jamk.fi

Palomuurin IDS-raportointi

Jari Lehtonen

Opinnäytetyö
Toukokuu 2018
Tekniikan ja liikenteen ala
Insinööri (AMK), tietoverkkotekniikan tutkinto-ohjelma

Jyväskylän ammattikorkeakoulu
JAMK University of Applied Sciences

jamk.fi

Kuvailulehti

Tekijä(t)	Julkaisun laji	Päivämäärä
Lehtonen, Jari	Opinnäytetyö, AMK	08/05/2018
	Sivumäärä 48 + 33	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty: x

Työn nimi
Palomuurin IDS-raportointi

Tutkinto-ohjelma
Tietoverkkotekniikka

Työn ohjaaja(t)
Karo Saharinen, Mika Rantonen

Toimeksiantaja(t)
TNNet Oy

Tiivistelmä

Opinnäytetyön toimeksiantajana toimi jyväskyläläinen internetoperaattori ja laitesalipalveluntarjoaja TNNet Oy. Työn toimeksiantona oli implementoida tunkeutumisen havaitsemisjärjestelmän (IDS) palomuuripalveluun sekä suunnitella ja toteuttaa tämän pohjalta raportointijärjestelmä yrityksen nykyistä palomuuri-infrastruktuuria hyödyntäen. TNNet Oy käyttää palomuurijärjestelmänään pääosin virtualisoituja pfSense palomuureja.

IDS:llä tarkoitetaan tietoliikenteen valvontaa ja analysointia siten, että siitä voidaan havaita potentiaalisia uhkia ja väärinkäytöksiä. Palomuurit taas on suunniteltu estämään jo tunnettuja, etukäteen määriteltyjä uhkia. Nämä yhdistämällä, voitaisiin toteuttaa tietoturvallisempi palomuuripalvelu. Raportointijärjestelmän ensisijainen tavoite oli sekä parantaa nykyistä palomuuripalvelua tuomalla siihen lisäksi IDS-valvonta, että estää nykyisten asiakkaiden poistumista kilpailijoiden myydessä palomuuripalveluitaan raportin siivellä.

IDS:ää voidaan toteuttaa useilla eri tavoilla ja järjestelmillä, mutta pfSenseen helposti saatavilla oleva lisäpaketti SNORT on yksi tunnetuimmista IDS-järjestelmistä ja myös helposti liitettävissä nykyiseen palomuuripalveluun. PfSense taas generoi suodattamastaan liikenteestä syslogia, jota voidaan käsitellä erilaisilla tietokantajärjestelmillä, suosituimpana esimerkiksi ELK-Stackilla.

Työn tuloksena syntyi hyvin tavoitteita palveleva järjestelmä, joka toi myös yllättävää lisäarvoa TNNEt Oy:n muulle toiminnalle ja avasi uusia palvelutuotemahdollisuuksia. Palomuuripalvelu saatiin yhdistettyä toivotusti IDS-valvontaan ja niiden toiminnasta saatiin generoitua niin asiakkaille kuin sisäisesti omalle organisaatiolle lisäarvoa tuottava raportti.

Avainsanat (asiasanat)
ELK-stack, Elasticsearch, IDS, Kibana, Logstash, Loki, Palomuuri, PfSense, Syslog

Muut tiedot (salassa pidettävät liitteet)

jamk.fi

Description

Author(s)	Type of publication	Date
Lehtonen, Jari	Bachelor's thesis	08/05/2018
		Language of publication: Finnish
	Number of pages 48 + 33	Permission for web publication: x

Title of publication
Reporting service for Firewall IDS

Degree programme
Information Technology

Supervisor(s)
Karo Saharinen, Mika Rantonen

Assigned by
TNNet Oy

Abstract

This bachelor's thesis was assigned by TNNet Oy, a Jyväskylä-based Internet service and datacenter provider. The assignment was to implement the firewall service with Intrusion Detection System (IDS) and design and engineer a reporting service based on that, using the existing firewall infrastructure of the company. TNNet Oy uses virtualized pfSenses as their essential firewall operating system.

IDS in practice functions as a means to monitor and analyze telecommunications, in a way that potential threats and abuses can be encountered, whereas firewalls are designed to prevent well-known, predetermined threats. By combining these, a more secure firewall service can be achieved. The primary goal of the reporting system was to upgrade the existing firewall service by merging it with IDS, to prevent losing already existing customers to competitors who sell their firewalls with the existence of a reporting system.

IDS could be implemented in a variety of different designs and system; however the pfSense includes an easy-to-use add-on, called SNORT, which is also one of the best-known IDS systems, and is therefore easy to implement into the existing infrastructure. PfSense generates from its filtered traffic syslog by default, which can then be handled by various database systems, most popularly with e.g. ELK-Stack.

This assignment resulted in a system that served the goal well, and brought surprising added value to TNNet Oy's other products, opening the possibility of completely new service products. The firewall service was combined with IDS control as was desired and this functionality could be produced into a reporting system, adding value to both customers and the organization.

Keywords/tags (subjectshttp://vesa.lib.helsinki.fi/)
ELK-stack, Elasticsearch, Firewall, IDS, Kibana, Logstash, Log, PfSense, Syslog

Miscellaneous (Confidential information)

Sisältö

Lyhenteet

Kuviot

Lyhenteet

BSD	Berkley Software Distribution
DNS	Nimipalvelu, Domain Name Service
GDPR	General Data Protection Regulation
IDS	Tunkeutumisen havaitsemisjärjestelmä, Intrution Detection System
IETF	Internet Engineering Task Force
IPS	Tunkeutemisen estojärjestelmä, Intrution Prevention System
JVM	Java virtuaalikone, Java Virtual Machine
NAT	Network Address Translation
RAM	Keskusmuisti, Random Access Memory
RFC	Request for Comments
UUID	Unique Universal Identifier
VLAN	Virtuaalilähiverkko, Virtual local area network
VPN	Virtual Private Network

1 Lähtökohdat

1.1 Toimeksiantaja

Opinnäytetyön toimeksiantajana toimi TNNet Oy. TNNet Oy on jyväskyläläinen verkko-operaattori, joka tarjoaa sekä internet- että laitesalipalveluita. Pääasiallinen asiakaskunta ovat muut yritysasiakkaat, mutta myös kuluttajille tarjotaan palveluita. TNNet Oy tekee paljon yhteistyötä muiden paikallisten IT-yritysten kanssa, jotka hoitavat esimerkiksi yhteisten asiakkaiden sisäverkkotyöt. TNNet Oy:n henkilöstöosastot ovat asennus, verkko- ja palvelintekniikka sekä myynti ja asiakaspalvelu.

TNNet Oy:ssa työn tilaajana toimi verkkotekniikan osasto. Verkkotekniikan päätehtävä on rakentaa ja ylläpitää yrityksen tietoliikenneverkkoa loogisella tasolla, kun taas vastaavasti asennus hoitaa verkon fyysisen rakentamisen.

1.2 Toimeksianto ja työn tavoitteet

25. päivä toukokuuta 2018 voimaan astuva EU:n yleinen tietosuoja-asetus (General Data Protection Regulation, GDPR) on tuottanut useissa yrityksissä paljon päänvaivaa sekä mahdollistanut IT-yrityksille uusia tapoja myydä tuotteita. TNNet Oy:ssa huomattiinkin kevään aikana, että muutama olemassa olevista palomuuriasiakkaista vaihtoi palveluntarjoajaa, sillä kilpailijoilla oli tarjota heille palomuurista myös jonkinlaista raporttia. Syynä tälle oli käsitys siitä, että raportti olisi GDPR:n mukaan pakollinen. Samasta syystä osa asiakkaista myös kyseli mahdollisuutta saada vastaavaa raporttia tulevaisuudessa TNNet Oy:lta.

Opinnäytetyön toimeksiantona olikin tutkia, miten asiakkaille myytävästä palomuuriratkaisusta saataisiin teetettyä IDS-raportti, ja mitä sen tulisi sisältää, jotta raportti olisi sekä aidosti hyödyllinen että myyvä. Näillä ehdoilla päätettiin tehdä kompromissi sekä hyödyllisen että myyvän datan suhteen. Tavoitetilassa raportti sisältäisi ainakin seuraavat kohdat:

- Kuinka paljon liikennettä estetään sekä estetyn liikenteen yleisimmän viiden portin ja lähdeosoitteen listaus.
- Kuinka paljon liikennettä osuu sallintasääntöihin sekä listaus viidestä eniten osumia saavista säännöistä.

- Kuinka paljon dataa liikkuu missäkin virtuaalilähiverkossa (Virtual local area network, VLAN).
- Liikenne tunnettuihin bottiverkkoihin sekä lähde-IP tälle liikenteelle.
- Porttiskannaukseen ja SQL-injektioon viittaavaa liikenne.

Teoriaosuudessa käsitellään lokitusta sekä työn toteutuksessa käytettyjä työkaluja että niiden toimintaa. Tämän lisäksi tutustuttiin hieman EU:n tietosuoja-asetukseen henkilötietojen ja lokidatan osalta pyrkien samalla tulkitsemaan, miten se vaikuttaa tämän työn lokidataan.

Työn toteutuksessa luotiin ympäristö, jolla voitiin kerätä raporttiin tarvittavaa dataa. Tämän jälkeen raportti generoitiin kerätyn datan pohjalta. Raporttia lähdettiin muokkaamaan parempaan suuntaan iteratiivisesti, aluksi ainoastaan sisäisesti, mutta myöhemmässä vaiheessa myös loppuasiakkaita kuunnellen.

2 Lokitus

2.1 Lokituksen määritelmä

”Loki tarkoittaa aikajärjestyksessä kirjattua tallennetta tapahtumista ja niiden aiheuttajista. Tapahtumat ja muutokset tietojärjestelmissä, sovelluksissa, tietoverkoissa ja tietosisällöissä kirjataan lokiin, eli lokitetaan.” (Lokitusohje 2016, 2.)

Lokia ei kerätä ainoastaan muodon vuoksi, vaan ne ovat elintärkeitä tietojärjestelmien ylläpitoa varten. Lokitiedot kertovat muun muassa mitä, milloin ja miksi jotakin on tapahtunut. Lokeista voidaan myös päätellä, jos jotakin on jäänyt tapahtumatta, eli voidaan varmistua siitä, että ei ole syntynyt huomaamatonta virhettä. Lisäksi lokidataa voidaan käyttää todentamaan, että järjestelmät ovat toimineet oikein. (Lokitusohje 2016, 4.)

