daemon.notice;authpriv.info; *.alert @192.168.x.x:514
```

RADIUS-palvelimen suodatussäännöt määritettiin 17_radius.conf-tiedostoon. Onnistuneille autentikaatiotapahtumille luotiin suodatin, joka parsii viesteistä käyttäjänimen, kohdelaitteen IP-osoitteen ja käyttäjän IP-osoitteen. Lisäksi viesteihin asetettiin tyyppikenttä.

```
grok {
    match => [ "message", "%{DATA}: Login OK:
[%{WORD:radius_user}] \{(from client %{IP:radius_client_ip} port %{INT} cli
%{IP:radius_user_ip})" ]

    add_field => [ "radius_type", "login_ok" ]
}
```

Epäonnistuneille autentikaatiotapahtumille luotiin vastaava suodatin.

```
grok {
    match => [ "message", "%{DATA}: Login incorrect %{DATA}:
[%{WORD:radius_user}] \{(from client %{IP:radius_client_ip} port %{INT} cli
%{IP:radius_user_ip})" ]

    add_field => [ "radius_type", "login_incorrect" ]
}
```

Loppuun lisättiin mutate -suodatin:

```
mutate {
    remove_field => [ "tags", "@version" ]
}
```

Esimerkki onnistuneen autentikoinnin tuottamasta viestistä:

```
{"message": "<29>Oct 26 13:26:54 radius radiusd[9335]: Login OK: [test] (from
client 192.168.x.x port 0 cli 10.x.x.x)",
"@timestamp": "2013-10-26T10:27:00.224Z",
"host": "192.168.x.x",
"syslog_pri": "29",
"syslog_severity_code": 5,
"syslog_facility_code": 3,
"syslog_facility": "daemon",
"syslog_severity": "notice",
"hostname": "radius",
"program": "radiusd",
"pid": "9335",
"radius_user": "test",
"radius_client_ip": "192.168.x.x",
```

```
"radius_user_ip":"10.x.x.x",
"radius_type":"radius_login_ok"}
```

13.13 Sähköpostipalvelimet

LabraNet-ympäristö sisältää kaksi sähköpostipalvelinta. Kyseisiltä palvelimilta haettiin kerätä autentikaatioviestit, jotta palvelimille pääsyä voitaisiin seurata.

Molempien palvelimien rsyslog.conf -tiedostoon asetettiin kaikki authpriv-facilityn ja alert-vakavuustason viestit lähetettäväksi keräyspalvelimelle:

```
authpriv.*; *.alert @192.168.x.x:514
```

Kuten NTP-palvelimen yhteydessä, sähköpostipalvelimille ei määritetty erillisiä suodatussääntöjä vaan käytettiin 10_linux_syslog.conf-tiedoston sääntöjä.

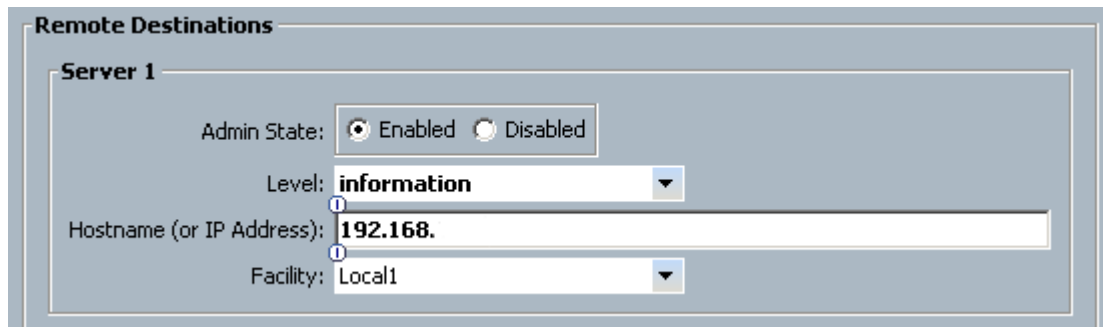
Esimerkki sähköpostipalvelimen autentikaatioviestistä:

```
{"message":"<86>Nov 3 13:38:48 mail sshd[25610]: Accepted password for test
fr
om 10.x.x.x port 52549 ssh2",
"@timestamp":"2013-11-03T11:38:57.663Z",
"host":"195.x.x.x",
"syslog_pri":"86",
"hostname":"mail",
"program":"sshd",
"pid":"25610",
"syslog_severity_code":6,
"syslog_facility_code":10,
"syslog_facility":"security/authorization",
"syslog_severity":"informational"}
```

13.14 Cisco UCS

Cisco UCS -bladepalvelinta käytetään VMware ESXi -palvelinten ajamiseen. Sen asetuksiin määritettiin lokitapahtumat lähetettäväksi Syslog-protokollalla Collector2-keräyspalvelimelle.

Kuviossa 30 nähdään UCS-palvelimelle määritetyt Syslog-asetukset.



Kuvio 30. Cisco UCS Syslog-asetukset

UCS-palvelimen tuottamien lokiviestien formaatti on sama kuin Cisco kytkinten, joten suodatukseen höydynnettiin aiemmin luotua suodatinta, jonka ehtolauseeseen lisättiin UCS:n IP-osoitteet:

```
if [host] in "192.168.x.x, 192.168.x.x, 192.168.x.x" {
```

Esimerkki UCS-palvelimen lokiviestistä:

```
{"message": "<142>: 2013 Oct 26 12:53:19 EET: %UCSM-6-AUDIT:
[session][internal][creation][ ] Web A: remote user ucs-XXX\test logged in from
10.x.x.x",
"@timestamp": "2013-10-26T10:53:25.621Z",
"host": "192.168.x.x",
"syslog_pri": "142",
"cisco_facility": "UCSM",
"cisco_severity": "6",
"cisco_msg_type": "AUDIT",
"syslog_severity_code": 6,
"syslog_facility_code": 17,
"syslog_facility": "local1",
"syslog_severity": "informational"}
```

13.15 Windows-palvelimet

LabraNet-ympäristö sisältää useita Windows-palvelimia työasemien hallintaan ja verkko-osoitteiden jakamiseen. Palvelimista kaksi ylläpitää Active Directory -hakemistopalvelua, joka sisältää Labranet-toimialueen käyttäjätietokannan. Toiset kaksi palvelinta jakavat verkko-osoitteet DHCP-palvelun avulla työasemille ja sisältävät käyttäjien verkkojaot. Lisäksi erillinen VPN-palvelin hallinnoi verkon etäyhteyksiä.

Palvelimilta haluttiin kerätä Active Directoryn kautta tapahtuvat käyttäjien onnistuneet ja epäonnistuneet kirjautumiset sekä uloskirjautumiset. Windows ei tue Event log -viestien lähettämistä etäpalvelimelle Syslog-protokollan avulla, joten tarkoitusta varten palvelimille asennettiin Nxlog-sovellus.

Nxlog on lokiviestien keräämiseen ja edelleen lähetykseen tarkoitettu sovellus, joka on saatavilla Windows-, Linux-, BSD- ja Android-käyttöjärjestelmille. Se tukee useita eri viestiformaatteja, kuten Syslog, CSV, XML, GELF ja Event log. Viestejä on myös mahdollista parsia ja suodattaa. (About 2013c.)

Nxlog valittiin työhön sen monipuolisten ominaisuuksien ja laajan formaattituen perusteella. Erityisen tärkeänä pidettiin tukea Event logille ja viestien suodatusta Xpath-hakukielen avulla.

Sovellus asennettiin AD (Active Directory)- ja VPN-palvelimille. Asetustiedosto nxlog.conf sijaitsee Program Files (x86)/Nxlog/conf -kansion alla. Tiedostossa on ensimmäisenä määritetty sovelluksen käyttämät tiedostopolut.

```
define ROOT C:\Program Files (x86)\nxlog
```

```
Moduledir %ROOT%\modules  
CacheDir %ROOT%\data  
Pidfile %ROOT%\data\nxlog.pid  
SpoolDir %ROOT%\data  
LogFile %ROOT%\data\nxlog.log
```

Seuraavana asetettiin GELF-moduuli käyttöön. Viestien lähetykseen keräyspalvelimelle käytettiin GELF-formaattia, koska Nxlog pystyy erottelemaan Event login sisällön kenttiin, ilman että erottelua tarvitsee erikseen suorittaa parsintapalvelimella.

```
<Extension gelf>  
  Module xm_gelf  
</Extension>
```


Seuraavaksi asetettiin sisääntulo eli mistä sovellus lukee viestit. Moduuliksi asetettiin `im_msvistalog`, jota käytetään Event Logien lukemiseen Windows Vista/Server 2008 ja uudemmissa Windows-käyttöjärjestelmistä. `SavePos`- ja `ReadFromLast`-parametrit määrittävät sovelluksen keräämään vain sen käynnistyksen jälkeen julkaistut viestit. Näin estetään kaikkien Event Login sisältämien viestien lähetys kerralla sovelluksen käynnistettäessä.

Query määrittää Xpath-lausekkeen, jonka perusteella keräyspalvelimelle lähetettävät viestit valitaan. Hakulausekkeeseen asetettiin Security-kanavan viestit, joiden EventID-arvo on 4624, 4634 tai 4771.

<Input in>

```
Module      im_msvistalog
SavePos     FALSE
ReadFromLast TRUE
```

```
Query <QueryList>\
      <Query Id="0">\
        <Select Path="Security">*[System[(EventID=4624 or
EventID=4634 or EventID=4771)]]</Select>\
      </Query>\
    </QueryList>
```

</Input>

EventID 4624 (An account was successfully logged on) tapahtuma syntyy käyttäjän kirjautuessa onnistuneesti toimialueeseen. EventID 4634 (An account was logged off) tapahtuma syntyy käyttäjän kirjautuessa ulos toimialueesta. EventID 4771 (Kerberos pre-authentication failed) tapahtuma syntyy käyttäjän kirjautumisen epäonnistuuessa.

Ouput eli ulostulo määrittää minne kerätyt viestit ohjataan. Viestit asetettiin lähetettäväksi GELF-tyyppisinä Collector2-keräyspalvelimelle porttiin 12201.

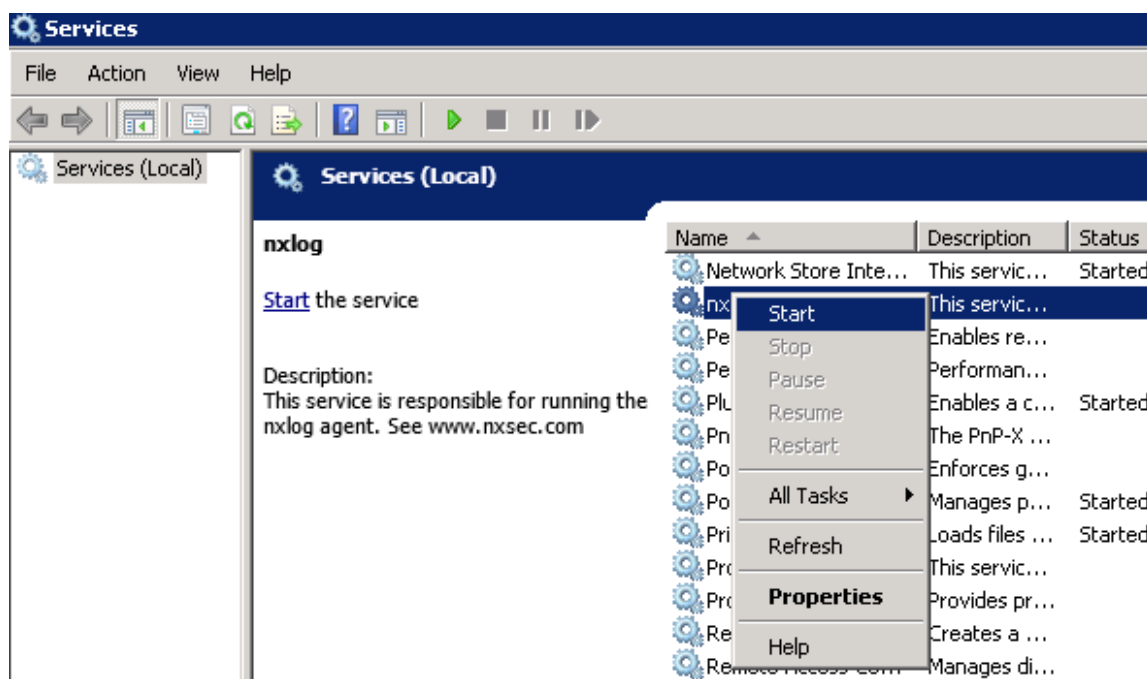
<Output out>

```
Module      om_udp
Host        192.168.x.x
Port        12201
OutputType  GELF
</Output>
```

Lopuksi asetettiin route-määrittely, joka määrää mitä moduuleja viestien käsittelyyn käytetään ja missä järjestyksessä. Tässä tapauksessa käsittelyyn käytetään ainoastaan input- ja ouput-moduuleja.

```
<Route 1>
  Path    in => out
</Route>
```

Kuviossa 31 nähdään, kuinka Nxlog-palvelu käynnistetään services-valikosta.



Kuvio 31. Nxlog-palvelun käynnistys

Windows-palvelimien suodatussäännöt määritettiin 19_windows.conf-tiedostoon. Sääntöihin asetettiin ainoastaan mutata-suodatin, jolla viesteistä karsitaan ylimääräiset kentät pois.