On tärkeää, että loki on tarpeeksi kattava. Puutteellinen loki saattaa jäädä kokonaan hyödyttömäksi, jos sillä voidaan esimerkiksi ainoastaan havaita poikkeama, mutta ei saada tarkemmin selville, miksi ja milloin poikkeama on tapahtunut. Viestintäviraston ohjeistuksen mukaan käyttökelpoinen lokitieto sisältää vähintään seuraavat määritteet:

- Aikaleima (milloin tapahtuma oli?).
- Tapahtuma (mitä tehtiin tai yritettiin tehdä?).
- Toimija (kuka tai mikä teki?).
- Tapahtuman lähde (mistä tehtiin, mistä muutostieto on peräisin?).
- Tapahtuman kohde (mihin tietoon tai järjestelmään toiminta kohdistui?).
- Tapahtuman tila (onnistui / ei onnistunut / epäonnistumisen syy).

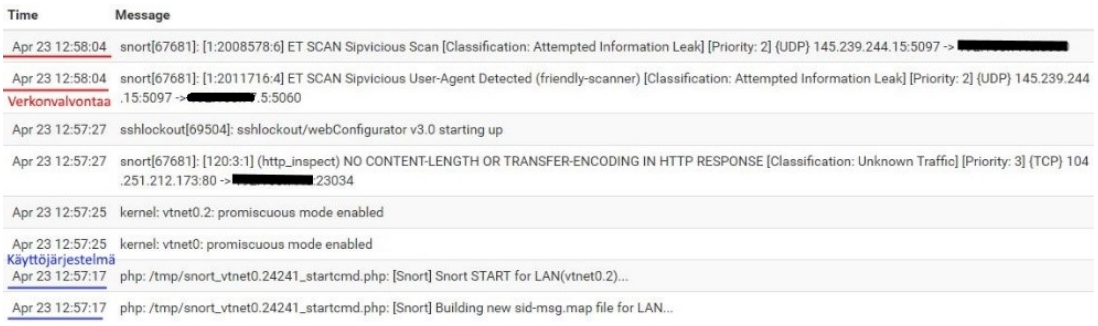
(Lokitusohje 2016.)

Näiden lisäksi lokitiedossa saattaa olla myös tarkempaa dataa lokityypistä riippuen. On kuitenkin tärkeää myös osata mitoittaa lokitus sopivaksi. Jos lokissa on liikaa tietoa, tai lokia tulee liian paljon, se voi tukkia lokijärjestelmän ja tärkeääkin tietoa voi hukkua massiivisen datamäärän alle. (Lokitusohje 2016, 4.)

2.2 Lokityypit

Lokeilla voidaan valvoa hyvinkin erityyppisiä tapahtumia ja järjestelmiä, jonka vuoksi myös erilaisia lokityyppejä on useita. Lokeja voitaisiin jaotella esimerkiksi sen mukaan, mikä on lokin käyttötarkoitus ja mikä laite tai järjestelmä on tuottanut kyseisen lokin. Yksi tapa kategorisoida lokit on Lokitusohjeen (2016) jaottelu neljään eri päätyyppiin: ylläpitoloki, käyttöloki, muutosloki ja virheloki. Näistä esimerkiksi virheloki sisältäisi tiedon järjestelmässä tapahtumista virheistä ja käyttöloki tietokantatapahtumista. Käytännössä yksittäinen loki voi kuitenkin sisällön mukaan olla yhdistelmä montaa eri lokityyppiä. (Lokitusohje 2016, 29.)

Myös eri laitevalmistajat tekevät omia lokijaotteluitaan. Esimerkiksi laitevalmistaja pfSense jaottelee lokinsa sen mukaan, mikä palvelu on lokin tuottanut. Näitä ovat esimerkiksi Firewall-, NTP-, VPN- ja DHCP-lokit. Tarkasteltaessa lokien sisältöä voidaan havaita, että palvelukeskeinen jaottelu on tässä tapauksessa hyvä, sillä yksittäisen palvelun loki voi sisältää hyvinkin erilaista tietoa. Esimerkiksi kuviossa 1 havaitaan pfSensen System-lokissa tietueita, jotka viittaavat itse käyttöjärjestelmään ja tietueita, jotka taas koskevat verkon valvontaa.



Kuvio 1. pfSense System Log

2.3 Lokien käsittely

Termillä “lokien käsittely” viitataan lokin koko elinkaareen, johon kuuluvat lokien kerääminen, analysointi, säilyttäminen, luovuttaminen ja poistaminen tai arkistointi. Koska lokidataa tulee usein huomattavan paljon, liikaa manuaalisesti tarkasteltavaksi, on valikoitava olennaisimmat kohteet, jotka halutaan lokittaa. Tämän jälkeen näitä haluttuja lokitietoja voidaan käsitellä jollakin teknisellä lokien analysointiin tehdyllä työvälineellä ja samalla poistaa ”turhat” lokitiedot. Automaattisen lokin käsittelyn jälkeen saatu data voidaan vielä käsitellä manuaalisesti, ja lokidatan lopullisen käsittelyn tekeekin usein ihminen. (Lokiohje 2009, 15.)

Lokien käsittelyllä pyritään saavuttamaan ja varmistamaan lokitapahtuman osapuolet, kiistämättömyys, kulku sekä ongelmien tai poikkeustilanteiden havaitseminen. Osapuolilla tarkoitetaan kuka, tai mikä toimija (esimerkiksi laitteen IP-osoite) liittyi kyseiseen tapahtumaan. Kiistämättömyyden tavoitteena on, ettei yksikään tapahtuman osapuolista olisi kirjattuna tapahtumaan aiheettomasti. Tapahtuman kulku voidaan dokumentoida lokidataan kronologiseen järjestykseen, jolloin voidaan tarkastella, missä ja miten lokitapahtuma on edennyt. Ongelmien sekä poikkeustilanteiden havaitsemisella voidaan varmistaa järjestelmän asianmukainen toiminta ja mahdollisiin virhetilanteisiin voidaan puuttua, kun ne havaitaan ja niiden selvityksessä voidaan palata takaisin tapahtumaan juurisyyn löytämiseksi. (Lokiohje 2009, 15.)

2.4 Lainsäädännön ja GDPR:n vaatimukset

Lokitietoja kerättäessä tulee ottaa huomioon sisältääkö loki henkilötietoja. Henkilötiedoilla tarkoitetaan dataa, jolla henkilö voidaan tunnistaa yksittäistä henkilöä koskevaksi. Mikäli lokissa on henkilötietoja, tulee lokista henkilörekisteri. Henkilörekisterissä tulee huomioida etenkin henkilötietolain asettamat velvoitteet ja näin ollen lokista on tuotettava rekisteriseloste (Lokitusohje 2016, 6).

GDPR direktiivi ei suoraan määrittele mikä luetaan henkilötiedoksi, vaan kuvailee kaiken yksilöivän tiedon henkilötiedoiksi. Perinteisesti henkilötietoja ovat esimerkiksi henkilön katuosoite ja sosiaaliturvatunnus, mutta tietoliikenteessä yksilöivä tieto on vaikeammin määriteltävissä. Yksilöivää tietoa voisi olla esimerkiksi loki IP- ja MAC-osoitteista. Direktiivin mukaan anonymisoitu tieto ei kuitenkaan ole direktiivin alaista. (Mitä jokaisen kuuluu tietää GDPR:stä N.d.)

GDPR artiklassa 4 määritellään henkilötiedoiksi kaikki tunnistettuun tai tunnistettavissa olevaan luonnolliseen henkilöön liittyvä tieto, jolla voidaan suoraan tai epäsuorasti tunnistaa kyseinen henkilö. Artiklassa 26 taas täsmentää, että ”pseudonymisoidut tiedot, jotka voitaisiin yhdistää luonnolliseen henkilöön lisätietoja käyttämällä, olisi katsottava tiedoiksi, jotka koskevat tunnistettavissa olevaa luonnollista henkilöä.” Lisäksi samassa artiklassa täsmennetään, että GDPR:n säädöksiä ei tule soveltaa anonyymeihin tietoihin, eli tietoihin joiden tunnistettavuus on poistettu siten, ettei tiedoilla voida enää tunnistaa yksittäistä henkilöä. (GDPR – EU:n uusi tietosuoja-asetus N.d., artiklat 4 ja 26.)

Mikäli loki sisältää GDPR:n alaisia henkilötietoja, asettaa se lokille tiettyjä ehtoja: lokin keräykseen tulee saada kyseisiltä henkilöiltä lupa, eli kyseisten henkilöiden pitää olla tietoisia lokista. Tietoturvaloukkauksista tulee ilmoittaa vaikutuksen alaisille henkilöille 72 tunnin sisällä loukkauksen havaitsemisesta, henkilöllä tulee olla oikeus tulla unohdetuksi ja henkilöllä tulee olla oikeus nähdä hänestä kerätty tieto. (Mitä jokaisen kuuluu tietää GDPR:stä N.d.)

GDPR jättää kuitenkin edelleen huomattavasti tulkinnanvaraa henkilötiedon määritelmälle erityisesti verkkotunnistetietojen kohdalla; missä menee anonyymin ja pseudonyymin tiedon raja? Tälle kysymykselle tullaan todennäköisesti saamaan tar-

kempi vastaus, kun asetus on astunut voimaan ja ensimmäiset ennakkotapaukset julkaistaan.

Tässä työssä kerätty lokidata ei TNNet Oy:n tulkinnan mukaan ole GDPR:n alaista, eli kerätty lokitieto ei muodosta henkilörekisteriä. Lokitiedoista löytyvällä datalla ei mitenkään pysty tunnistamaan yksittäistä ihmistä, eikä edes tietokonetta, ellei käytä apunaan jotakin toista rekisteriä, kuten DHCP-lokia, yrityksen sisäistä IP-listaa tai muuta vastaava. Nämä muut rekisterit taas sijaitsevat joko täysin eri järjestelmässä omien suojausten takana, tai suoraan loppuasiakkaalla itsellään, jolloin dataa voidaan pitää tarpeeksi hajautettuna ja täten anonymisoituna.

2.5 Syslog

Tässä työssä käytetty syslog on yksi yleisimmistä lokiprotokollista, jota käytetään lokiviestien hallintaan ja välitykseen. Syslog on standardisoitu IETF:n (Internet Engineering Task Force) RFC (Request for Comments) -dokumentissa 5424, jolla korvattiin syslogin aiempi standardi RFC 3164. (Eaton 2003.)

Syslog kehitettiin alun perin lokittamaan *sendmail* ohjelmistoa BSD (Berkley Software Distribution) -käyttöjärjestelmällä. Syslog yleistyi pian myös muiden käyttöjärjestelmien käyttöön, mutta standardisoimattomana siitä tuli useita yhteensopimattomia versioita. Syslog kuitenkin standardisoitiin IETF:n toimesta vuonna 2001 RFC 3164 standardilla, minkä jälkeen kehitys on ollut yhtenäistä eri syslogin versioiden välillä. (Eaton 2003.)

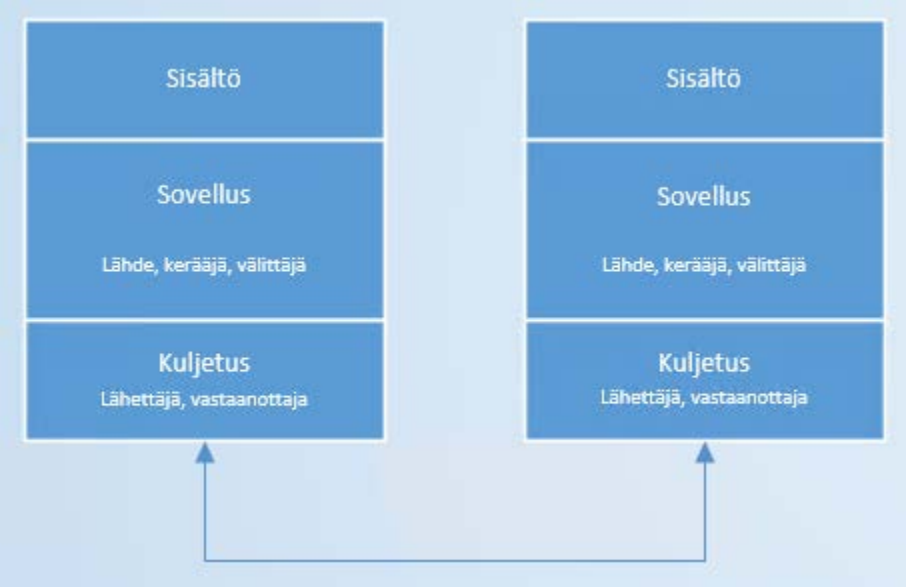
Syslogin arkkitehtuuri on kerroksittainen, jonka tarkoitus on erottaa viestien sisältö kuljetustavasta. Tasoja on kaikkiaan kolme, ja jokaisella niistä on omat tehtävänsä:

- Sisältö: Nimensä mukaisesti lokiviestien sisältö. Sisältöön ei ole olemassa omaa standardia, vaan se voi olla käytännössä mitä tahansa.
- Sovellus: Vastaa lokiviestien tuottamisesta, tulkinnasta, säilytyksestä ja reitityksestä.
- Kuljetus: Huolehtii siitä, että lokiviestit lähtevät ja niitä voidaan vastaanottaa.

(Gerhards 2009.)

Lisäksi jokaiselle tasolle on eritelty vielä omat funktionsa, joista kerrokset huolehtivat. Sovellustasolla funktiot ovat viestin lähde (originator), kerääjä (collector) sekä välittäjä (relay). Yhdessä nämä tuottavat sisällön viestiin, kokoavat viestit analysointia varten sekä vastaanottavat ja lähettää edelleen viestejä kerääjiltä tai muilta välit-

täjiltä. Kuljetustasolla sijaitsee lähettäjä- (transport sender) ja vastaanottajafunktio (transport receiver). Nimensä mukaisesti nämä funktiot lähettävät ja ottavat vastaan lokiviestejä. Sisältötasolla ei ole omaa funktiotaan. Tasojen ja funktioiden toiminta on vielä esiteltynä kuviossa 2. (Gerhards 2009.)



Kuvio 2. Syslogin tasot

Syslogin viesti koostuu aina kolmesta osasta, jotka ovat PRI, HEADER ja MSG. PRI-kenttä määrittelee viestin vakavuusarvon sekä sen, mikä sovellus tai sovellustyyppi on tuottanut viestin. HEADER-kentän tulisi sisältää uusimman RFC standardin mukaan *VERSION-, TIMESTAMP-, HOSTNAME-, APP-NAME-*, *PROCID-* ja *MSGID*-kentät, joilla viestejä voidaan jaotella jatkokäsittelyä tehdessä. MSG-kenttä sisältää nimensä mukaisesti vapaamuotoisen viestin, jolla yleensä pyritään kuvamaan lokiviestin aiheuttanutta tapahtumaa. (Gerhards 2009.)

3 Lokilähteet

3.1 Palomuuri

3.1.1 Yleistä

Palomuuri on verkkolaite, joka valvoo sekä sisään- että ulospäin menevää liikennettä ja päättää sille määriteltyjen asetusten pohjalta, sallitaanko liikenne vai ei. Palomuuri ei siis itse pysty päättelemään, mikä liikenne on haitallista, vaan vaatii aina esimääritellyt säännöt. Luonteensa vuoksi palomuurin läpi kulkee kaikki liikenne sisäverkon ja ulkoverkon välillä. Tämä mahdollistaa myös lokitiedon keräämisen palomuurilta koskien sekä ulko- että sisäverkonliikennettä. (What is a firewall N.d.)

Palomuurilla on useita tietoverkon turvallisuuteen liittyviä tehtäviä, mutta näistä olennaisimmat ovat seuraavat tehtävät:

- IP-osoitteiden muunnos ja reititys.
    - IP-muuttaminen, eli NAT (Network Address Translation) mahdollistaa useiden sisäverkkojen luomisen, vaikka käytössä ei olisi kuin yksi julkinen IP-osoite. Reititys taas mahdollistaa liikenteen näiden luotujen sisäverkkojen välillä.
- Tietoverkkojen eriyttäminen.
    - Palomuurilla luodaan rajat, mistä verkoista voidaan liikennöidä mihinkin verkkoon. Esimerkiksi yrityksen oma toimistoverkko ja vierasverkko on hyvä eriyttää siten, että vierasverkosta ei päästä toimistoverkon laitteisiin käsiksi.
- IP-osoitteiden ja porttien suodattaminen.
    - Palomuurit suodattavat liikennettä lukemalla IP-pakettien osoitteita ja portteja OSI-mallin tasoilla 3 ja 4. Palomuurille voidaan määritellä esimerkiksi web-palvelin, johon ei haluta pääsyä kuin tietystä IP-aliverkosta ja ainoastaan HTTPS-porttiin. Tällöin tulee estää kaikki portit ja lähde IP-osoitteet, pois lukien HTTPS-portti 443 ja toivotut lähde IP-osoitteet.
- IP-pakettien porttiohjaus
    - Mahdollistaa esimerkiksi ulkoverkosta pääsyn sisäverkon osoitteisiin käyttäen palomuurin julkista IP-osoitetta. Mahdollistaa myös esimerkiksi kaiken suojaamattoman HTTP-liikenteen pakottamisen salatuksi HTTPS-liikenteeksi muuttamalla kaikki yhteydet porttiin 80 menemään porttiin 443.
- Lokitus
    - Yksi olennaisimmista, mutta usein unohdettu palomuurin ominaisuus. Lokitus on tärkeää, jotta voidaan selvittää miksi jokin ei toimi (onko liikenne väärin estetty muurilla?) tai vaihtoehtoisesti voidaan havaita, jos johonkin pääsee epätoivottua liikennettä (estosääntö ei ole tarpeeksi kattava).

(Amon, Amon & Shimonski 2003, 54-55)

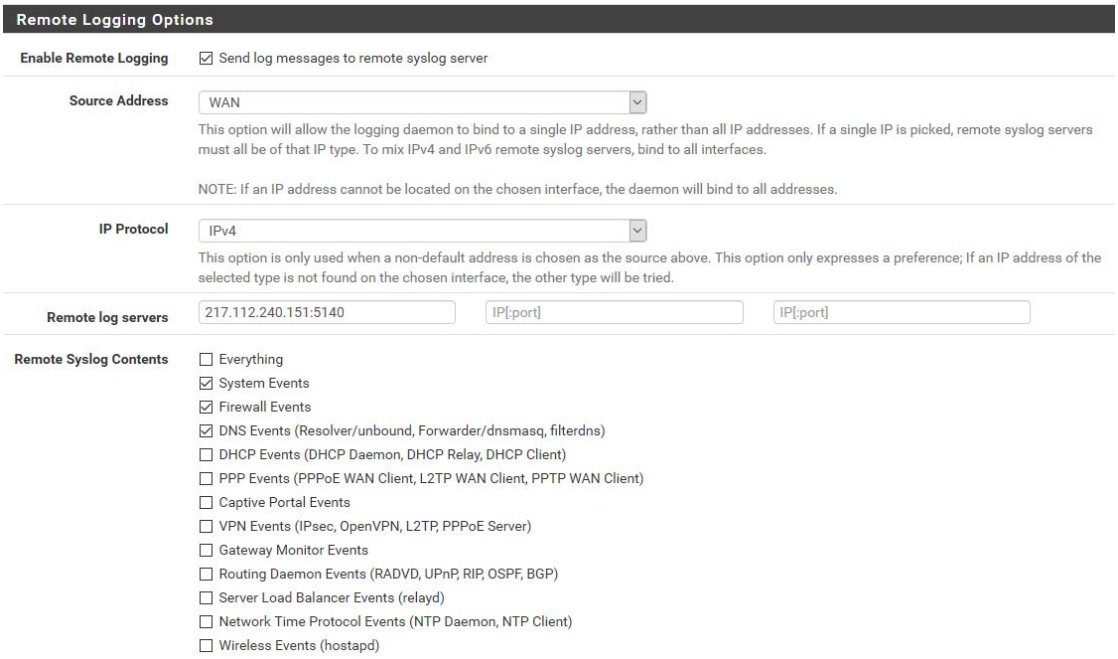
Palomuurityyppejä on muutamia, mutta näistä nykypäivänä yleisin on tilallinen palomuuri. Tilallisuus nopeuttaa palomuurin toimintaa ja vähentää palomuurin tarvit-

semia resursseja, sillä sen ei tarvitse käsitellä aivan jokaista läpi kulkevaa IP-pakettia. Tilallinen palomuuri lukee muiden palomuurien tapaan IP-paketeista ainoastaan IP-osoitteet sekä porttinumerot ja tekee suodatuspäätökset näiden perusteella. Tilallinen muuri on kuitenkin kykenevä muistamaan, jos sisäverkosta yhteyden muodostanut laite vastaanottaa ACK-paketin TCP-protokollaa käyttäessä. Tällöin palomuuri voi olettaa, että yhteys on haluttu, eikä käsittele tätä liikennettä enää jatkossa, ennen kuin tila lopetetaan FIN-paketilla. Tiloja voidaan tavallaan hyödyntää UDP-protokollaa käyttäessä, sillä palomuuri tekee omaan yhteystauluunsa merkinnän, minkä kahden laitteen välillä UDP liikennettä on havaittu ja hyväksyy nämä automaattisesti jatkossa. (Amon ym. 2003, 56-57).