```
mutate {
    remove_field => [ "version", "@version", "tags", "Keywords", "Version",
    "Task", "ProcessID", "ThreadID", "Channel", "OpcodeValue", "SubjectUserSid",
    "SubjectUserName", "SubjectDomainName", "SubjectLogonId",
    "LogonProcessName", "LogonGuid", "TransmittedServices", "LmPackageName",
    "AuthenticationPackageName", "KeyLength", "ProcessName", "IpPort",
    "SourceModuleName", "SourceModuleType" ]
}
```

Esimerkki parsitusta EventID 4624 -tyypin viestistä:

```
{
  "host": "192.168.x.x",
  "short_message": "An account was successfully logged on.",
  "Subject": "Security ID",
  "level": 6,
  "facility": "Microsoft-Windows-Security-Auditing",
  "@timestamp": "2013-10-29T14:36:45.000Z",
  "message": "An account was successfully logged on. Subject: Security ID: \\xxx Account Name: \\- Account Domain: \\- Logon ID: \\0 Logon Type: \\3 New Logon: Security ID: \\XXX Account Name: \\XXX$ Account Domain: \\LABRANET Logon ID: \\0xedxx Logon GUID: \\{XXX} Process Information: Process ID: \\0 r Process Name: \\- Network Information: Workstation Name: \\ Source Network Address: \\192.168.xx.xx Source Port: \\5816 Detailed Authentication Information: Logon Process: \\Kerberos Authentication Package: \\Kerberos Transited Services: \\- Package Name (NTLM only): \\- Key Length: \\0 This event is generated when a logon session is created. It is generated on the computer that was accessed. The subject fields indicate the account on the local system which requested the logon. This is most commonly a service such as the Server service, or a local process such as Winlogon.exe or Services.exe. The logon type field indicates the kind of logon that occurred. The most common types are 2 (interactive) and 3 (network). The New Logon fields indicate the account for whom the new logon was created, i.e. the account that was logged on. The network fields indicate where a remote logon request originated. Workstation name is not always available and may be left blank in some cases. The authentication information fields provide detailed information about this specific logon request. - Logon GUID is a unique identifier that can be used to correlate this event with a KDC event. - Transited services indicate which intermediate services have participated in this logon request. - Package name indicates which sub-protocol was used among the NTLM protocols. - Key length indicates the length of the generated session key. This will be 0 if no session key was requested.",
  "EventType": "AUDIT_SUCCESS",
  "SeverityValue": 2,
  "Severity": "INFO",
  "EventID": 4624,
  "SourceName": "Microsoft-Windows-Security-Auditing",
  "ProviderGuid": "{xxx}",
  "RecordNumber": xxx,
  "Category": "Logon",
  "Opcode": "Info",
  "TargetUserSid": "xxx",
}
```

```
"TargetUserName":"xxx$",
"TargetDomainName":"LABRANET",
"TargetLogonId":"xxx",
"LogonType":"x",
"AuthenticationPackageName":"Kerberos",
"KeyLength":"0",
"IpAddress":"192.168.x.x",
"EventReceivedTime":"2013-10-29 16:36:46"}
```

Kuviosta 32 nähdään, kuinka Graylog2-verkkokäyttöliittymä esittää viestin huomattavasti selkeämmässä muodossa:

Message 50200c50-423e-11e3-a8a0-0050568535cb

An account was successfully logged on. Subject: Security ID

In which terms was this message broken to?

From: 192.168.

Date: 2013-10-31 17:08:28 +0200

Level: Info

Facility: logstash-gelf

File: %{path}:-1

facility: Microsoft-Windows-Security-Auditing

EventType: AUDIT_SUCCESS

EventID: 4624

level: 6

TargetDomainName: LABRANET

EventReceivedTime: 2013-10-31 17:08:29

ProviderGuid:

Category: Logon

TargetUserSid:

host: 192.168.

SeverityValue: 2

Opcode: Info

Severity: INFO

RecordNumber:

TargetUserName:

LogonType:

TargetLogonId:

SourceName: Microsoft-Windows-Security-Auditing

IpAddress: 192.168.

Full message:

An account was successfully logged on.

Subject:

Security ID:

Account Name: -

Account Domain: -

Logon ID: 0x0

Kuvio 32. Event log -viesti Graylog2-verkkokäyttöliittymässä

13.16 LDAP-palvelin

LDAP (Lightweight Directory Access Protocol) on hakemistopalvelujen verkon yli tapahtuvaan käyttöön kehitetty protokolla. LabraNet-ympäristön useat palvelut käyttävät protokollaa käyttäjien tunnistamiseen. Verkon LDAP-tietokantapalvelin asetettiin tallentamaan tietokantaan tehdyt haut ja muutokset lokiin.

Palvelimen käyttöjärjestelmänä toimii CentOS ja lokienhallinnasta vastaa Rsyslog. LDAP-tietokantaa hallitseva Slapd-prosessi tallentaa tapahtumat lokiin local4-facility arvolla, joten Rsyslog asetettiin lähettämään local4-facility lokiviestit keräyspalvelimelle. Lisäksi asetettiin authpriv-facilityn ja alert-vakavuustason viestit, kuten aiemmin.

```
authpriv.info;local4.*; *.alert @192.168.x.x:514
```

Rsyslog asetuksiin lisättiin vastaanotettavien viestien määrää rajoittavat parametrit, jotka estävät esimerkiksi vikaantuneen palvelun tuottaman valtavan viestimäärän täyttämästä palvelimen tallennustilaa.

```
$SystemLogRateLimitInterval 2
$SystemLogRateLimitBurst 200
```

LDAP-palvelimen suodatussäännöt määritettiin 18_ldap.conf-tiedostoon. Viestejä varten määritettiin grok-suodattimet, jotka poimivat LDAP-viesteistä connection-, operation-, file descriptor-, error- ja tag -numerot sekä tapahtuman tyyppin. Näiden kenttien arvoja voidaan käyttää hyväksi LDAP-operaatioiden seuraamiseen.

```
grok {
    match => [ "message", "%{DATA}:
conn=%{INT:ldap_connection_num} op=%{INT:ldap_operation_num}
%{WORD:ldap_type}" ]
}
```

```
grok {
    match => [ "message", "%{DATA} fd=%{INT:ldap_file_description}
%{WORD:ldap_type}" ]
}
```

```
grok {
    match => [ "message", "%{DATA} err=%{INT:ldap_error_num}" ]
}
```

```
grok {
    match => [ "message", "%{DATA} tag=%{INT:ldap_tag_num}" ]
}
```

Loppuun lisättiin mutate-suodatin:

```
mutate {
    remove_field => [ "tags", "@version" ]
}
```

Esimerkki LDAP-lokiviestistä:

```
{"message": "<167>Nov 6 17:40:44 ldap slapd[19766]: conn=1940 op=2 SRCH
attr=cn description entryUUID",
"@timestamp": "2013-1106T15:40:56.408Z",
"host": "192.168.x.x",
"syslog_pri": "167",
"hostname": "ldap",
"program": "slapd",
"pid": "19766",
"ldap_connection_num": "1940",
"ldap_operation_num": "2",
"ldap_type": "SRCH",
"syslog_severity_code": 7,
"syslog_facility_code": 20,
"syslog_facility": "local4",
"syslog_severity": "debug"}
```

13.17 Student-palvelin

LabraNet-ympäristön Student-palvelin tarjoaa ICT-koulutusohjelmien opiskelijoille verkkosivutilaa ja muita yleisiä UNIX/Linux-ympäristön palveluja. Palvelimelta ha-
luttiin kerätä autentikaatioviestit käyttäjien kirjautumisista.

Rsyslog asetukseen määritettiin authpriv-facilityn info-vakavuustason ja kaikki alert-
tason viestit lähetettäväksi etäpalvelimelle:

```
authpriv.info;*.alert @192.168.x.x:514
```

Student-palvelimelle ei määritetty omaa sääntötiedostoa vaan käytettiin
10_linux_syslog.conf-tiedoston sääntöjä.

Esimerkki Student-palvelimen autentikaatioviestistä:

```
{ "message": "<86>Nov 17 12:43:08 student sshd[6031]: pam_ sss(sshd:auth):
authentication success; logname= uid=x euid=x tty=ssh ruser= rhost=1
0.x.x.x user=test",
"@timestamp": "2013-1117T10:43:15.696Z",
"host": "195.x.x.x",
"syslog_pri": "86",
"hostname": "student",
"program": "sshd",
"pid": "6031",
"pam_auth_type": "success",
"pam_rhost": "10.x.x.x",
"pam_source_user": "test",
"syslog_severity_code": 6,
"syslog_facility_code": 10,
"syslog_facility": "security/authorization",
"syslog_severity": "informational" }
```

13.18 Lokipalvelimet

Myös itse työssä käytetyt lokipalvelimet asetettiin lähettämään lokinsa järjestelmään. Lokipalvelimien kiinnostavia tapahtumia ovat lähinnä autentikaatioviestit ja korkean vakavuustason viestit.

Palvelinten Rsyslog asetuksiin määritettiin authpriv-facilityn info-vakavuustason ja kaikki alert-tason viestit lähetettäväksi etäpalvelimelle:

```
authpriv.info:*alert @192.168.x.x:514
```

Palvelinten viestien parsimiseen käytettiin jälleen 10_linux_syslog.conf-tiedoston sääntöjä.

Esimerkki log-storage -palvelimen autentikaatioviestistä:

```
{ "message": "<85>Nov 19 16:45:20 log-storage sshd[4079]: pam_unix(sshd:auth):
authentication failure; logname= uid=x euid=x tty=ssh ruser= rhost=10.x.x.x
user=test",
"@timestamp": "2013-11-19T14:45:23.628Z",
"host": "192.168.x.x",
"syslog_pri": "85",
"hostname": "log-storage",
"program": "sshd",
"pid": "4079",
"pam_auth_type": "failure",
"pam_rhost": "10.x.x.x ",
"pam_source_user": "test",
```

```
"syslog_severity_code":5,
"syslog_facility_code":10,
"syslog_facility":"security/authorization",
"syslog_severity":"notice"}
```

13.19 Graylog2-verkkokäyttöliittymä

Graylog2-verkkokäyttöliittymän messages-pääsivu näyttää yleisnäkymän uusimista vastaanotetuista viesteistä. Oikeanpuoleinen paneeli sisältää viestit ja vasemmanpuoleinen graafin parin viime tunnin aikana vastaanotettujen viestien määrästä. Vasemmassa paneelissa on myös linkki server health -sivulle, joka sisältää tiedot palvelimen tilasta.

Kuviossa 33 on esitetty Graylog2-verkkokäyttöliittymän messages-sivu.

Date	Host	Level	Facility	Message
2013-11-12 20:10:21.364	195.	Info	logstash-gelf	<30>Nov 12 20:10:08 named[5442]: queries: info: client 195. #39302: query: jamk.fi IN A + (195.)
2013-11-12 20:10:21.358	195.	Info	logstash-gelf	<30>Nov 12 20:10:08 named[5442]: queries: info: client 195. #42339: query: labranet.jamk.fi IN AAAA + (195.)
2013-11-12 20:10:20.960	195.	Info	logstash-gelf	<30>Nov 12 20:10:07 named[5442]: queries: info: client 192.168. #52221: query: p5-hjctnw3in3cgw-shr64mtyfb5rvkyu-723046-i1-v6exp3-ds.metric.gs ...
2013-11-12 20:10:20.960	195.	Info	logstash-gelf	<30>Nov 12 20:10:07 named[5442]: queries: info: client 192.168. #52352: query: p5-hjctnw3in3cgw-shr64mtyfb5rvkyu-723046-i2-v6exp3-v4.metric.gs ...

Kuvio 33. Graylog2 messages-sivu

Overview-paneelista valittaessa haluttu viesti, avautuvat sen tiedot oikeaan paneeliin. Paneeli sisältää viestin sisällön, kentät ja mihin viestivirtoihin se kuuluu. Kentistä From, Date, Level, Facility ja File ovat Graylog2:n viestille määrittämiä ja loput Logstash-sovelluksen parsinnan tuloksena saatuja. Kentän arvon valitsemalla nähdään kaikki kyseisen arvon sisältävät viestit.

Kuviossa 34 nähdään DNS query -loki viestin tiedot.

Message d8464f10-4bbf-11e3-86f3-0050568535cb ^x

```
<30>Nov 12 19:28:17      named[25677]: queries: info: client
192.168.      #64894: query: wns.notify.windows.com.akadns.net IN A +
(195.      )
```

In which terms was this message broken to?