3.1.2 pfSense

TNNetin käyttämä palomuuriratkaisu on avoimeen lähdekoodiin perustuva pfSense. PfSense pohjautuu FreeBSD-käyttöjärjestelmään, joka on vapaasti kustomoitavissa vastaamaan sekä TNNetin että loppuasiakkaiden tarpeita. Tämä tekee järjestelmästä erittäin joustavan ratkaisun. Järjestelmään on lisäksi saatavilla useita eri laajennuspaketteja, joilla pfSenseä voi käyttää myös esimerkiksi DHCP- ja VPN- (Virtual private network) palvelimena. PfSenseä voidaan hallita joko konsoliruudulla tai web-käyttöliittymällä, mikä tekee siitä hyvin käyttäjäystävällisen järjestelmän.

Jotta pfSensen syslogia voitiin analysoida ja koostaa saadusta datasta raportti, tuli lokidata saada siirrettyä jollekin keskitetylle palvelimelle. Tämä onnistui suoraan pfSensen lokiasetuksista kytkemällä päälle *”remote logging”*. Asetuksiin määriteltiin halutun kohdepalvelimen IP-osoite ja portti sekä mitä lokidataa haluttiin lähettää. Vaatimusmäärittelyn mukaisesti valittiin järjestelmä, palomuuri sekä nimipalvelun (Domain Name Service, DNS) tapahtumat (ks. Kuvio 3). Lisäksi palomuurisäännöistä tuli kytkeä lokitus päälle niihin sääntöihin, joissa se ei jo entuudestaan ollut.



Kuvio 3. pfSense lokiasetukset

3.2 IDS ja Snort

3.2.1 Yleistä

Tunkeutumisen havaitsemisjärjestelmän (Intrution Detection System, IDS) tehtävä on tarkkailla kaikkea sen ”läpi” kulkevaa liikennettä ja tehdä hälytys, mikäli se havaitsee epäilyttävää tai vaarallista liikennettä. IDS:ää ei tule kuitenkaan sekoittaa palomuureihin tai tunkeutumisen estojärjestelmiin (Intrution Prevention System, IPS), jotka tarkkailun lisäksi myös estävät liikennettä. IDS:n ei siis tarvitse puuttua liikenteeseen millään tavalla, vaan ainoastaan tarkkailla sitä ja pyrkiä havaitsemaan mahdollisia uhkia muodostaen samalla havainnoistaan lokimerkintöjä tai jopa suoria hälytyksiä järjestelmänvalvojille.

Vaikka IDS kuulostaa hieman samalta kuin palomuurit, on niiden toimintaperiaate hyvinkin erilainen. Palomuuri valvoo liikennettä ainoastaan sille annettujen määräysten mukaan, eli käytännössä palomuuri mahdollistaa ainoastaan tunnettujen uhkien estämisen. IDS puolestaan analysoi liikennettä IP-pakettien otsikoiden sisältä yrittäen tunnistaa mahdollisia uhkia tietoliikenteestä. Näitä voivat olla esimerkiksi liikennöinti tunnettuihin bottiverkkoihin, SQL-injektiot tai TOR-liikenne. (Shiner 2005.)

Palomuurit eivät siis yksinään voi estää kaikkia mahdollisia uhkia, eikä IDS voi luonteensa mukaisesti estää havaitsemiaan uhkia. Näin ollen yhdistämällä molemmat komponentit saadaan aikaan hyvä ratkaisu sekä tunnettujen, että tuntemattomien uhkien havaitsemiseen ja estämiseen.

IDS voi olla joko konekohtaista (host-based) tai tietoverkkokohtaista (network-based). Konekohtaisessa lähestymistavassa IDS asennetaan yhteen järjestelmään, jonka dataa käytetään tunkeutumisten havaitsemiseen. Tämä vaihtoehto on parempi yhden tietokoneen suojaamiseen, sillä se vertaa liikennettä kyseisen koneen normaaliin liikenteeseen ja etsii koneesta odottamattomia tapahtumia, jotka voisivat viitata hyökkäyksiin. Tietoverkkokohtainen lähestymistapa taas on kattavampi, sillä se voi analysoida dataa koko lähiverkosta. Tietoverkkokohtainen IDS analysoi IP-pakettien otsikoita ja niiden sisältöjä. Tämä mahdollistaa konekohtaisten hyökkäysten havainnoimisen lisäksi verkkohyökkäykset, kuten esimerkiksi palvelunestohyökkäyksen. (Shiner 2005.)

3.2.2 Snort

Snort on yksi monista IDS työkaluista. Snortin tapa toteuttaa IDS:ää on tietoverkkokohtainen, eli sillä pystytään valvomaan koko lähiverkkoa. Snort pystyy tekemään reaaliaikaista IP-pakettien analysointia ja pystyy etsimään niistä uhkamalleja, joilla voidaan tunnistaa useita erilaisia verkkohyökkäyksiä. Snort saa käyttämänsä mallit joko suoraan Snortin kehittäjiltä tai yhteisön tekeminä. Suosittu tapa onkin yhdistellä näitä malleja, jolloin saadaan aikaan todennäköisesti hyvin ylläpidetty ja päivitetty lista tunnetuista hyökkäysmalleista. (Snort Official Documentation N.d.)

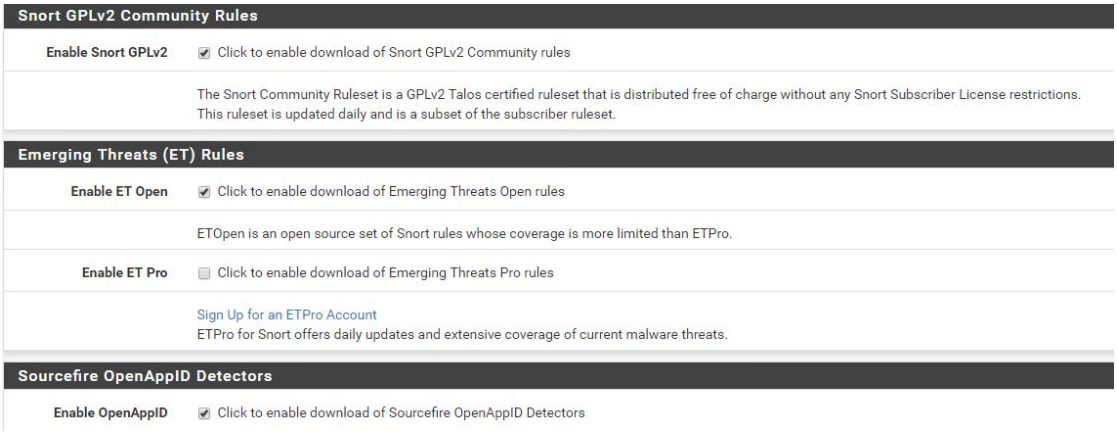
Snortin uhkamallien tunnistamisellakaan ei kuitenkaan voida havaita uusimpia hyökkäyksiä, joiden sormenjälkeä ei vielä olla saatu selville. Tähän ratkaisuna olisi tilastollinen liikenteen vertailu, jossa yritetään havaita poikkeamia normaalista verkkoliikenteestä. Tämä tapa tuottaa kuitenkin enemmän vääriä hälytyksiä kuin mallien tunnistaminen, eikä Snort tue tätä havaitsemistapaa. (Shiner 2005.)

Tässä opinnäytetyössä päätettiin kuitenkin käyttää Snortia, sillä se on suoraan saatavilla lisäosana pfSensen palomuureihin. Lisäksi työn tarkoituksena oli teettää raportti,

josta voitaisiin manuaalisesti etsiä tilastollisia kummallisuuksia. Raportin avulla voitaisiin siis käsin estää nämä tapaukset jälkikäteen, mikä todettiin riittäväksi.

Snortin asennus pfSenselle onnistui hyvin yksinkertaisesti pfSensen omasta pakettien hallinnasta. Kun Snort oli asentunut palomuurille, se piti vielä kuitenkin konfiguroida käyttöön. Tätä varten tuli aluksi valita rajapinta, jota halutaan valvoa. Yleisesti IDS kannattaa sijoittaa sisäverkkoon, joten palomuurilta päätettiin valita yksi sisäverkon rajapinnoista, johon Snort lisättiin. Tämän jälkeen Snortiin tuli vielä konfiguroida valvottavat säännöt sekä asettaa Snort-prosessi päälle.

Säännöistä päätettiin ottaa GPLv2, ET open sekä OpenAppID (ks. Kuvio 4). Nämä ovat kaikki ilmaisia, mutta osaan on saatavilla myös maksulliset Pro-versiot. Maksullisen version etuna ovat ainoastaan hieman nopeammat päivitykset uusiin uhkiin. GPLv2 säännöstö on Snortin yhteisön tekemä säännöstö, joka on käytännössä sama, kuin Snort VRT:n (Snortin kehittäjän luoma säännöstö) maksullinen säännöstö. Erona on ainoastaan muutaman viikon myöhäisempi päivitysaikataulu. ET Open on ET Labsin luoma säännöstö, ja OpenAppID valvoo yleisesti tunnettujen ohjelmistojen käyttöä verkossa.