```
From: 195.
Date: 2013-11-12 19:28:30 +0200
Level: Info
Facility: logstash-gelf
File: %{path}:-1
syslog_pri: 30
dns_client_port: 64894
pid: 25677
syslog_facility: daemon
dns_target_domain: wns.notify.windows.com.akadns.net
syslog_facility_code: 3
syslog_severity_code: 6
syslog_severity: informational
dns_client_ip: 192.168.
hostname:
host: 195.
program: named
dns_type: query
```

Full message:

```
<30>Nov 12 19:28:17      named[25677]: queries: info: client
192.168.      #64894: query: wns.notify.windows.com.akadns.net IN A +
(195.      )
```

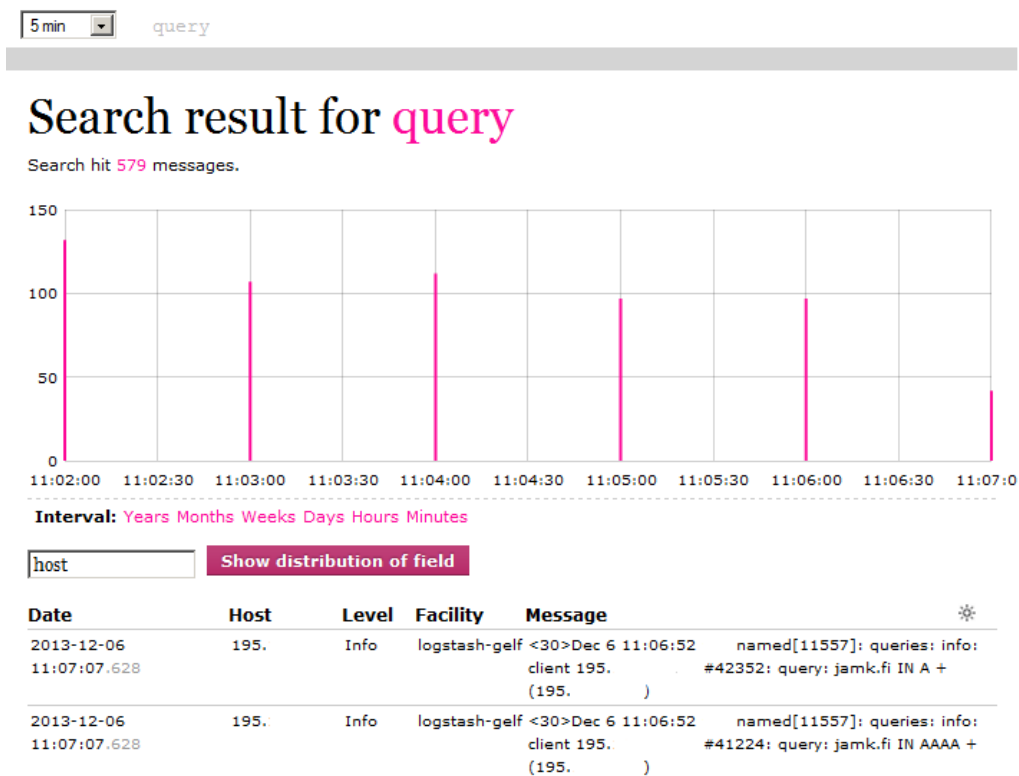
Filed in following streams:

- All_messages
- DNS_query

Kuvio 34. Graylog2 viestin tiedot

Viestejä voidaan hakea verkkokäyttöliittymässä hakupalkin tai quickfilter-suodattimen avulla. Hakupalkki mahdollistaa vapaan haun viestin sisällöstä. Esimerkiksi haettaessa hakusanalla ”query”, saadaan tuloksena kaikki kyseisen sanan sisältävät viestit. Hakupalkkiin voidaan myös asettaa aikaväli, jolta viestit haetaan.

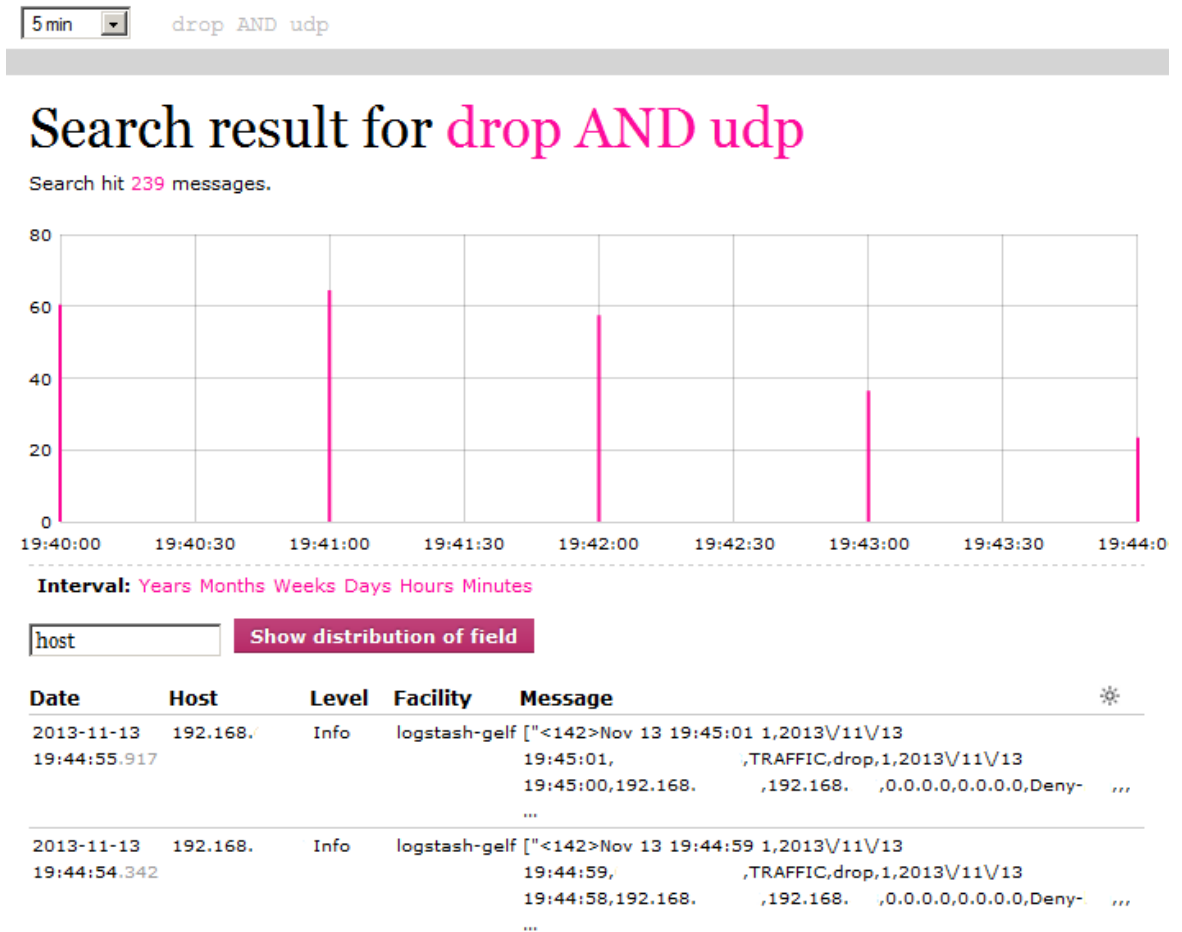
Kuviossa 35 on haettu kaikki viestit viimeisen viiden minuutin ajalta, joissa esiintyy sana ”query”.



Kuvio 35. Graylog2 hakuesimerkki 1

Hakukenttä tukee kahdenlaisia jokerimerkkejä: ? täsmää mihin tahansa yhteen merkkiin ja * yhteen tai useampaan merkkiin. Hauissa voi myös käyttää totuusarvomuuttujia AND (&&), OR (||) ja NOT (!). Plus (+) ja miinus (-) -merkeillä voidaan määrittää kuuluuko tietty sana tulokseen, vai ei.

Kuviossa 36 haetaan kaikki viestit viimeisen 5 minuutin ajalta, joissa esiintyy sana "drop" ja "udp":



Kuvio 36. Graylog2 hakuesimerkki 2.

Hakuun täsmävien viestien lisäksi verkkokäyttöliittymä esittää viestien määrästä kuvion, jonka aika-akselin välin voi asettaa minuuteista vuosiin.

Quickfilter-suodatin mahdollistaa hakukenttää tarkempien ehtojen määrittämisen. Se sallii viestien suodatuksen sisällön lisäksi tarkan aikavälin, facility-arvon, tiedostonimen, rivinumeron, vakavuustason, lähteen ja käyttäjän määrittämisen kenttien perusteella.

Hakuehdoista hyödyllisin on Additional field, jolla pystytään hakemaan viestit mikä tahansa kentän arvon perusteella. Kenttiä voidaan lisätä hakuun haluttu määrä. Tällä ominaisuudella pystytään tehokkaasti hyödyntämään viestien parsinnassa tehtyjen kenttien erottelu.

Kuviossa 37 haetaan kaikki viestit, joissa dns_type-kentän arvo on query ja dns_target_domain-kentän arvo on labranet.jamk.fi:

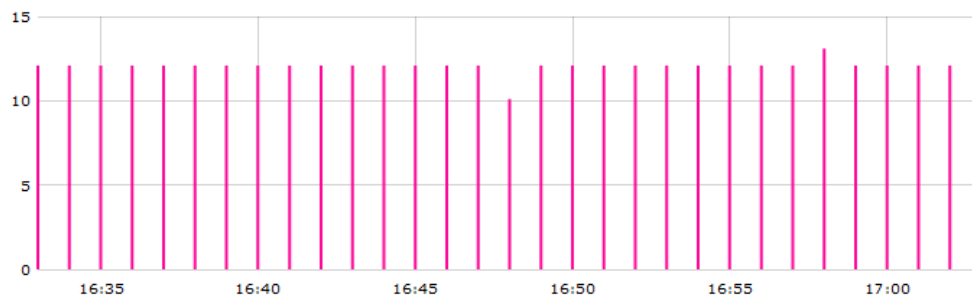
Overview

Currently containing **9.534.143** messages.

Message:
 Full Message:
 Timeframe:
 Facility:
 File:
 Line number:
 Level: or higher
 Host:
 dns_type:
 dns_target_domain:

Add additional field **Run filter**

Quickfilter hit **361** messages.



Interval: [Years](#) [Months](#) [Weeks](#) [Days](#) [Hours](#) [Minutes](#)

Show distribution of field

Date	Host	Level	Facility	Message	⚙
2013-11-14 17:03:01.839	195.	Info	logstash-gelf	<30>Nov 14 17:02:47 : named[5442]: queries: info: client 195. #50220: query: labranet.jamk.fi IN A + (195. .)	
2013-11-14 17:03:01.838	195.	Info	logstash-gelf	<30>Nov 14 17:02:47 : named[5442]: queries: info: client 195. #45700: query: labranet.jamk.fi IN AAAA + (195. .)	



Kuvio 37. Graylog2 quickfilter-esimerkki 1

Haetuista viesteistä voidaan tarkastella tietyn kentän arvojen jakaumaa valitsemalla “Show distribution of field”. Tätä ominaisuutta voidaan hyödyntää erilaisten trendien seuraamiseen, kuten mihin IP-osoitteeseen avataan eniten yhteyksiä.

Kuviossa 38 haetaan kaikki palomuurin lokiviestit 14.11.2013 08:00 – 14:00 väliseltä ajalta ja esitetään action-kentän arvojen jakauma.

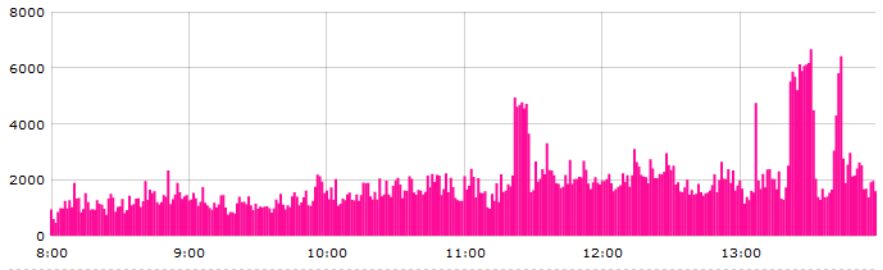
Overview

Currently containing 9,508,331 messages.

Message: 
 Full Message:
 Timeframe: from 2013-11-14 08:00:00 to 2013-11-14 : 
 Facility:
 File:
 Line number:
 Level: or higher
 Host: 192.168.

[Add additional field](#) [Run filter](#)

Quickfilter hit 667,697 messages.



Interval: [Years](#) [Months](#) [Weeks](#) [Days](#) [Hours](#) [Minutes](#)

[Show distribution of field](#)

Count	Term
594078	allow
66105	deny
2558	alert
1764	block-url
9	drop-all-packets
2	forward
1	reset-both

Kuvio 38. Graylog2 quickfilter-esimerkki 2

Verkkokäyttöliittymä sisältää myös erillisen hosts-sivun, jonka kautta voidaan nopeasti suodattaa vain tietystä lähteestä vastaanotetut lokiviestit.

Kuviossa 39 nähdään Graylog2-verkkoliittymän hosts-sivu.

Hosts

Quick jump to host: [Go!](#)

Monitoring 20 hosts.



Kuvio 39. Graylog2 hosts

13.20 Graylog2 streams

Graylog2-verkkokäyttöliittymän streams-valikkoon määritettiin viestivirrat, joiden kautta ylläpitäjät pääsevät helposti tarkastelemaan ympäristön toiminnan kannalta oleellisia viestejä. Viestivirrat myös mahdollistavat automaattisten hälytysten asetuksen.

Viestivirrat jaettiin kategorioihin lokilähteiden (esim. palomuuuri, kytkimet) ja vakaavuustason (esim. error, critical) perusteella. Kategorioiden sisällä virrat määritettiin tapahtuman tyyppin perusteella (esim. traffic, threat).

Seuraavassa on lueteltu asetetut kategoriat ja viestivirrat.

- Severity
 - o Emergency
 - o Critical
 - o Alert
 - o Error
- Firewall
 - o All_firewall
 - o Traffic
 - o Threat
 - o Config
 - o System
- Switches
 - o All_switches
 - o Input commands
- DNS
 - o DNS_query
 - o DNS_error
 - o DNS_zone
- AD
 - o Account login
 - o Account login failed
 - o Account logout
- Radius
 - o All_radius
 - o Login OK
 - o Login incorrect
- LDAP
 - o All_ldap
 - o Invalid credentials
- UCS
 - o All_UCS
 - o Audit

- Authentication
 - o All authentication messages

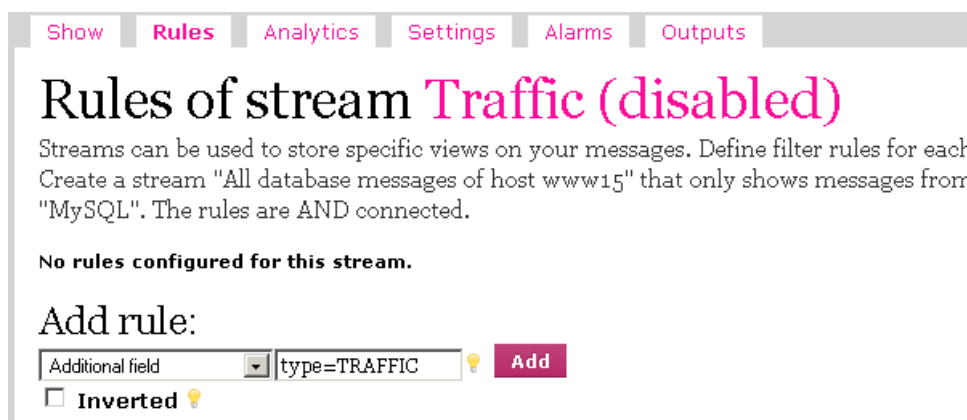
Uusi kategoria luodaan streams-sivulta valitsemalla Manage Categories, jonka jälkeen asetetaan kategorian nimi. Kuviossa 40 nähdään esimerkki Firewall-kategorian luonnista.



Kuvio 40. Graylog2 stream-kategorian luonti

Viestivirrat luodaan streams-sivulla määrittämällä virralle nimi ja valitsemalla Create stream. Seuraavaksi määritetään säännöt, joiden perusteella viestit asetetaan kyseiseen virtaan. Säännöt voivat perustua esimerkiksi viestin sisältöön, lähteosoitteeseen, facility- ja severity-arvoihin, tai muiden kenttien arvoihin.


Kuviossa 41 nähdään, kuinka palomuurilokeja varten luotiin Traffic-virta, johon asetettiin kuulumaan viestit joiden type-kentän arvo on TRAFFIC.



Kuvio 41. Graylog2 viestivirran säännön asetus

Viestivirta voidaan asettaa haluttuun kategoriaan Settings-välilehdeltä. Virta tulee myös ottaa käyttöön poistamalla Stream disabled -valinta.

Kuviossa 42 Traffic-virta on asetetty firewall-kategoriaan ja otettu käyttöön.



[Show](#) [Rules](#) [Analytics](#) **Settings** [Alarms](#) [Outputs](#)

Settings of stream Traffic

Stream disabled

General

These settings are only applied for your user.

Favorite

Category

Category: Firewall [Set category](#)

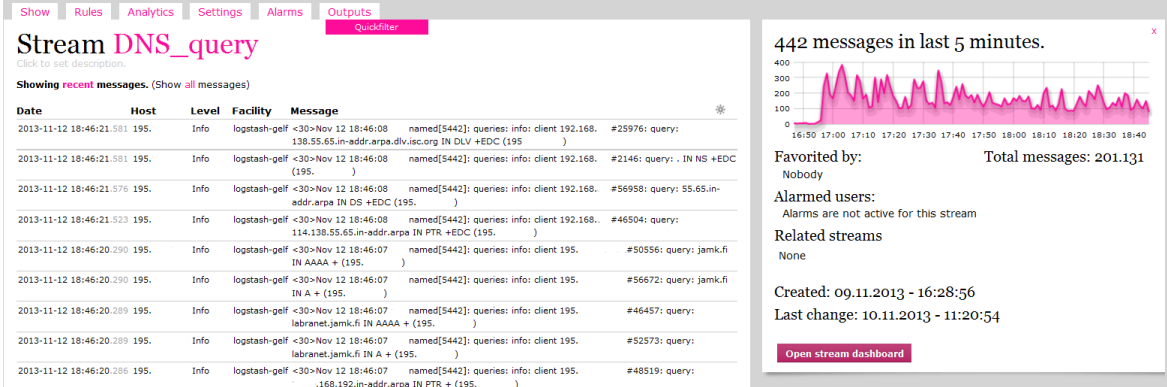
Currently in category Firewall.

Kuvio 42. Graylog2 viestivirran kategorian asetukset

Muut kategoriat ja viestivirrat määritettiin vastaavalla tavalla. Virtaan kuuluvia viestejä pääsee tarkastelemaan klikkaamalla sen nimeä streams-sivulla.

Show-välilehden kautta voidaan tarkastella ja hakea virtaan kuuluvia viestejä messages-sivun tapaan.

Kuviossa 43 nähdään DNS_query-virran viestit.



[Show](#) [Rules](#) [Analytics](#) [Settings](#) **Messages** [Alarms](#) [Outputs](#)

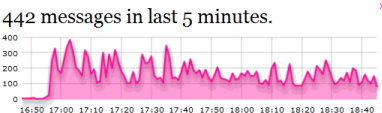
Stream DNS_query

Click to set description.

Showing recent messages. (Show all messages)

Date	Host	Level	Facility	Message
2013-11-12 18:46:21.581	195.	Info	logstash-gelf	<30>Nov 12 18:46:08 named[5442]: queries: info: client 192.168.138.55.in-addr.arpa.div.isc.org IN DLV +EDC (195.)
2013-11-12 18:46:21.581	195.	Info	logstash-gelf	<30>Nov 12 18:46:08 named[5442]: queries: info: client 192.168.195.)
2013-11-12 18:46:21.576	195.	Info	logstash-gelf	<30>Nov 12 18:46:08 named[5442]: queries: info: client 192.168.55.65.in-addr.arpa IN DS +EDC (195.)
2013-11-12 18:46:21.523	195.	Info	logstash-gelf	<30>Nov 12 18:46:07 named[5442]: queries: info: client 192.168.114.138.55.65.in-addr.arpa IN PTR +EDC (195.)
2013-11-12 18:46:20.290	195.	Info	logstash-gelf	<30>Nov 12 18:46:07 named[5442]: queries: info: client 195. IN AAAA + (195.)
2013-11-12 18:46:20.290	195.	Info	logstash-gelf	<30>Nov 12 18:46:07 named[5442]: queries: info: client 195. IN A + (195.)
2013-11-12 18:46:20.289	195.	Info	logstash-gelf	<30>Nov 12 18:46:07 named[5442]: queries: info: client 195. labranet.jamk.fi IN AAAA + (195.)
2013-11-12 18:46:20.289	195.	Info	logstash-gelf	<30>Nov 12 18:46:07 named[5442]: queries: info: client 195. labranet.jamk.fi IN A + (195.)
2013-11-12 18:46:20.286	195.	Info	logstash-gelf	<30>Nov 12 18:46:07 named[5442]: queries: info: client 195. .168.192.in-addr.arpa IN PTR + (195.)

442 messages in last 5 minutes.



Favorited by: Nobody Total messages: 201.131

Alarmed users:
Alarms are not active for this stream

Related streams
None

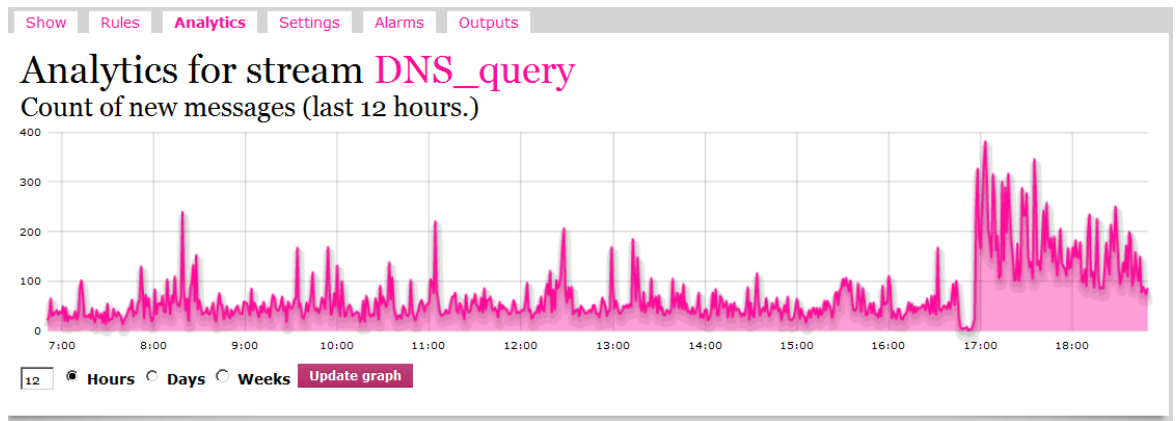
Created: 09.11.2013 - 16:28:56
Last change: 10.11.2013 - 11:20:54

[Open stream dashboard](#)

Kuvio 43. Graylog2 viestivirran viestien tarkastelu

Analytics-välilehdeltä nähdään graafi vastaanotettujen viestien määrästä halutun pituiselta ajanjaksolta. Tietyllä aikavälillä vastaanotetut viestit voidaan hakea maa-laamalla kuviosta haluttu aikaväli ja valitsemalla Show messages in range.

Kuviossa 44 nähdään graafi DNS_query-virran viesteistä viimeisen 12 tunnin ajalta.



Kuvio 44. Graylog2 viestivirran graafi

Settings-välilehdellä voidaan aiemmin mainittujen asetusten lisäksi muuttaa virran nimeä, asettaa suosikkeihin, määrittää lyhyempi nimi, määrittää siihen liittyvät muut virrat ja asettaa viesteihin lisäsarakeita. Viestivirrasta voidaan luoda kloni tai poistaa se kokonaan.

Kuviossa 45 nähdään DNS_query-virran asetukset.

Settings of stream **DNS_query**

Stream disabled

General

These settings are only applied for your user.

Favorite

Category

Category: [Set category](#)

Currently in category **DNS**.

Short name

Used to identify streams in the Analytics Shell without knowing the whole ID. Must be unique and only contain alphanumeric characters or underscores.

Short name: [Set](#)

Rename stream

New name: [Rename](#)

Related streams regex

All streams with titles that match this regular expression will be listed as related streams

Matcher (regex): [Set](#)

Additional Columns

Add additional columns to display more fields in the messages table.

■ No additional columns configured.

New column: [Add](#)

[Delete this stream](#)

[Clone this stream](#)

Kuvio 45. Graylog2 viestivirran asetukset

Alerts-välilehdeltä virralle voidaan asettaa hälytyksiä. Hälytys laukeaa, kun vastaanotettujen viestien määrä tietyllä aikavälillä ylittää asetetun raja-arvon. Grace period -arvolla voidaan määrittää hälytysten välinen viive. Hälytyksistä lähetään käyttäjille ilmoitus sähköpostin tain XMPP-protokollan välityksellä. Hälytyksiä ei vielä tässä vaiheessa otettu käyttöön.

Kuviossa 46 nähdään DNS_query-virran alerts-välilehti.

The screenshot shows the 'Alerts' tab in the Graylog2 interface for the stream 'DNS_query'. The page title is 'Alarm settings of stream DNS_query'. Below the title, there is explanatory text: 'An alarm is triggered when the number of messages in the defined timespan is higher than the maximum. All users who enabled receiving of alarms will be notified of the alarm. You can also define that all users are alarmed, no matter what they defined.' An important note states: 'Important: Alarms will be sent via built in (Email, XMPP) or plugin transports. If you want to use the built in transports, they must be enabled and configured in your graylog2.conf. Addresses of users are configured in the user settings.' There are three checkboxes: 'Active' (unchecked), 'I want to receive alarms of this stream' (unchecked), and 'Force for all users' (unchecked). Below these are input fields for 'Maximum number of messages', 'Minutes', and 'Grace period (minutes)', followed by a 'Save' button. A note explains: 'The grace period defines for how many minutes the system will wait until sending the next alert. The check runs once a minute.' The 'Alarm callbacks' section states: 'All currently installed alarm callback plugins are listed here. You can enable them for this stream if they are not enforced in the system settings.' and shows 'No alarm callbacks installed'.

Kuvio 46. Graylog2 viestivirran hälytykset

Outputs-välilehti mahdollistaa virran viestien viemisen ulkoisiin kohteisiin lisäosien avulla.

Kuviossa 47 nähdään DNS_query-virran outputs-välilehti.

The screenshot shows the 'Outputs' tab in the Graylog2 interface for the stream 'DNS_query'. The page title is 'Outputs of stream DNS_query'. Below the title, there is explanatory text: 'Every message that arrives in this stream can be written to as many outputs as you want. The Elasticsearch output is always enabled.' A note states: 'No outputs configured for this stream.' Below this is a section titled 'Add output' with the text: 'You have no output plugins installed.'

Kuvio 47. Graylog2 viestivirran ulostulot

13.21 Säilytyspalvelin

13.21.1 Kuvaus ja resurssit

Säilytyspalvelimen tehtävä on lokiviestien arkistointi vähintään KATAKRI-kriteeristön vaatiman kuuden kuukauden ajalta. Parsimattomat viestit noudetaan AMQP-vaihteesta ja tallennetaan lähteen perusteella nimettyihin tekstitiedostoihin, jotka pakataan kerran vuorokaudessa GZIP-formaatin paketteihin.

Logstash-sovellus noutaa viestit vaihteen log.storage-jonosta ja tallentaa ne loki-lähteen IP-osoitteen mukaan nimettyihin tiedostoihin. Logrotate-sovellus pakkaa tiedostot kerran vuorokaudessa ja lisää nimeen päivämäärän. Logrotate säilyttää ainoastaan 183 kappaletta yhden lokilähteen tiedostoa, eli puolen vuoden loki-viestit. Kun 183 tiedostoa tulee täyteen, vanhin tiedosto poistetaan.

Palvelinta varten luotiin "log-storage"-niminen virtuaalikone seuraavilla resursseilla:

- CPU: 2 kpl
- muisti: 4 GB
- tallennustila: 66 GB.

Tallennustila jaettiin jälleen kahteen kiintolevyyn, joista järjestelmälevylle asetettiin 16 gigatavua tallennustilaa ja datalevylle 50 gigatavua. Järjestelmälevy sisältää käyttäjärjestelmän tiedostot ja datalevy arkistoidut lokiviestit. Osiointi suoritettiin LVM-osiohallintakomponentilla.

13.21.2 Logstash asetukset

Logstash asetustiedostoon määritettiin RabbitMQ-sisääntulo viestien vaihteesta noutamiseksi. Parametreiksi asetettiin vaihteen IP-osoite ja nimi, jonon nimi, reititysavain sekä jonon kestävyys.

```
input {
  rabbitmq {
    host => "192.168.x.x"
    exchange => "Labranet-loki"
```

```

        queue => "log.storage"
        key => "log.raw"
        durable => true
        auto_delete => false
        exclusive => false
    }
}

```

Logstash asetettiin tallentamaan lokitiedostot /log_storage/-kansion alle lokilähdekohtaisesti alihakemistoihin. %{host}-parametria hyödynnettiin kansioiden ja tiedostojen nimeämisessä.

```

output {
    file {
        path => "/log_storage/%{host}/%{host}.log"
    }
}

```

Logrotate-sovelluksen asetustiedosto /etc/logrotate.conf sisältää järjestelmälaajuiset parametrit, jotka periytyvät erillisiin /etc/logrotate.d/-kansion alla sijaitseviin asetustiedostoihin. LabraNet-verkon lokiviestejä varten luotiin /etc/logrotate.d/labranet_lokit-asetustiedosto, johon määritettiin seuraavat parametrit:

```

/log_storage/*/*.log {
    daily
    missingok
    sharedscripts
    rotate 183
    compress
    dateformat -%Y-%m-%d
    postrotate
        /etc/init.d/logstash restart > /dev/null
    endscript
}

```

Ensimmäinen rivi määrittää polun käsiteltäviin tiedostoihin, joka sisältää kaikki log_storage-hakemiston ja sen alihakemistojen .log-päätteiset tiedostot. Daily-parametri määrittää tiedostot käsiteltäväksi päivittäin. Missingok-parametri ohjeistaa Logrotatea lokitiedoston puuttuessa siirtymään seuraavaan ilman virheilmoitusta. Rotate-parametri määrittää säilytettävien tiedostojen määrän. Compress-parametri määrittää käyttämään pakkausta. Dateformat-parametri asettaa pakattujen tiedostojen nimen loppuun lisättävän päivämäärän muodon. Postrotate- ja endscrip-parametrien väliin voidaan määrittää tiedostojen käsittelyn jälkeen suoritettavat komennot. Logstash asetettiin käynnistymään uudelleen, jotta se ymmärtää siirtyä tallentamaan lokiviestit uuteen tiedostoon.

Esimerkki arkistoiduista lokiviesteistä:

```
[root@log-storage ~]# ls -la /log_storage/192.168.x.x/
total 40
drwxr-xr-x. 2 root root 4096 Oct 27 03:48 .
drwxr-xr-x. 18 root root 4096 Oct 23 16:25 ..
-rw-r--r--. 1 root root 10365 Oct 27 12:03 192.168.x.x.log
-rw-r--r--. 1 root root 2710 Oct 23 03:10 192.168.x.x.log-2013-10-23.gz
-rw-r--r--. 1 root root 1876 Oct 24 03:40 192.168.x.x.log-2013-10-24.gz
-rw-r--r--. 1 root root 1663 Oct 25 03:40 192.168.x.x.log-2013-10-25.gz
-rw-r--r--. 1 root root 1723 Oct 26 03:09 192.168.x.x.log-2013-10-26.gz
-rw-r--r--. 1 root root 1718 Oct 27 03:48 192.168.x.x.log-2013-10-27.gz
```

14 Työn tulosten arviointi

14.1 Lokitallenteiden kattavuus

Lokitallenteiden kattavuutta arvioitaessa tulee kiinnittää huomiota kahteen seikkaan: kerätäänkö lokiviestejä tarpeeksi monesta lähteestä ja sisältävätkö viestit riittävästi informaatiota.

Lokiviestien lähteenä työssä käytettiin pääasiassa vaatimusmäärittelyn yhteydessä asetettuja. Lisäksi työhön otettiin mukaan myöhemmässä vaiheessa mieleen tulleita lähteitä, kuten student-palvelin. Kaikkia vaatimusmäärittelyn lähteistä ei pystytty käytännössä toteuttamaan niiden puutteellisten ominaisuuksia vuoksi (esimerkiksi osa Cisco kytkimistä). Lokijärjestelmä kattaa LabraNet-ympäristön toiminnan kannalta kriittiset kohteet: palomuurin, kytkimet, nimipalvelimet ja autentikointipalvelimet (LDAP, AD, RADIUS). Lisäksi järjestelmä sisältää useita muita päivittäin käytettyjä palvelimia, kuten Cisco UCS ja Student.

Lokilähteistä pyrittiin keräämään mahdollisimman kattavasti tietoa kuormittamatta lokijärjestelmää liikaa ja aiheuttamatta tärkeiden viestien hukkumista viestipalvelin joukkoon. Korkeimpana prioriteettina olivat audit- ja autentikaatio-tapahtumat, jotka haluttiin kerätä mahdollisuuksien mukaan kaikista kohteista. Audit-viestit sisältävät järjestelmiin kirjautumiset ja käyttäjien suorittamat toiminnot. Syslog-protokollaa käyttävistä lokilähteistä kerättiin kaikki viestit, joissa vakavuustaso on alert, error, critical tai emergency. Lisäksi monille lähteille määritettiin erilliset säännöt, joilla kerätään tietyn palvelun lokiviestit (esimerkiksi Bind-nimipalvelun viestit).

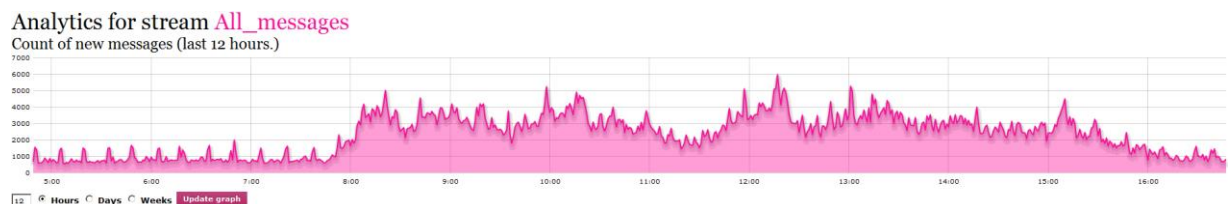
KATAKRI-kriteeristöissä asetettiin IV-tason vaatimukseksi, että tallenteiden kattavuus riittää tietomurtojen ja niiden yritysten jälkikäteiseen todentamiseen. Audit-viestit ovat erittäin tärkeitä tietomurtojen havaitsemiseen, koska niiden avulla voidaan nähdä järjestelmiin tehdyt onnistuneet ja epäonnistuneet kirjautumiset, mistä kirjautuminen on suoritettu sekä mitä toimia tunkeutuja on tehnyt. Esimerkiksi suuri määrä epäonnistuneita kirjautumisyrityksiä lyhyen ajan sisällä viittaa brute-force-hyökkäykseen. Palomuurin tuottamat lokiviestit ovat myös tärkeitä tietomurtojen selvityksessä, koska ne sisältävät tiedot kaikista laitteen läpi kulkevista yhteyksistä. Palomuri pystyy myös liikennettä tutkimalla havaitsemaan ja kirjaamaan loikiin erilaisia uhkia, kuten virus- ja haittaohjelmasovelluksia. Audit- ja palomuriviestien sisältämää tietoa yhdistämällä luodaan hyvät edellytyksen tietomurtojen todentamiseen ja selvittämiseen.

14.2 Lokiviestien tarkastelu ja haku

Lokiviestien tarkastelu ja haku tapahtuu helposti Graylog2-verkkokäyttöliittymän avulla. Käyttöliittymä näyttää viestit saapumisjärjestyksessä ja erottelee niiden kentät selkeästi toisistaan. Valmiiksi määritetyt viestivirrat mahdollistavat tietyn tyyppisten viestien nopean tarkastelun ilman hakumääriä.

Graylog2 mahdollistaa viestien tehokkaan hakemisen. Lokiviestien parsiminen erillisiin kenttiin Logstash-sovelluksella helpottaa tietyn tyyppisten viestien suodattamista ja hakua kentän arvon perusteella. Voidaan esimerkiksi hakea kaikki viestit, joissa user-kenttä sisältää tietyn käyttäjänimen. Hakuja voidaan myös tehdä vapaasti viestien sisällöstä.

Viestien määrästä on mahdollista tuottaa graafeja, joiden avulla voidaan havaita esimerkiksi toimintahäiriö verkossa. Graafeja tarkastelemalla nähdään myös milloin ympäristön käyttöaste on korkeimmillaan. Kuvio 48 huomataan kuinka viestien määrä nousee työpäivän alkaessa aamu kahdeksan aikaan ja alkaa laskea iltapäivällä kello neljää kohti.



Kuvio 48. Graafi päivän viestimäärästä

Viestien tarkastelun keskittäminen yhteen paikkaan mahdollistaa hyvän kokonaiskuvan saamisen ympäristön tapahtumista ja erilaisten vikatilanteiden sekä trendien havaitsemisen. Kuvion 49 esimerkissä lokiviesteistä on haettu kaikki domain nimet, joita LabraNet-verkon nimipalvelimilta on kysytty yleisyysjärjestyksessä esitettynä:

Count	Term
104244	jamk.fi
100993	labranet.jamk.fi
77844	.168.192.in-addr.arpa
69182	feeds.feedburner.com
50227	.168.192.in-addr.arpa
47241	daisy.ubuntu.com
44258	staging-api.engin.io
42151	http
32753	.168.192.in-addr.arpa
29772	.168.192.in-addr.arpa
27278	www.staging-engin.io
25863	.
23270	www.iltasanomat.fi
22800	rss.kauppalehti.fi
21589	safebrowsing.clients.google.com
12733	199.79.48.86.in-addr.arpa
11522	apresolve.spotify.com
10751	play.spotify.com

Kuvio 49. Yleisimpien domain nimien haku

14.3 Viestien arkistointi ja poisto

KATAKRI-kriteeristö asetti lokiviestien säilytyksen vähimmäisajaksi kuusi kuukautta. Arkistointia varten luotiin säilytyspalvelin, jonka konfigurointi on esitetty kappaleessa 13.21. Viestit tallennetaan tekstitiedostoon Logstash-sovelluksen tuottamassa JSON-formaatissa. Tekstitiedostoista viestejä voidaan hakea esimerkiksi grep-komennolla. Tiedostot pakkaamalla niiden kuluttama tila saadaan pudotettua noin kymmeneen prosenttiin alkuperäisestä. Pakatuista tiedostoista viestejä voidaan hakea zgrep-komennolla. Säilytyspalvelin toimii pitkälti automaattisesti ilman, että sen toimintaan tarvitsee puuttua. Palvelimen tilankäyttöä on hyvä seurata aika ajoin, jotta tallennustila ei pääse loppumaan.

14.4 Tietoturva

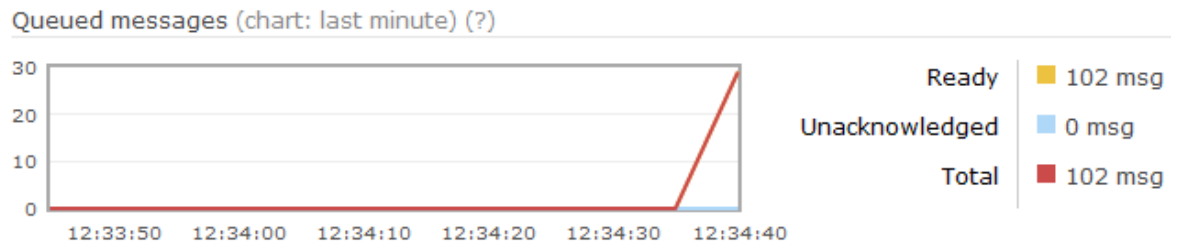
Kolmas KATAKRI-kriteeristön asettama vaatimus on suojattavaa lokitietoa sisältävien viestien asianmukainen suojaus, jolla tarkoitetaan pääsynhallintaa, käsittelyä ja poistoa. Pääsynhallinta on toteutettu osassa palvelimista ja laitteista keskitetyn LDAP-tietokannan kautta, jolloin ylläpitäjät kirjautuvat omilla henkilökohtaisilla tunnuksillaan. Osassa palvelimissa käytetään yhteisiä ylläpitotunnuksia, joiden salasanoja säilytetään salatussa tietokannassa. Kaikki kirjautumiset järjestelmiin tallennetaan lokiin ja osassa kohteista myös ylläpitäjien suorittamat komennot. Palvelimille pääsee kirjautumaan ainoastaan ylläpitäjien työasemilta salattua SSH-protokollaa tai VMware vSphere -konsolia käyttäen.

Lokiviestit liikkuvat eristetyssä hallintaverkossa ja viestejä pääsevät tarkastelemaan ainoastaan ylläpitäjät. Lokijärjestelmän palvelimet on lisäksi pyritty suojaamaan käyttämällä paikallista palomuuria, joka sallii ainoastaan lokiviestien siirtoon tarkoitetut yhteydet ja hallintayhteydet. Lisäksi palvelimilla on käytössä SELinux-teknologia, joka mahdollistaa tarkan pääsynhallinnan järjestelmän eri osiin. RabbitMQ- ja Graylog2-verkkokäyttöliittymissä otettiin käyttöön SSL-salaus.

14.5 Skaalautuvuus ja luotettavuus

Virtuaaliympäristö mahdollistaa resurssien nopean lisäämisen ja sen myötä skaalautumisen suurempien viestimäärien käsittelyyn. Esimerkiksi Graylog-palvelimeen asetettiin lisää prosessoriytimiä ja tallennustilaa, jotta se kykenisi käsittelemään alun perin arvioitua suurempia viestimääriä.