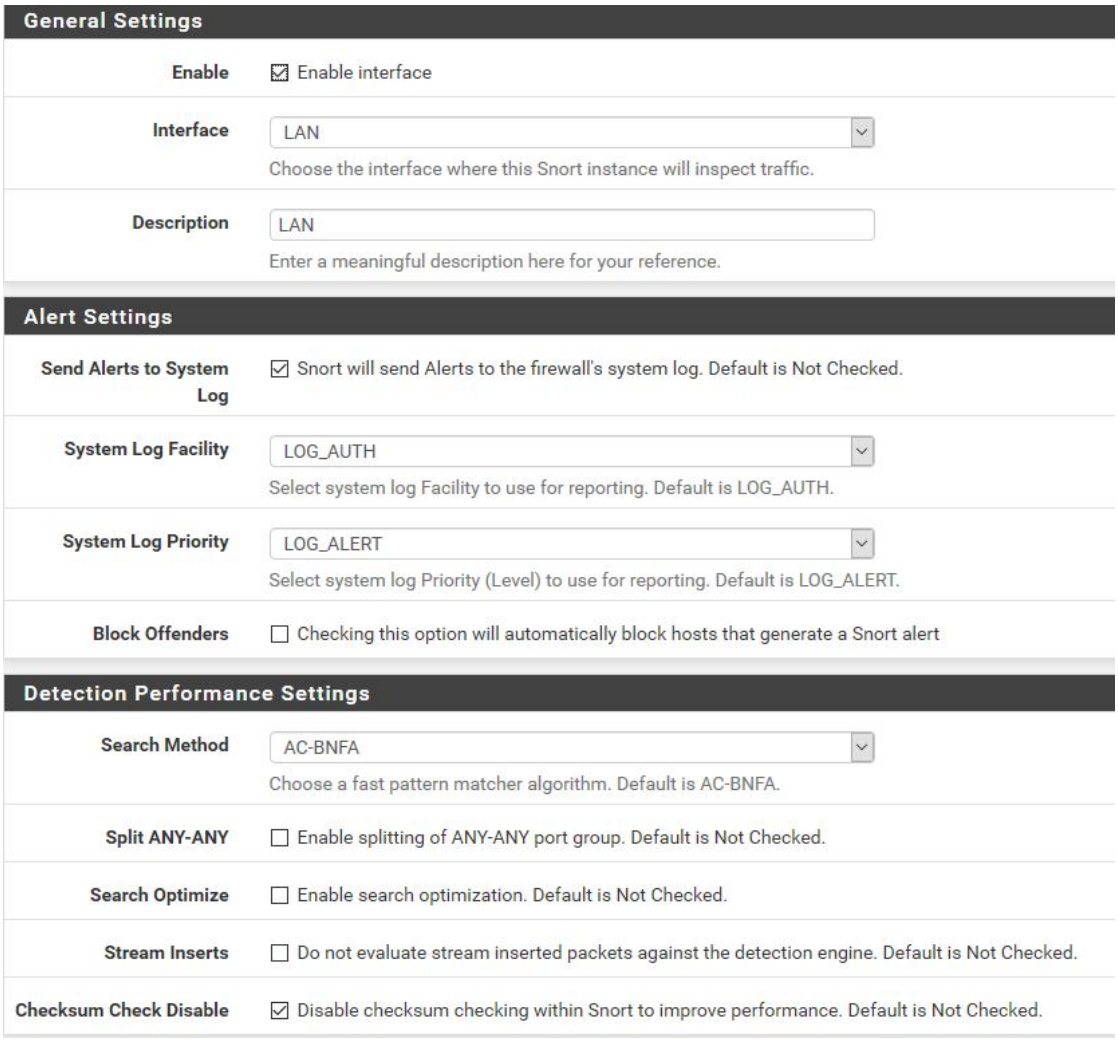


Kuvio 4. Snort säännöstöjen asennus

Näiden lisäksi voitiin vielä valita haluttu päivitysväli säännöstöille. Työssä valittiin päivitysten väliksi yksi vuorokausi ja ajastettiin päivitysten tarkistus alkamaan keskiyöllä, jolloin muurin mahdollinen hidastelu ei todennäköisimmin aiheuttaisi asiakasimpaktia. Kun nämä olivat konfiguroitu, tuli Snort vielä pakottaa tekemään säännöstöjen päivitys *”Updates”* -välilehdellä.

Snortiin voitaisiin konfiguroida myös Pass Lists, IP Lists sekä Suppress lists, mutta nämä jätettiin tyhjäksi, sillä tiedossa ei ollut mitä hälytyksiä verkosta voisi löytyä ja mitkä niistä olisivat turhia. Näillä voitaisiin kuitenkin hiljentää väärät hälytykset, kun sellaisia havaitaan. Väärille hälytyksille ei voitu myöskään tehdä mitään oletuslistaa, sillä väärät hälytykset ovat hyvin palomuurikohtaisia.

Kun kaikki yleiset konfiguraatiot olivat kunnossa, palattiin vielä konfiguroimaan sisäverkon rajapinnan asetukset kuntoon (ks. Kuvio 5). Rajapinnan asetuksista Snort käynnistettiin valitsemalla *”Enable interface”* ja asetettiin Snort laukaisemaan lokiviestit suoraan syslogiin, jotta ne voitaisiin myöhemmin lähettää palomuurin syslogista eteenpäin. Lisäksi poistettiin IP-pakettien summatarkisteiden valvonta, sillä havaittiin, että Snort vei hyvin paljon palomuurin resursseja tämän ollessa päällä.



Kuvio 5. Snort rajapinnan asetukset

Kun kaikki saatiin konfiguroitua, voitiin palomuurin syslogista tarkistaa, onko sinne ilmestynyt Snortin tuottamia hälytyksiä. Kuvion 6 mukaan havaittiin, että listasta löytyi heti sekä ET openin että GLPv2 säännöstöjen hälytyksiä, joten Snort todettiin toimivaksi.

Last 50 General Log Entries. (Maximum 50)

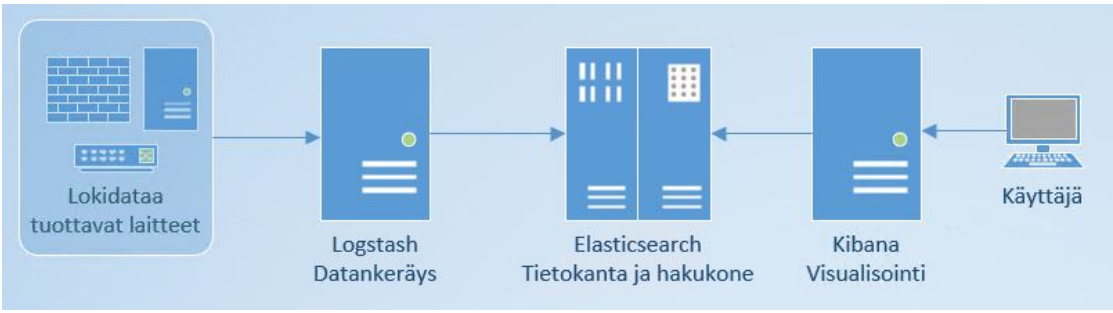
Time	Message
Apr 28 17:19:20	snort[73900]: [1:2403498:40135] ET CINS Active Threat Intelligence Poor Reputation IP TCP group 100 [Classification: Misc Attack] [Priority: 2] {TCP} 104.148.42.208:45484 -> [redacted]:8080
Apr 28 17:14:31	snort[73900]: [1:2402000:4791] ET DROP Dshield Block Listed Source group 1 [Classification: Misc Attack] [Priority: 2] {TCP} 80.82.77.139:31802 -> [redacted]:993
Apr 28 17:14:31	snort[73900]: [1:2403432:40135] ET CINS Active Threat Intelligence Poor Reputation IP TCP group 67 [Classification: Misc Attack] [Priority: 2] {TCP} 80.82.77.139:31802 -> [redacted]:993
Apr 28 17:13:41	snort[73900]: [120:3:1] (http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE [Classification: Unknown Traffic] [Priority: 3] {TCP} 107.191.41.71:80 -> [redacted]:40191

Kuvio 6. Snort hälytyksiä syslogissa

4 ELK-Stack

4.1 Yleistä

ELK-Stack on akronyymi kolmelle eri avoimen lähdekoodin projektille, jotka ovat Elasticsearch, Logstash ja Kibana. Yhdistettynä nämä työkalut mahdollistavat erilaisen datan, kuten esimerkiksi verkkoliikenne tai palvelimen prosessidatan keräämisen, analysoimisen ja visualisoimisen. ELK-Stackin keskeisin komponentti on Elasticsearch, joka toimii tietokantana ja hakukoneena. Logstash taas lähettää tiedon Elasticsearchiin, kun Kibana tekee hakupyyntöjä ja visualisoi tulokset (ks. Kuvio 7). (Elk-Stack N.d.)



Kuvio 7. ELK-Stackin toimintaperiaate

Lokidataa generoidaan usein liian paljon paljaaltaan luettavaksi, joten sitä tulee käsitellä ja muokata jotenkin, ennen kuin lokeja voidaan hyödyntää. Erityisesti visualisointi on tärkeää, sillä ihmisaivot kykenevät käsittelemään monimutkaista dataa paremmin väreillä ja muodoilla visualisoituna, kuin esimerkiksi numeerisista taulukoista luettuna (Data Visualization N.d.).

4.2 Elasticsearch

4.2.1 Yleistä

Elasticsearch on hakutyökalu, jota käytetään pääasiassa sanojen etsimiseen isoista tekstimääristä ja hakutulosten tallentamiseen optimoituun muotoon. Käytännössä Elasticsearchia voidaan käyttää myös muihin hakutehtäviin, mutta parhaimmillaan se on juurikin isojen tekstikokonaisuuksien käsittelyssä kun halutaan tehdä nopeita hakuja (Cholakian 2013). Elasticsearch on lähes reaaliaikainen, mikä tarkoittaa sitä, että kerätty data ilmestyy haettavaksi hyvin pienellä viiveellä tallennushetkestä, yleensä alle sekunnissa (Elasticsearch Reference N.d.).