Viestien luotettava siirto pyrittiin varmistamaan asettamalla Logstash asetuksiin durable- ja persistent-parametrit, jotka varmistavat jonojen ja niissä olevien viestien säilymisen, vaikka vaihde uudelleenkäynnistettäisiin. Viestejä ei myöskään menetetä kohdepalvelimen vikatilanteessa, koska vaihde tallentaa viestit jonoon kunnes kohdepalvelin muodostaa yhteyden uudelleen. Tämä toiminta nähdään kuviossa 50.

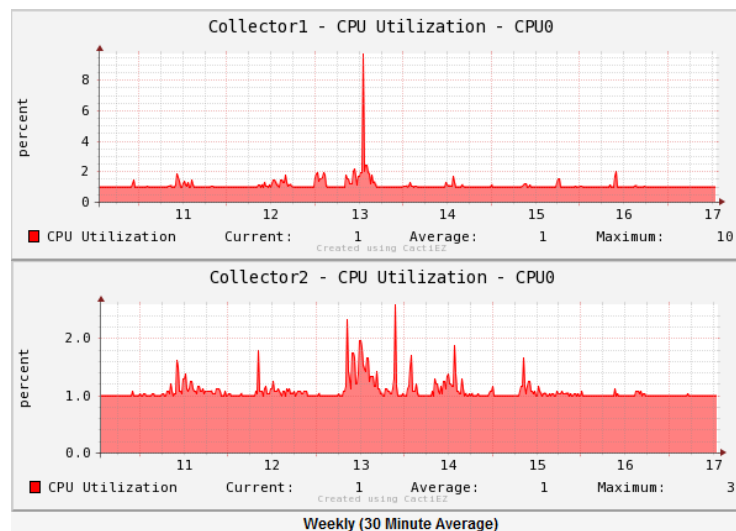


Kuvio 50. AMQP-vaihteen viestien lisäys jonoon

14.6 Suorituskyky

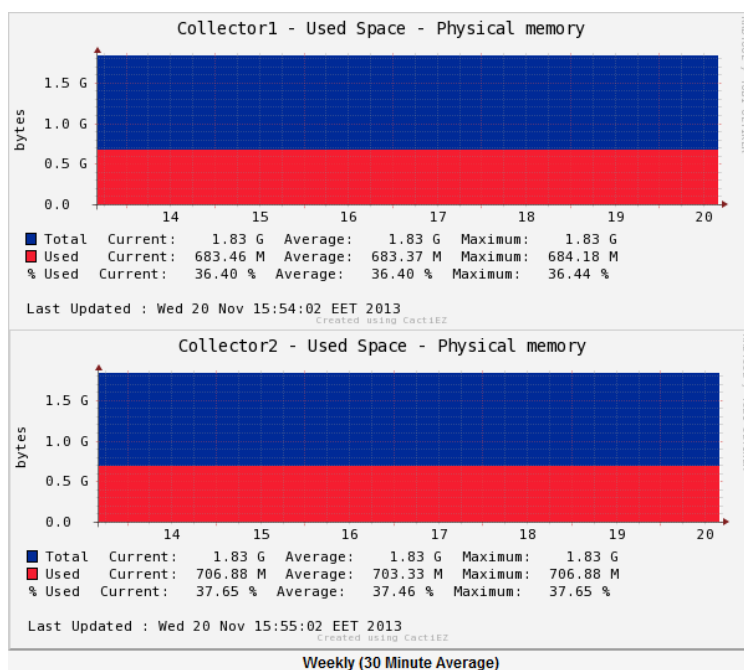
Järjestelmän eri palvelinten suorituskykyä seurattiin noin viikon ajan. Palvelimista kerättiin tietoa prosessorikuormasta, muistin kulutuksesta, kiintolevyjen I/O-operaatioiden määrästä, kiintolevyjen tilankulutuksesta ja verkkoliikenteen määrästä. Tieto kerättiin SNMP-protokollan avulla Cacti-valvontasovellukseen, joka laati kuviot resurssien kulutuksesta.

Collector1- ja Collector2-palvelimilla kiinnitettiin eniten huomiota prosessorikuormaan, joka vaikuttaa kuinka nopeasti ne pystyvät käsittelemään viestejä. Kuviosta 51 nähdään, että kummankaan prosessorikuorma ei ole erityisen korkea, joten ne pystyvät käsittelemään viestejä ilman ongelmia.



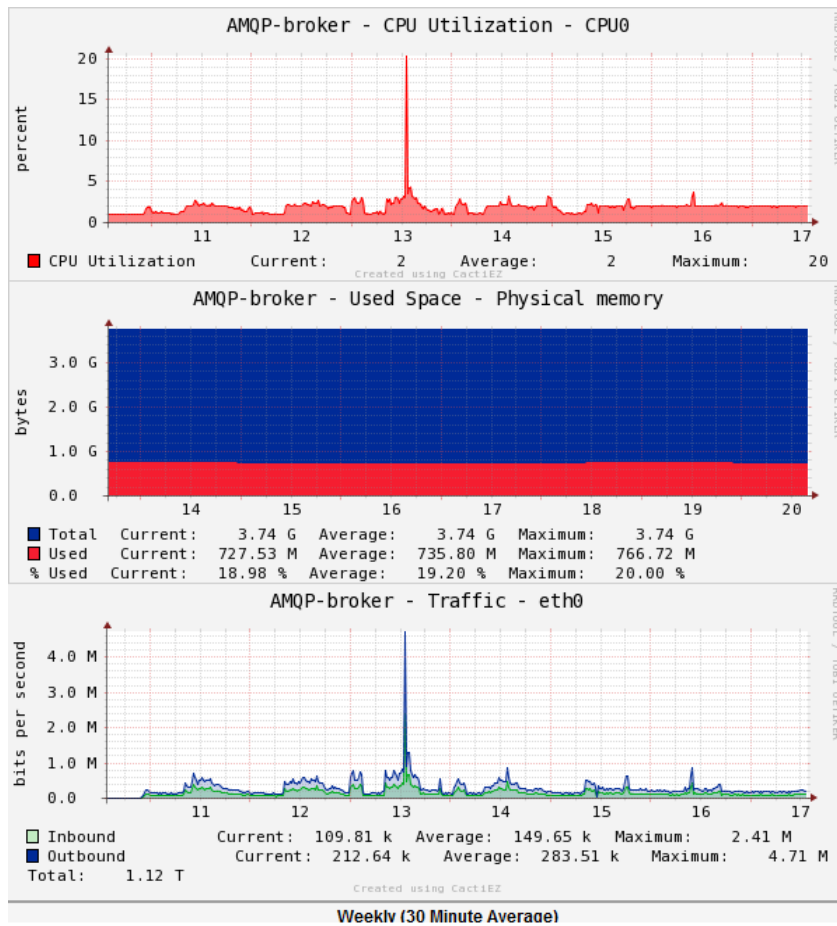
Kuvio 51. Keräyspalvelimien prosessorikuorma

Kuviosta 52 nähdään, että myöskään muistin kulutus ei nouse korkeaksi.



Kuvio 52. Keräyspalvelimien muistin kulutus

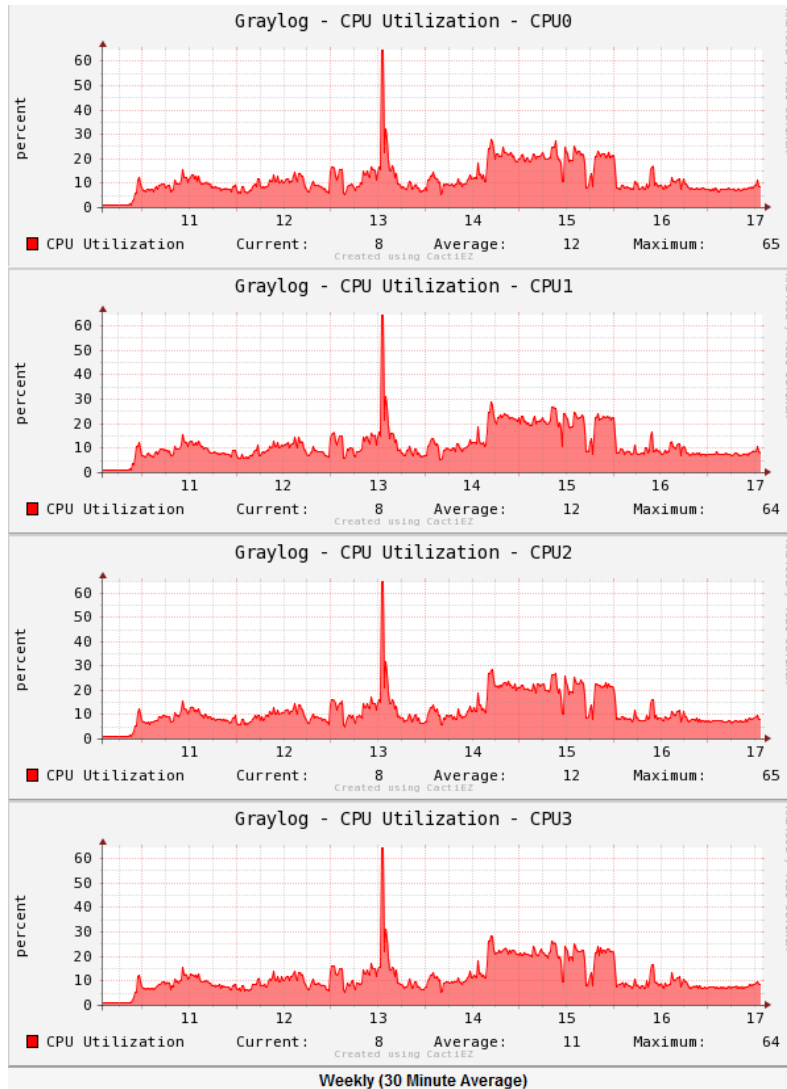
AMQP-vaihteen resurssien kulutusta kuvioista 53 tarkasteltaessa huomataan, että se ei kuluta paljon resursseja vaikka kaikki järjestelmän viestit kulkevat sen läpi.



Kuvio 53. AMQP-vaihteen resurssien kulutus

Järjestelmän palvelimista selkeästi eniten resursseja kuluttaa parsinta- ja tarkastelupalvelin, koska se suorittaa jokaiselle vastaanotetulle viestille parsinnan ja tallentaa ne Elasticsearch-tietokantaan. Palvelimen prosessorien kuorma on keskimäärin noin 12 %, joka ei normaalitilanteessa aiheuta ongelmia viestien käsittelylle. Seurannassa huomattiin suurten viestipiikkien aikana (~1000 viestiä sekunnissa) kuorman nousevan lähelle sataa prosenttia, jolloin palvelimen toiminta hidastui huomattavasti.

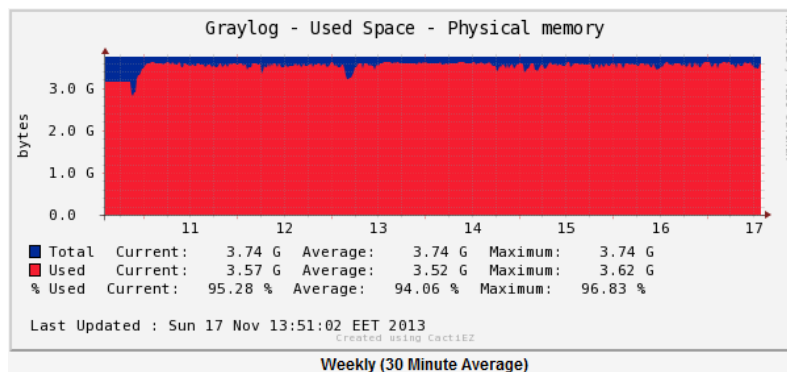
Kuviosta 54 nähdään parsinta- ja tarkastelupalvelimen prosessorien kuorma.



Kuvio 54. Parsinta- ja tarkastelupalvelimen prosessorien kuorma

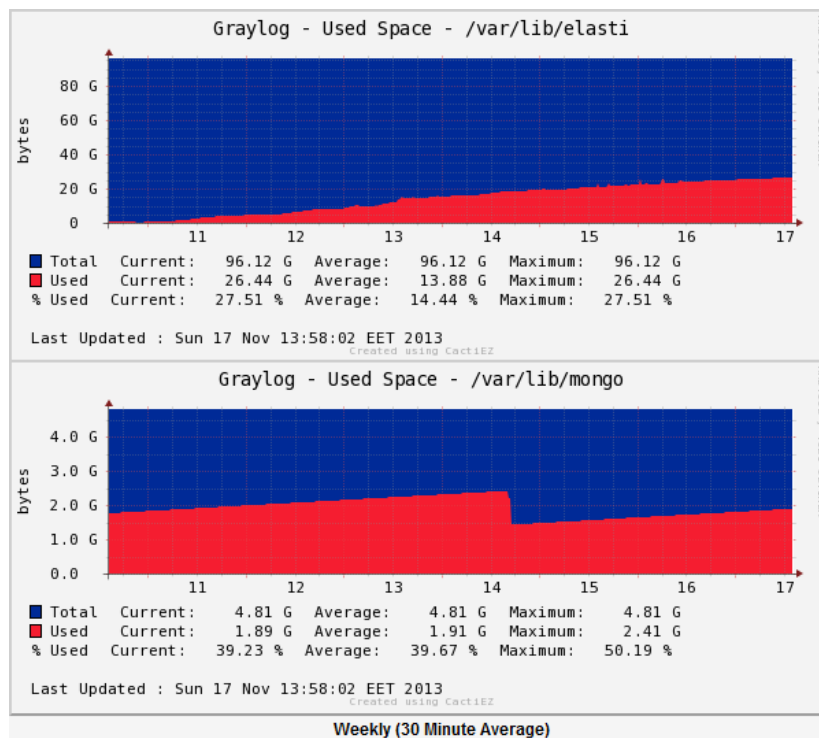
Palvelimen muistin kulutus pysyi seurannan aikana tasaisen korkeana. Muistia voidaan joutua myöhemmin lisäämään kuormituksen kasvaessa.

Kuviosta 55 nähdään parsinta- ja tarkastelupalvelimen muistin kulutus.



Kuvio 55. Parsinta- ja tarkastelupalvelimen muistin kulutus

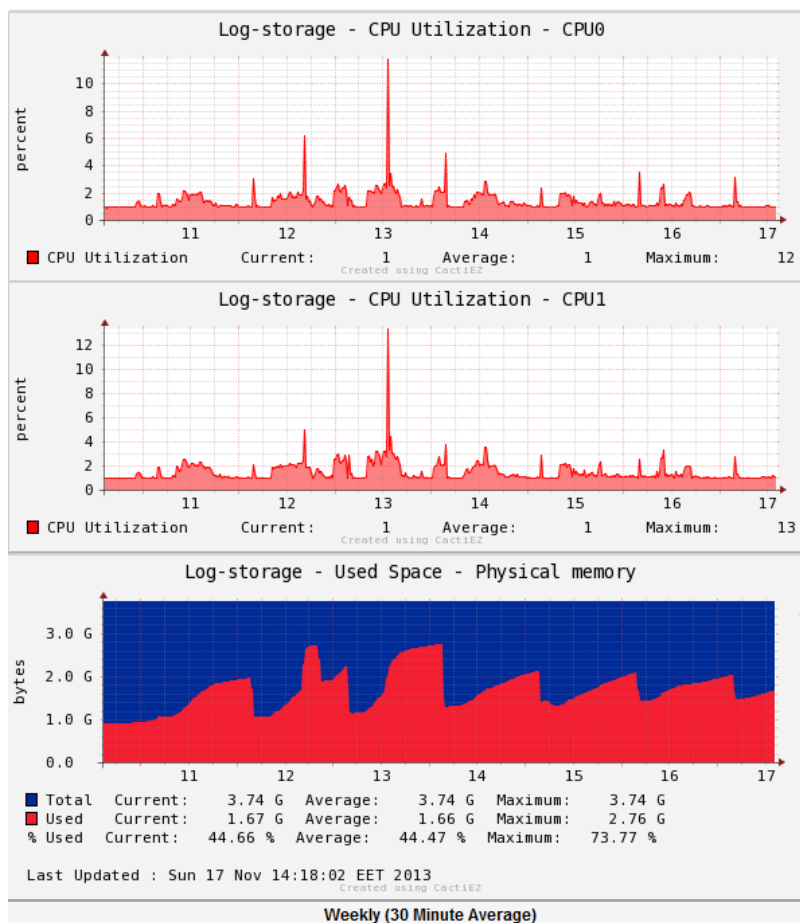
Parsinta- ja tarkastelupalvelimen toiminnan kannalta tärkein resurssi on sen tallennustila. Tallennustilan loppuessa koko palvelimen toiminta lakkaa, joten sen riittävyyteen tulee kiinnittää erityistä huomiota. Palvelimelta valvottiin elasticsearch- ja mongo-osioiden tallennustilaa, joihin Graylog2-palvelin tallentaa lokiviestit. Kuvioista 56 huomataan, että tallennustila ei tule riittämään kuukauden lokiviestien tallentamiseen. Tästä syystä palvelimelle luotiin uusi virtuaaliikiintolevy, joka lisättiin LVM-talenneryhmiin sekä jaettiin elasticsearch- ja mongo-logistenosioiden kesken.



Kuvio 56. Parsinta- ja tarkastelupalvelin tallennustilan kulutus

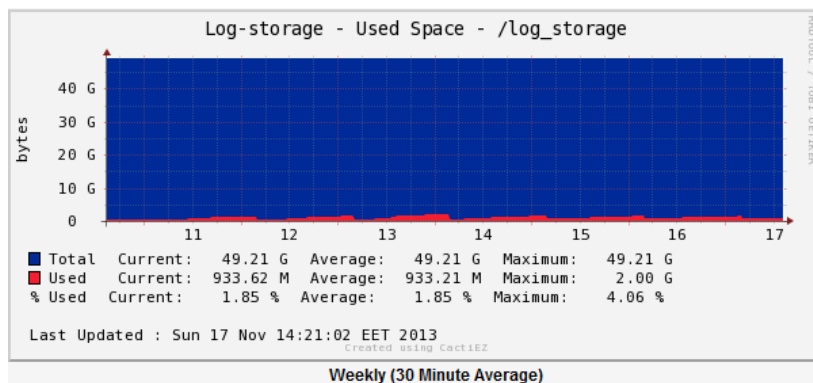
Säilytyspalvelin ei myöskään normaalitilassa aiheuta suurta prosessorikuormaa tai kuluta paljon muistia. Muistin kulutuksessa nähtävät piikit syntyvät, kun palvelin suorittaa lokitiedostojen pakkauksen.

Kuviosta 57 nähdään säilytyspalvelimen prosessorikuorma ja muistin kulutus.



Kuvio 57. Säilytyspalvelimen prosessorikuorma ja muistin kulutus

Säilytyspalvelimella on myös tärkeä tarkkailla tallennustilan riittävyyttä lokitiedostojen arkistointiin. Kuvio 58 nähdään hyvin kuinka tilankulutus kasvaa päivän aikana ja tippuu yöllä, kun lokitiedostot pakataan. Viikon seurantajakson perusteella palvelimelle ei ole tarvetta lisätä nykyisellä kulutuksella lisää tallennustilaa, vaan se riittää mainiosti puolen vuoden lokiviestien säilytykseen.



Kuvio 58. Säilytyspalvelin tallennustilan kulutus

Lokijärjestelmä suoriutui normaalista päivittäisestä viestimäärästä ilman ongelmia. Ainoastaan suurissa viestipiikeissä järjestelmä hidastui, mutta pysyi kuitenkin toiminnassa.

15 Yhteenveto

15.1 Työn toteutus ja tulokset

Työssä käytettyihin sovelluksiin tutustuminen ja testaaminen aloitettiin hyvissä ajoin, joka nopeutti työn toteutusvaihetta. Alkuperäistä suunnitelmaa muutettiin hieman toteutuksen aikana yhdistämällä viestien parsinta ja tarkastelu yhteen palvelimeen. Tällä pyrittiin selkeyttämään järjestelmän rakennetta ja vähentämään tarvittavien virtuaalikoneiden määrää. Muuten toteutus sujui suunnitelmien mukaan ilman suuria ongelmia. Selkeästä eniten aikaa kuluttivat eri lokiformaattien suodatussääntöjen laatiminen ja testaus.

Työn tuloksena saatiin lokijärjestelmä, joka täyttää vaatimusmäärittelyssä asetetut vaatimukset. Joitain vaatimusmäärittelyssä asetettuja lokilähteitä ja kerättäviä tietoja ei pystytty toteuttamaan laitteiden puutteellisten ominaisuuksien vuoksi. Lisäksi joitain lähteitä, kuten VMware-palvelimia ei ehditty ottaa työhön mukaan.

Lokijärjestelmä todettiin heti käyttöönoton jälkeen hyödylliseksi työkaluksi järjestelmien vikatilanteiden havaitsemiseen ja ongelmien ratkaisuun. Lokiviestejä tarkastelemalla pystyttiin havaitsemaan vika verkossa ja paikallistamaan lähde IP-osoitteen perusteella. Lokijärjestelmää tullaan oletettavasti hyödyntämään tulevaisuudessa paljon vianselvitykseen.

Lokiviestien kerääminen yhteen tietokantaan ja tiedon parsiminen yhtenäiseen muotoon yhdistettynä tehokkaiisiin hakuominaisuuksiin antaa ylläpitäjille erittäin hyvät edellytykset ympäristön toiminnan valvomiseen sekä vikatilanteiden ja väärinkäytösten jälkikäteiseen selvittämiseen. Lokiviestien arkistointi mahdollistaa useiden kuukausien takaisten tapausten selvittämisen.

15.2 Tulevaisuuden kehityskohteet

Lokijärjestelmän tehokkaan hyödyntämisen kannalta on tärkeää, että se pidetään ajan tasalla liittämällä esimerkiksi ympäristöön lisätyt uudet laitteet järjestelmään. Lokilähteiden osalta opinnäytetyötä on tarkoitus kehittää eteenpäin tutkimalla VMware-järjestelmien liittämistä osaksi lokijärjestelmää, koska ne ovat hyvin tärkeässä roolissa koko ympäristön toiminnan kannalta.

Työssä käytetyistä sovelluksista Logstash ja Graylog2 ovat melko uusia ja hyvin aktiivisessa kehityksessä. Sovellusten tulevat versiot näyttävät lupaavilta, sisältäen hyödyllisiä ominaisuuksia, jotka parantavat lokiviestien prosessointia ja analysointia entisestään. Lokijärjestelmä on tarkoitus päivittää käyttämään sovellusten uusia versioita niiden ilmestyessä. Järjestelmän modulaarisuus mahdollistaa myös kokonaan uusien sovellusten testaamisen nykyisen järjestelmän rinnalla.

Lähteet

- About. 2013a. Kibana verkkosivut. Viitattu 15.9.2013. <http://kibana.org/about.html>
- About. 2013b. Graylog2 verkkosivut. Viitattu 27.9.2013. <http://graylog2.org/about>
- About. 2013c. Nxlogin verkkosivut. Viitattu 3.11.2013. <http://nxlog-ce.sourceforge.net/about>
- Cholakian, A. 2013. Exploring Elasticsearch. Viitattu 21.9.2013. <http://exploringelasticsearch.com/>
- Chuvakin, A., Schmidt, K. & Crishtopher, P. 2012. Logging and Log Management: The Authoritative Guide to Understanding the Concepts Surrounding Logging and Log Management. Syngres. Viitattu 7.9.2013.
- Docs. 2013. Logstash verkkosivut. Viitattu 15.9.2013. <http://logstash.net/docs/1.2.1/>
- Eaton, I. 2003. The Ins and Outs of System Logging Using Syslog. Viitattu 26.8.2013. <https://www.sans.org/reading-room/whitepapers/logging/ins-outs-system-logging-syslog-1168>
- Event Logs. Microsoft Technet Library. Viitattu 8.9.2013. <http://technet.microsoft.com/en-us/library/cc722404.aspx>
- Features. 2013. Rsyslog verkkosivut. Viitattu 31.8.2013. <http://www.rsyslog.com/features/>
- Gerhards, R. 2009. The Syslog Protocol. Request for Comments: 5424. Network Working Group. Viitattu 25.8.2013. <http://tools.ietf.org/html/rfc5424>
- Grimes, R. 2010. Living the log management lifecycle. InfoWorld 4.8.2010. Viitattu 4.8.2013. <http://www.infoworld.com/t/security-eventinformation-management/living-the-log-management-lifecycle-765>
- Graylog Extended Log Format. 2013. Graylog2 verkkosivut. Viitattu 27.9.2013. <http://graylog2.org/about/gelf>
- Hallam-Baker, P. & Behlendorf, B. N.d. Extended Log File Format. W3C Working Draft. Viitattu 7.9.2013. <http://www.w3.org/TR/WD-logfile>
- Home. 2013a. Logstash verkkosivut. Viitattu 15.9.2013. <http://logstash.net/>
- Home. 2013b. Kibana verkkosivut. Viitattu 15.9.2013. <http://kibana.org/>
- Jami, A. 2009. The difference between SEM, SIM and SIEM. Viitattu 17.8.2013. <http://amirjamil.blogspot.fi/2009/07/difference-between-sem-sim-and-siem.html>
- Kansallinen turvallisuusauditointikriteeristö. 2013. Puollustusministeriö. Viitattu 10.8.2013. http://www.defmin.fi/files/1870/KATAKRI_versio_II.pdf

- Kansallinen turvallisuusauditointikriteeristö (KATAKRI). 2013. Suomen Puolustusministeriön verkkosivut. Viitattu 10.8.2013. [http://www.defmin.fi/hallinnonala/puolustushallinnon_turvallisuustoiminta/kansallinen_turvallisuusauditointikriteeristo_\(katakri\)](http://www.defmin.fi/hallinnonala/puolustushallinnon_turvallisuustoiminta/kansallinen_turvallisuusauditointikriteeristo_(katakri))
- Kent, K., Souppaya, M. 2006. Guide to Computer Security Log Management. Viitattu 17.8.2013.
- Klishin, M. 2013. AMQP 0-9-1 Model Explained. Viitattu 22.9.2013. <http://www.rabbitmq.com/tutorials/amqp-concepts.html>
- Kuč, R. & Rogoziński M. 2013. Elasticsearch Server. Viitattu 21.9.2013. Packt Publishing
- Liiketoiminnan jatkuvuus. 2013. Suomen Pankin verkkosivut. Viitattu 14.8.2013. http://www.suomenpankki.fi/fi/rahoitusjarjestelman_vakaus/liiketoiminnan_jatkuvuus/pages/default.aspx
- Lokiohje. 2009. Valtionvarainministeriön verkkosivut. Viitattu 3.8.2013. http://www.vm.fi/vm/fi/04_julkaisut_ja_asiakirjat/01_julkaisut/05_valtionhallinnon_tietoturvallisuus/20090511Lokioh/Vahti_3_NETTI.pdf
- Lonvick, C. 2011. The BSD syslog Protocol. Request for Comments: 3164. Network Working Group. Viitattu 26.8.2013. <http://www.ietf.org/rfc/rfc3164.txt>
- Mauro, D. & Schmidh, K. 2005. Essential SNMP. Viitattu 1.9.2013. O'Reilly Media
- Menn, V. 2006. New Tools for Event Management in Windows Vista. Technet Magazine. Viitattu 8.9.2013. <http://technet.microsoft.com/en-us/magazine/2006.11.eventmanagement.aspx>
- Miao, F., Ma, Y. & Salowy, Y. 2009. Transport Layer Security (TLS) Transport Mapping for Syslog. Request for Comments: 5425. Network Working Group. Viitattu 1.9.2013. <http://tools.ietf.org/html/rfc5425>
- Plugins. 2013. RabbitMQ verkkosivut. Viitattu 24.9.2013. <http://www.rabbitmq.com/plugins.html>
- Platform support. 2013. BalaBit IT Security verkkosivut. Viitattu 31.8.2013. <http://www.balabit.com/network-security/syslog-ng/opensource-logging-system/overview/platforms>
- Platforms. 2013. Rsyslog wiki. Viitattu 31.8.2013. <http://wiki.rsyslog.com/index.php/Platforms>
- Pricing. 2013. Splunkin verkkosivut. Viitattu 29.9.2013. <http://www.splunk.com/view/pricing/SP-CAAADFV>
- Product features and benefits. 2013. BalaBit IT Security verkkosivut. Viitattu 31.8.2013. <http://www.balabit.com/network-security/syslog-ng/opensource-logging-system/features>
- Sarjakivi, P. 2013a. Lokienhallinta. Viitattu 14.8.2013. <http://www.nixu.com/fi/palvelualueet/lokienhallinta>

Sarjakivi, S. 2013b. Kaikki hyöty irti tietoturvasojen lokinhallintavaatimuksesta. Viitattu 17.8.2013. <http://tietovastuu.fi/blogi/2013/04/kaikki-hyoty-tietoturvasojen-lokienhallintavaatimuksesta/>

Schroeck, M., Shockley, R. & Smart, J., Romero-Morales, D., Tufano, P. 2012. Analytics: The real-world use of big data. IBM Institute for Business Value. Viitattu 17.8.2013.

[http://www.ibm.com/smarterplanet/global/files/se_sv_se_intelligence_Analytics - The real-world use of big data.pdf](http://www.ibm.com/smarterplanet/global/files/se_sv_se_intelligence_Analytics_-_The_real-world_use_of_big_data.pdf)

SpiderNet. 2009. LabraNetin verkkosivut. Viitattu 28.9.2013. <http://student.labranet.jamk.fi/SpiderNet/>

Study Network for ICT. 2009. LabraNetin verkkosivut. Viitattu 28.9.2013. <http://student.labranet.jamk.fi/LabraNet/>

The Foundation of Log Management. 2013. BalaBit IT Security verkkosivut. Viitattu 31.8.2013. <http://www.balabit.com/network-security/syslog-ng>

Tulloch, M. 2010. Managing event logs from the command line. WindowsNetworking.com verkkosivu. Viitattu 8.9.2013.

<http://www.windowsnetworking.com/kbase/WindowsTips/WindowsServer2008/AdminTips/Admin/Managingeventlogsfromthecommandline.html>

Tutustu JAMKiin. 2013. Jyväskylän ammattikorkeakoulun verkkosivut. Viitattu 28.9.2013. <http://www.jamk.fi/>, tutustu.

Webopedia. 2013. Big data. Viitattu 17.8.2013. http://www.webopedia.com/TERM/B/big_data.html

What can RabbitMQ do for you?. 2013. RabbitMQ verkkosivut. Viitattu 24.9.2013. <http://www.rabbitmq.com/features.html>

What is Splunk Enterprise?. 2013. Splunkin verkkosivut. Viitattu 29.9.2013. <http://www.splunk.com/view/splunk/SP-CAAAG57>

Liitteet

Liite 1. Ohje uuden lokilähteen lisäykseen

Uutta lokilähdettä lisättäessä konfiguroidaan lokilähde ja parsintapalvelin. Lokilähteen asetuksiin tulee määrittää keräyspalvelimen IP-osoite ja portti. Keräyspalvelimena voidaan käyttää Collector1 tai Collector2-palvelinta. Keräyspalvelimet kuuntelevat Syslog-viestejä portissa 514 ja Collector2 voi myös vastaanottaa GELF-tyypin viestejä porttiin 12201.

Parsintapalvelimen Logstash-sovellukselle tulee määrittää suodattimet lokiviestien parsimiseen. Ennen suodatinsääntöjen luomista kannattaa tutustua lokilähteen tuottamaan lokiformaattiin ja valmistajan dokumentaatioon asiasta. Asetustiedostot sijaitsevat /opt/logstash/logstash.d/-kansion alla. Osalle lokilähteistä, kuten Linux-palvelimille sekä Juniper- ja Cisco-laitteille on valmiiksi määritetyt suodatussäännöt, joita voidaan hyödyntää uusia lokilähteitä lisättäessä. Tällöin suodatussäännön ehtolauseeseen tulee lisätä uuden lähteen IP-osoite.

Jos lokilähteen tuottama viestiformaatti ei täsmää olemassa olevien suodatussääntöjen kanssa, tulee sille luoda erilliset säännöt. Logstash.d-kansion alle luotava uusi tiedosto tulee nimetä muodossa: "<järjestysnumero>_<nimi>.conf". Järjestysnumeron tulee olla 00 ja 99 väliltä.

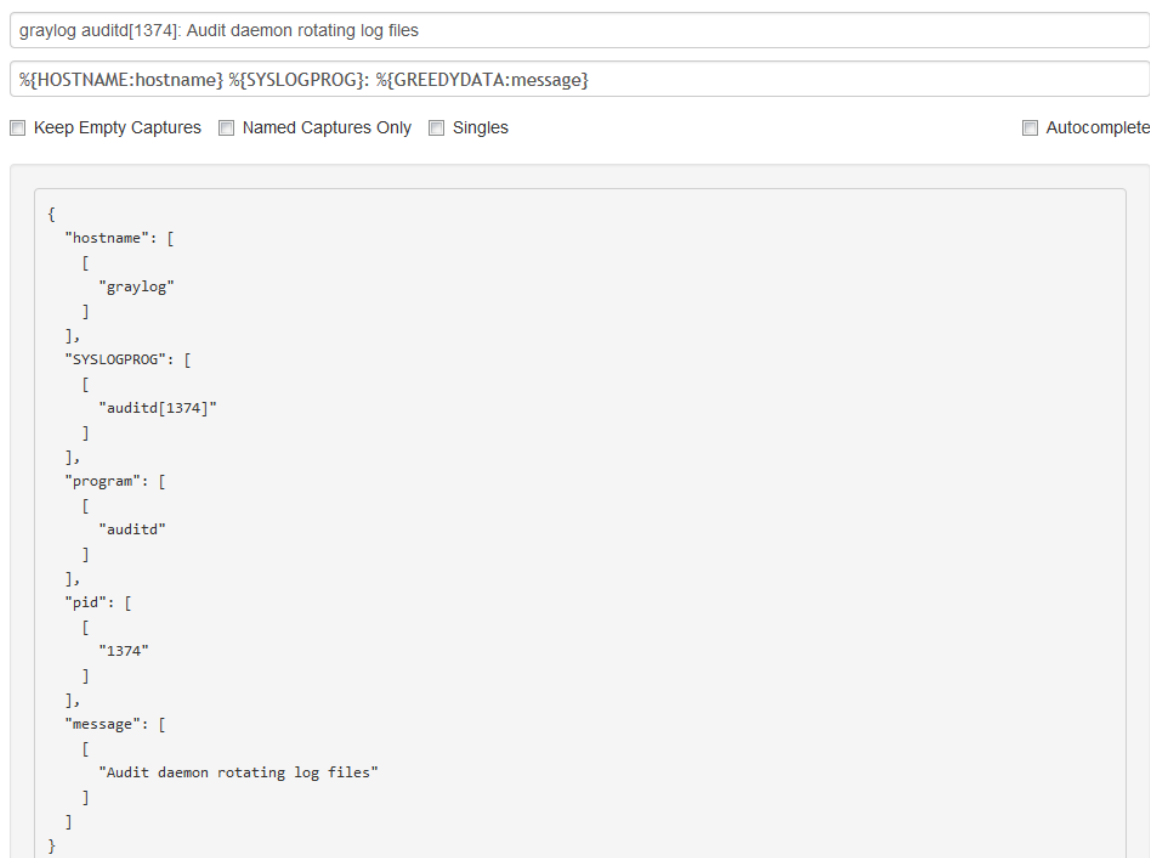
Tiedoston rakenne on seuraava:

```
filter {
    if [host] in "<lähteen IP-osoite>" {
        suodatussäännöt...
    }
}
```

Suodatussäännöt ovat Logstash-lisäosia, joilla voidaan muokata lokiviestien muotoa ja sisältöä. Suodatussääntöjä luotaessa kannattaa tutustua Logstash-sovelluksen dokumentaatioon, joka sisältää lisäosien kuvaukset ja käytettävät parametrit. Dokumentaatio löytyy osoitteesta: <http://logstash.net/>. Mallia kannattaa katsoa myös olemassa olevista suodatussäännöistä.

Tärkein viestien suodatukseen käytettävä lisäosa on grok, jolla viesteistä parsitaan halutut arvot erillisiin kenttiin. Tämä tapahtuu mallien avulla, jotka ovat käytännössä valmiiksi määritettyjä regular expression -lausekkeita. Lista malleista löytyy osoitteesta: <https://github.com/logstash/logstash/tree/master/patterns>.

Kuviossa 59 nähdään havainnollinen esimerkki grok-suodattimen toiminnasta. Kuva on peräisin Grok Debugger -sovelluksesta, joka on erittäin hyödyllinen grok-sääntöjen toimivuuden testaukseen. Sovellus löytyy osoitteesta: <http://grokdebug.herokuapp.com/>.



Kuvio 59. Grok Debugger esimerkki

Viimeiseksi suodatussäännöksi on hyvä lisätä mutata-suodatin, joka poistaa ylimääräiset version- ja tags-kentät viesteistä:

```

mutate {
  remove_field => [ "tags", "@version" ]
}

```

Suodattimen luonnin jälkeen Logstash-prosessi tulee käynnistää uudelleen komennolla:

service logstash restart

Tämän jälkeen lokiviestien pitäisi alkaa ilmestyä Graylog2-verkkokäyttöliittymään.