Tarkemmin kuvailtuna Elasticsearch on Javalla koodattu tietokantapalvelin, joka vastaanottaa raakadatan ja säilöö sen optimoidussa muodossa jatkokäsittelyä ajatellen. Elasticsearchin tallentaman datan jatkokäsittely onkin helppoa, sillä Elasticsearchiin on implementoituna HTTP- ja JSON-tuki, joita useat nykyajan web-sovellukset käyttävät. (Cholakian 2013.)

Elasticsearch perustuu jo vuonna 1999 luotuun Apache Luceneen. Elasticsearch käyttää samoja algoritmejä ja Java-kirjastoja, kuin Lucene. Elasticsearchin tuoma lisäarvo Luceneen verrattuna on helpommin käytettävissä oleva rajapinta, hakutyökalut sekä skaalautuvuus, jonka mahdollistavat Elasticsearchin klusterointi- ja replikointimahdollisuudet. (Cholakian 2013.)

4.2.2 Rakenne

Elasticsearch koostuu useasta eri palasesta, jotka voidaan jaotella klustereihin, solmuihin, hakemistoihin, tiedostoihin sekä siruihin ja replikoihin.

Klusteri on joko yhden tai useamman solmun, eli palvelimen kokonaisuus, joka säilöö tallennetun datan jaettuna tasaisesti kaikille klusterin solmuille. Elasticsearchissa klusteri tunnistetaan uniikilla nimellä, joka on oletuksena ”elasticsearch”. Jokaiselle klusterille on annettava uniikki nimi, sillä tällöin klusteriin ei voi liittyä kuin ne solmut, jotka ovat määriteltyjä liittymään johonkin klusteriin nimen perusteella. (Elasticsearch Reference N.d.)

On tärkeää pitää huoli siitä, että kahdelle klusterille ei vahingossakaan anneta samaa nimeä. Tämä mahdollistaisi solmujen liittymisen vääriin klustereihin, jonka seurauksena arkaluontoistakin lokidataa voisi joutua väärän klusterin ja tätä käsittelevän henkilön nähtäville. Vaikka klusteri yleensä mielletäänkin usean palvelimen kokonaisuudeksi, Elasticsearchissa on aivan normaalia, että klusteri on vain yhden solmun kokoinen. (Elasticsearch Reference N.d.)

Solmu on yksittäinen palvelin jossakin klusterissa. Solmun tehtävänä on tallettaa dataa, sekä osallistua klusterin pyytämiin datan jäsennys- ja hakutehtäviin. Aivan kuten klusteritkin, myös solmut nimetään uniikisti. Solmut saavat oletuksena satunnaisesti generoidun universaalin uniikin tunnisteen (Universally Unique Identifier UUID), joka annetaan solmulle kun se käynnistetään. Solmut voidaan nimetä myös käsin, joka onkin järkevää. Käsin nimeäminen helpottaa tunnistamaan, mitkä palvelimet ovat osana mitäkin klusteria. (Elasticsearch Reference N.d.)

Solmu voidaan konfiguroida liittymään haluttuun klusteriin määrittelemällä solmun konfiguraatioon halutun klusterin nimi. Oletuksena jokainen solmu liittyy klusteriin, jonka nimi on *”elasticsearch”*, eli klusterin oletusnimi. Tämän seurauksena kaikki solmut liittyvät automaattisesti yhteen ja samaan klusteriin, *elasticsearchiin*. Elasticsearchin klusterin ja solmun oletuskonfiguraatio siis muodostaa aina yhden klusterin, *”elasticsearch”*, johon kaikki solmut liittyvät automaattisesti. (Elasticsearch Reference N.d.)

Hakemisto on kokoelma tiedostoja, jotka omaavat joitakin samoja piirteitä tai kenttiä. Esimerkiksi asiakashakemisto sisältäisi tiedostoja, joissa on asiakasdataa ja tuotehakemisto taas sisältäisi ainoastaan tiedostoja, joissa on tuotedataa. Hakemisto tunnistetaan nimellä, jonka tulee olla kirjoitettuna kokonaan pienillä kirjaimilla. Nimeä käytetään, kun klusterilla suoritetaan haku-, päivitys- tai poistotoimintoja hakemis-

ton tiedostoja kohtaan. Hakemistot eivät ole klusterikohtaisia, eli jokaisessa klusterissa voi olla rajattomasti hakemistoja. (Elasticsearch Reference N.d.)

Tiedosto on yksi tietue, joka voidaan laittaa hakemistoon. Tällainen tietue voisi olla esimerkiksi syslogia kerättäessä yksi lokitietue, joka voidaan tunnistaa esimerkiksi lokin aikaleimalla tai jollakin muulla identifioivalla osiolla. Tiedostot tallennetaan JSON muodossa, jota on helppo käsitellä useilla eri web-protokolilla. Yksi hakemisto voi sisältää rajattomasti tiedostoja. (Elasticsearch Reference N.d.)

Sirut ovat pieniä osia yksittäisistä hakemistoista. Sirut ovat luotu ratkaisemaan ongelma suurien hakemistojen kanssa, joista hakutoimintojen suorittaminen olisi erittäin hidas ja raskas prosessi. Esimerkiksi yksi hakemisto voi sisältää useita miljoonia tiedostoja, jotka vievät levytilaa yli teratavun verran. Tällainen hakemisto voi olla liian suuri yhden solmun käsiteltäväksi, joten se tulee jakaa pienempiin osioihin, siruihin. Sirut voidaan jaotella klusterin kaikille solmuille tasaisesti. Kun hakemisto on siroteltu kaikille klusterin solmuille, voi hakutoiminnon viemä aika pienentyä suunnilleen sirujen lukumäärällä jaettavaan murto-osaan. (Elasticsearch Reference N.d.)

Replikat ovat kahdennettuja siruja. Replikat siis ovat nimensä mukaisesti täydellisiä kopioita siruista, jotka ovat jaoteltuna alkuperäiseen siruun nähden eri solmuille. Tämän ansiosta data on käytettävissä, vaikka ongelmatilanteessa alkuperäisen sirun omaava solmu tippuisi offline -tilaan. Oletuksena Elasticsearch luo jokaiselle hakemistolle viisi sirua ja yhden replikan, joka tarkoittaa toista viittä sirua, luoden siis yhteensä kymmenen sirua. Replikat vaativat kuitenkin toimiakseen vähintään kaksi solmua, jotta alkuperäiset sirut ja replikat voidaan jaotella eri solmuihin. Mikäli solmuja ei ole kuin yksi, voidaan Elasticsearchia konfiguroidessa määritellä, että replikoita ei luoda ollenkaan. (Elasticsearch Reference N.d.)

4.2.3 Konfiguraatio

Elasticsearchin oletuskonfiguraatio on elasticsearch.yml, joka oli oletuksena lähes täysin valmis käytettäväksi tässä työssä käytettävässä yhden klusterin ja silmun järjestelmässä. Aiemmin käsitellyn mukaisesti oletuskonfiguraatio luo klusterin nimeltä *elasticsearch* sekä lisää siihen automaattisesti viisi sirua per hakemisto. Ainoat muutokset tähän oletuskonfiguraatioon olivat verkkoasetukset kuvion 8 mukaisesti.

```
# ---------------------- Network ---------------------
network.host: 217.112.240.151
http.port: 9200
```

Kuvio 8. Elasticsearch.yml

Elasticsearchin luodessa oletusarvoisesti myös viisi sirua replikoita varten, asetettiin replikointi vielä pois päältä (ks. Kuvio 9). Näin palvelimen resursseja ei tule allokoitua turhaan käyttämättömille siruille.

```
root@elkpfsense:~# curl -XPUT -H 'Content-Type: application/json'
http://217.112.240.151:9200/_settings -d '

{
    "index" : {
        "number_of_replicas" : 0
    }
}'
{"acknowledged":true}
```

Kuvio 9. Elasticsearch-replikoiden poisto

Tämän lisäksi klusterin suorituskykyä parannettiin allokoimalla Elasticsearchin käyttämälle Java-virtuaalikoneelle (JVM, Java Virtual Machine) lisää keskusmuistia (RAM, Random Access Memory). Palvelimella itsellään oli yhteensä 16GB keskusmuistia, joten tästä voitiin turvallisesti allokoida Elasticsearch JVM:lle 12GB, jolloin palvelimen muille prosesseille jäi vielä tarpeeksi muistia prosessien ylläpitämiseen. Muistin lisäys tehtiin jvm.options tiedostossa (ks. Kuvio 10). Xms ja Xmx arvojen suositellaan olevan samankokoisia, sillä muutoin Javan kanssa saattaa ilmetä ongelmia (Elasticsearch Reference N.d.).

```
# Xms represents the initial size of total heap space
# Xmx represents the maximum size of total heap space

-Xms12g
-Xmx12g
```

Kuvio 10. Elasticsearch jvm.options

4.3 Logstash

4.3.1 Yleistä

Logstash on avoimen lähdekoodin datankeräysohjelmisto, joka kykenee "putkittamaan" kerättyä dataa reaaliajassa useisiin eri pipelineihin. Tämä mahdollistaa datan yhdentämisen ja normalisoinnin dynaamisesti useista erilaisista lähteistä käyttäjän haluamiin kohteisiin, esimerkiksi analytiikkapalveluun, kuten Elasticsearchiin. Logstash kykenee käsittelemään kaikkia tunnetuimpia lokidatan muotoja, kuten esimerkiksi syslogia, Windows event logia ja NetFlowia. (Logstash Reference N.d.)

Logstashin pipelinet sisältävät kaksi pakollista konfiguraatioelementtiä: Input eli sisääntulo ja output eli ulostulo. Näiden lisäksi pipelinessä voi olla myös yksi vapaaehtoinen elementti, filter, eli suodatin. Logstashin pipelinen toiminta on kuvattuna myös kuviossa 11. Sisääntulossa data otetaan vastaan määritellyistä lähteistä ja porteista. Käytetyimmät sisääntulot ovat:

- Tiedosto (file): Lukee dataa järjestelmässä olevasta tiedostosta, hyvin samantapainen kuin UNIX:n komento ”tail -F”
- Syslog: Kuuntelee oletuksena porttia 514, johon syslog viestit lähetetään RFC3164 standardin mukaisesti.
- Redis: Lukee redis-palvelimelta dataa käyttäen sekä redis kanavia että listoja. Käytetään usein jakamaan dataa usean eri Logstash instanssin kesken isoissa Logstash-klustereissa.

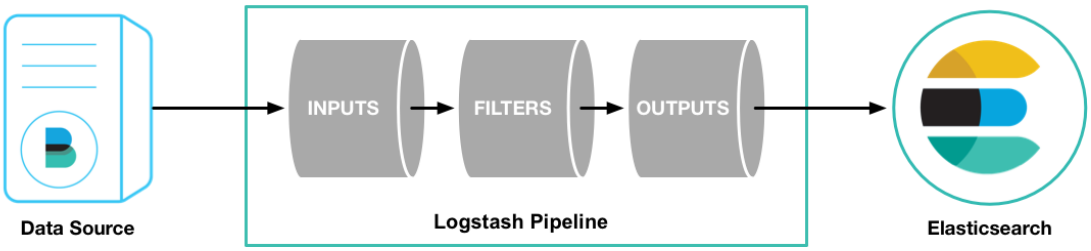
(Logstash Reference N.d.)

Suodattimissa vastaanotettua dataa voidaan muokata useita erilaisia suodattimia yhdistelemällä. Suodattimiin voidaan asettaa konditionaalisia ehtoja, joiden täyttyessä tehdään muutoksia. Vaihtoehtoisesti voidaan tehdä muutoksia jokaiselle sisääntulolle. Hyödyllisimpiä suodattimia ovat seuraavat:

- Grok: Käytetään jäsentymättömän ja muokkaamattoman raakadatan muokkaamiseen siten, että sitä on helppo indeksoida ja hakea jatkossa. Logstash sisältää 120 sisäänrakennettua mallia, joilla Grokia voidaan konfiguroida. Lisäksi näitä malleja voi luoda itse lisää.
- Mutate: Mahdollistaa parametrien muuttamisen esimerkiksi uudelleennimeämällä, poistamalla tai vaihtamalla.
- Drop: Pudottaa sisääntulon kokonaan.
- Clone: Tekee kopion sisääntulosta, samalla mahdollisesti lisäten tai poistaen parametrejä.
- GeoIP: Lisää IP-osoitteen perusteella sisääntulolle geolokaation.

(Logstash Reference N.d.)

Ulostulossa suodatettu data voidaan lähettää seuraavaan kohteeseen, esimerkiksi tietokantaan tai hakukoneeseen. Logstashin valmistajan suosittelema hakukone on Elasticsearch, mutta myös esimerkiksi graphite tai statsd olisivat mahdollisia. Ulostuloja voi olla useita, mutta kun kaikki ulostulot on käyty läpi, on yksi Logstashin pipeline päättynyt. (Logstash Reference N.d.)



Kuvio 11. Logstash Pipeline (Logstash Reference N.d.)

4.3.2 Konfiguraatio

Logstashin yleiskonfiguraatiotiedostoon, *logstash.yml,* ei tarvinnut tehdä mitään muutoksia, vaan se kelpasi oletusarvoilla sellaisenaan. Konfiguraatio aloitettiin siis konfiguroimalla tarvittava pipeline tiedostossa *”pipelines.yml”*. Tiedostoon on konfiguroitu ainoastaan yksi pipeline, joka oletuksena saa tunnisteen *”main”*. Tälle pipelinelle on sen jälkeen määritelty tiedostopolku, jossa pipelinen konfiguraatiotiedostot sijaitsevat (ks. kuvio 12). Tässä tapauksessa tiedostot sijaitsevat alihakemistossa *conf.d/* ja tiedoston tulee päättyä *.conf* -päätteeseen.

```
- pipeline.id: main

  path.config:
"/etc/logstash/conf.d/*.conf
"
```

Kuvio 12. Logstash pipeline.yml

Conf.d/ alihakemisto sisältää kuvion 13 mukaiset tiedostot. Nämä tiedostot käsitellään nimenmukaisessa aakkosjärjestyksessä, minkä vuoksi tiedostot on nimetty aloittamalla ne numeerisesti. Näin ollen ensimmäiseksi luetaan tiedosto 01-inputs.conf, minkä jälkeen jatketaan järjestyksessä tiedostopolun viimeiseen tiedostoon, 30-

outputs.conf. Välissä olevat tiedostot ovat Logstashin pipelinen periaatetta noudattaen suodatintiedostoja.

```
root@elkpfsense:/etc/logstash/conf.d# ls
[  01-inputs.conf  10-syslog.conf  11-pfsense.conf  15-snort  30-outputs.conf  en  patterns
root@elkpfsense:/etc/logstash/conf.d#
```

Kuvio 13. conf.d tiedostohakemiston konfiguraatiotiedostot

01-inputs.conf tiedostoon on nimenmukaisesti määritelty sisääntulon parametrit. Aiemmin määriteltiin pfSense lähettämään syslog dataa käyttäen UDP porttia 5140, joten riittää, että tämä määritetään myös Logstashin konfiguraatioon (ks. Kuvio 14).

```
#udp syslogs stream via 5140

input {
  udp {
    type => "syslog"
    port => 5140
  }
}
```

Kuvio 14. Logstash 01-inputs.conf

10-syslog.conf -tiedostossa suodatetaan tyypillinen, standardisoitu syslog lokidata. Konfiguraatiotiedosto löytyy kokonaisuudessaan liitteestä 1. Konfiguraatiotiedoston alussa varmistetaan, että tietueesta löytyvät jatkokäsittelyä varten oikeat tagit, syslog ja pfSense. Tämän jälkeen lisätään vielä tagit, joilla asiakas tunnistetaan dataa visualisoitaessa IP-osoitteen perusteella helpommin ja lopuksi Grok-suodatinta käyttäen muokataan syslogin raakadata helpommin luettavaan muotoon.

11-pfsense.conf -tiedostossa jatketaan käsitellyn datan muokkaamista. PfSensen tuottama syslogi sisältää aiemmin käsitellyn standardisoidun syslog datan lisäksi myös omaa, pfSenselle tyypillistä dataa. Tähän konfiguraatiotiedostoon hyödynnettiin käyttäjän *a3ilson* Github-repositorya (Pfelk N.d.). Konfiguraatiotiedosto löytyy liitteestä 2. Tämä konfiguraatiotiedosto vaatii toimiakseen myös siinä määritellyn Grok-mallin, joka konfiguraation mukaan sijaitsee hakemistossa */patterns/pfsense2-4.grok”*. Tämä tiedosto perustuu saman Github-käyttäjän konfiguraatioon kuin *11-pfsense.conf* (ks. Liite 3) (Pfelk N.d.).

15-snort.conf -tiedostossa suodatetaan Snortiin liittyvä data, mikäli tietue sellaista sisältää. Tämä konfiguraatiotiedosto löytyy liitteestä 4. Alussa siis tarkistetaan, löytyykö tietueesta termiä *”snort”*. Jos ei, tämä tiedosto hypätään kokonaan yli. Mikäli kyseessä oli Snortin generoima tietue, se käsitellään jälleen Grokilla luettavampaan muotoon. Tämän jälkeen tietueen tageja vielä muokataan visualisoinnin tarpeiden mukaan. Huomioitavaa on se, että tässä tiedostossa yritetään tehdä GeoIP:n lisäys uudelleen, sillä se on Snort-tietueiden tapauksessa epäonnistunut poikkeuksetta aiemmassa yrityksessä 11-pfsense.conf -tiedostossa. PfSensen raakadatasta on suoraan luettavissa tietueen lähdeosoite, mutta Snortin tietueissa liikenne ilmaistaan arvolla *”lähdeosoite -> kohdeosoite”*. Näin ollen GeoIP voidaan etsiä vasta, kun kyseisestä ilmaisusta on suodatettu IP-osoitteet erilleen lähde- ja kohdeosoitteisiin.

Viimeisimpänä konfiguraatiotiedostona tehtiin Logstash pipelinen viimeinen osio, ulostulo, 30-outputs.conf (ks. Kuvio 15). Data haluttiin lähettää aiemmin konfiguroituun Elasticsearchiin, joten se ohjataan Elasticsearchin osoitteeseen.

```
output {
          elasticsearch {
          hosts => ["http://217.112.240.151:9200"]
          index => "logstash-%{+YYYY.MM.dd}"}
}
```

Kuvio 15. 30-outputs.conf

4.4 Kibana

4.4.1 Yleistä

Kuten ELK-stackin aiemmatkin osat, myös Kibana on avoimen lähdekoodin työkalu. Kibana on datan analysointiin ja visualisointiin tehty työkalu, joka on suunniteltu käytettäväksi Elasticsearchin kanssa. Kibana tekee suurten datamassojen käsittelystä ja ymmärtämisestä helppoa sekä mahdollistaa nopean ja dynaamisen analysoimisen ja visualisoimisen graafisella käyttöliittymä. (Kibana Reference N.d.)

Kibanan avulla siis käsitellään aiemmin Elasticsearchiin lähetettyä dataa. Kibana ei itse enää kykene muuttamaan tätä dataa, vaan datan tulee olla halutussa muodossa jo ennen Kibanan käyttöä. Näin ollen Kibana vaatii toimiakseen vähintään Elasticsearchin, josta Kibana hakee datansa. Jotta dataa voitaisiin analysoida järkevästi, Kibana vaatii lisäksi työkalun, esimerkiksi Logstashiin, jolla dataa voidaan muokata ja parsia haluttuun muotoon ennen Elasticsearchiin lähettämistä. Kun data on halutussa muodossa, sitä voidaan analysoida ja visualisoida millä tahansa datasta löytyvillä parametreillä. Tämän dynaamisuuden ansiosta Kibanan avulla voidaan luoda esimerkiksi lokianalytiikkaa joko tekstin tai graafien muodossa, sekä aikajanoja eri tapahtumista (What is Kibana N.d.).

4.4.2 Konfiguraatio

Kibanan asentaminen ja konfiguroiminen edellytti, että Elasticseach oli asennettuna. Lisäksi tuli varmistaa, että asennettava Kibanan versio on yhteensopiva asennetun Elasticsearchin kanssa. Tässä tapauksessa Elasticsearch oli versio 6.2.4 (ks. Kuvio 16), joten Logstashista piti asentaa vähintään versio 5.6.0.

```
root@elkpfsense: curl -XGET '217.112.240.151:9200' -u elastic

Enter host password for user 'elastic':
{
  "name" : "TaLwiFt",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "BvRVo-DxTh6at-smfHwH4w",
  "version" : {
    "number" : "6.2.4",
    "build_hash" : "ccec39f",
    "build_date" : "2018-04-12T20:37:28.497551Z",
    "build_snapshot" : false,
    "lucene_version" : "7.2.1",
    "minimum_wire_compatibility_version" : "5.6.0",
    "minimum_index_compatibility_version" : "5.0.0"
  },
  "tagline" : "You Know, for Search"
}
```

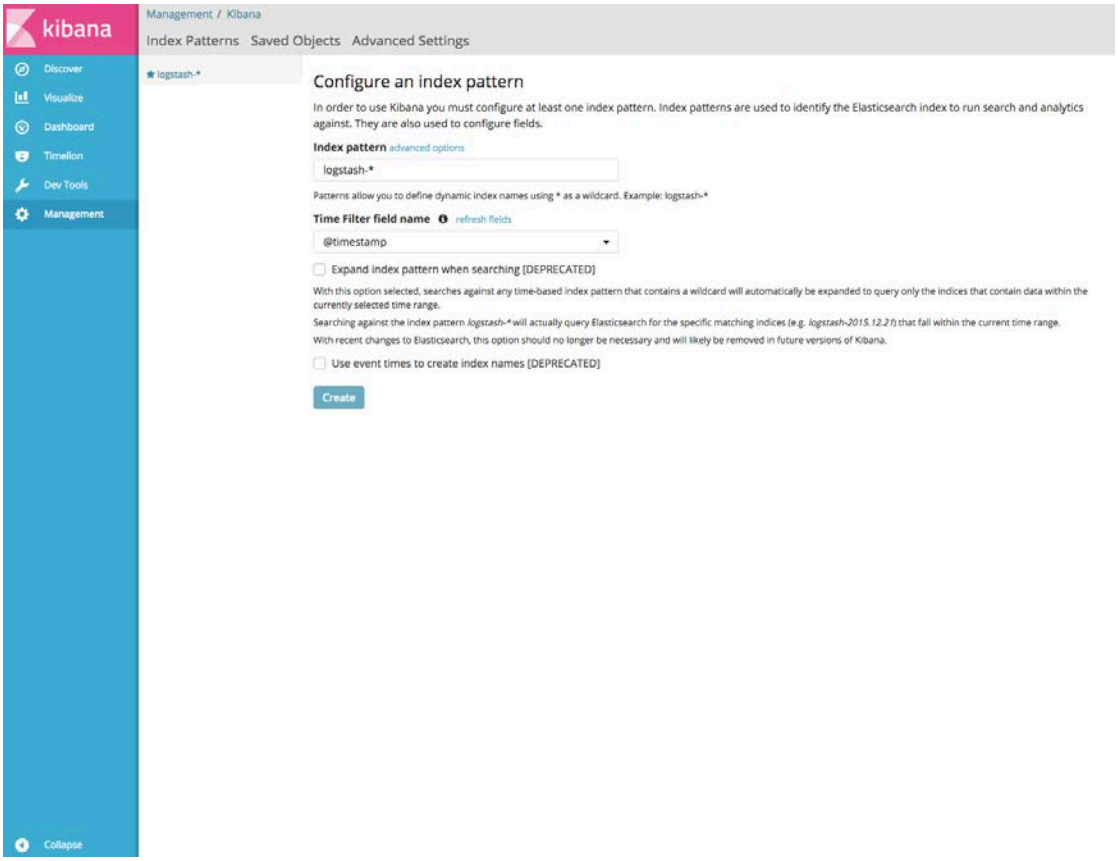
Kuvio 16. Elasticsearch versio

Kibana lukee käynnistyessään oletuksena konfiguraationsa tiedostosta *kibana.yml*, joka on jo oletuksena hyvin toimiva. Kibanan konfiguraatioon riittikin hyvin pienet muutokset, jotka koostettuna kuviossa 17.

```
#palvelimen käyttämä portti
server.port: 5601

#palvelimen osoite
server.host: "217.112.240.151"

#kasvatetaan oletusarvosta, jotta isot haut toimivat nopeammin
#server.maxPayloadBytes: 1048576
server.maxPayloadBytes: 10485760

#palvelimen nimi
# The Kibana server's name.  This is used for display purposes.
server.name: "elkpfsense"

#elasticsearch:n osoite ja portti
elasticsearch.url: "http://217.112.240.151:9200"

#kasvatetaan oletusarvosta, jotta isot haut eivät timeouttaa
#elasticsearch.requestTimeout: 30000
elasticsearch.requestTimeout: 60000
```

Kuvio 17. Kibana.yml

Kun Kibana saatiin konfiguroitua ja palvelu käynnistyi onnistuneesti, voitiin Kibanan toimivuus testata siirtymällä selailemalla *kibana.yml* -tiedostossa määriteltyyn osoitteeseen, eli tässä tapauksessa 217.112.240.151:5601, joka ohjaa Kibanan käyttöliittymään (ks. Kuvio 18).



Kuvio 18. Kibana käyttöliittymä

4.5 X-Pack

X-Pack on ELK-Stackin lisäosa, joka tuo pääosin lisäominaisuuksia Kibanaan ja Elasticsearchiin. X-Packin hyödyllisimmät lisäominaisuudet ovat käyttäjien luominen eri oikeuksilla, hälytysrajojen luominen ja valvominen sekä raportin automaattinen generoiminen. Vaikka X-Pack on ilmainen lisäosa, on osa sen tuomista palveluista lisensoituja maksullisen lisenssin alle. (What is Kibana N.d.)

X-Pack voidaan asentaa joko koko ELK-Stackiin, tai ainoastaan osalle ELK-Stackin komponenteista. Tässä tapauksessa koettiin hyödylliseksi asentaa X-Pack kaikkiin komponentteihin. X-Packin asennus oli hyvinkin nopea ja yksinkertainen toimenpide: aluksi lisäosa tuli asentaa jokaiseen komponenttiin erikseen, jonka jälkeen ELK-Stackin komponentteihin piti tehdä pieneiä konfiguraatiolisäyksiä (ks. Kuvio 19). Tämän jälkeen X-Pack oli täysin asennettu ja ELK-Stackiin ei enää kirjautumatta päässyt sisään tai syöttämään tietoja Elasticsearchiin.

```
sudo /usr/share/elasticsearch/bin/elasticsearch-plugin install x-
pack
sudo /usr/share/logstash/bin/logstash-plugin install x-pack
sudo /usr/share/kibana/bin/kibana-plugin install x-pack

###Kibana.yml lisäykset###
xpack.reporting.capture.loadDelay: "10000"
elasticsearch.username: "XXXXXX"
elasticsearch.password: "YYYYYY"

###Logstash.yml lisäykset###
xpack.monitoring.elasticsearch.url: "localhost:9200"
xpack.monitoring.elasticsearch.username: "XXXXXX"
xpack.monitoring.elasticsearch.password: "YYYYYY"

###Logstash conf.d/30-outputs.conf lisäykset###
user => "XXXXXX"
password => "YYYYYY"
```

Kuvio 19. X-Pack asennus ja konfiguraatio

4.6 Testaus

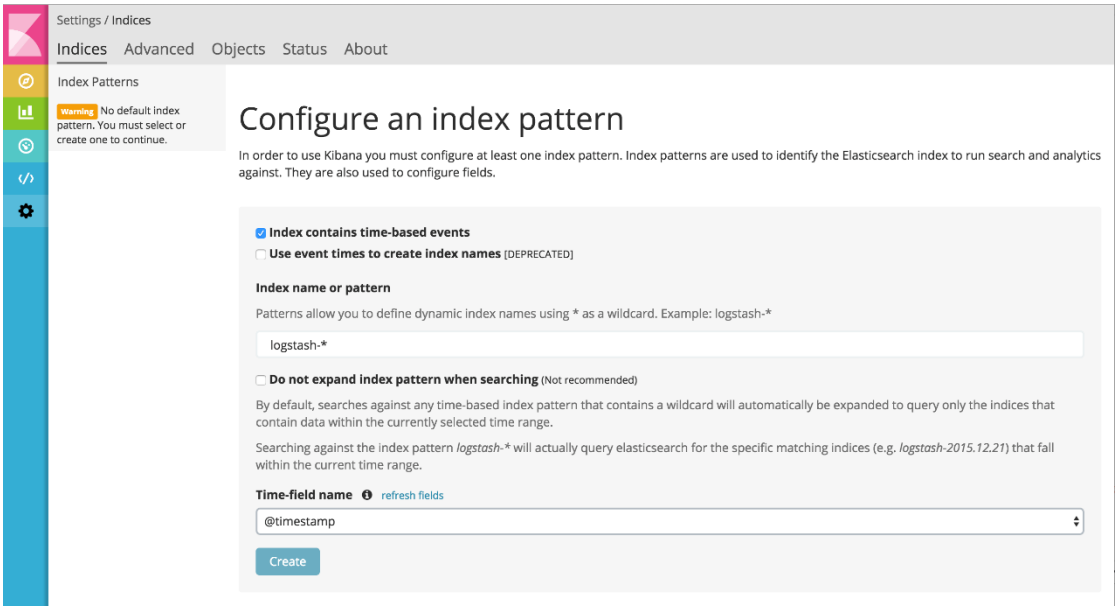
Kun kaikki komponentit saatiin asennettua lisäosineen ja palvelut olivat käynnistetty, voitiin vielä tarkistaa, että kaikki konfiguraatiotiedostoissa määritellyt portit olivat aktiivisena *”LISTEN”* -tilassa. Tämä tapahtui Unixin komennolla *”netstat -tulpn”* (ks. Kuvio 20)

```
root@elkpfsense:~# netstat -tulpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1121/sshd
tcp        0      0 217.112.240.151:5601    0.0.0.0:*               LISTEN      8847/node
tcp6       0      0 :::22                   :::*                    LISTEN      1121/sshd
tcp6       0      0 127.0.0.1:9600          :::*                    LISTEN      8901/java
tcp6       0      0 217.112.240.151:9200    :::*                    LISTEN      9010/java
tcp6       0      0 :::5140                 :::*                    LISTEN      8901/java
tcp6       0      0 217.112.240.151:9300    :::*                    LISTEN      9010/java
udp        0      0 0.0.0.0:5140            0.0.0.0:*                           8901/java
udp        0      0 0.0.0.0:68              0.0.0.0:*                           1016/dhclient
```

Kuvio 20. netstat -tulpn

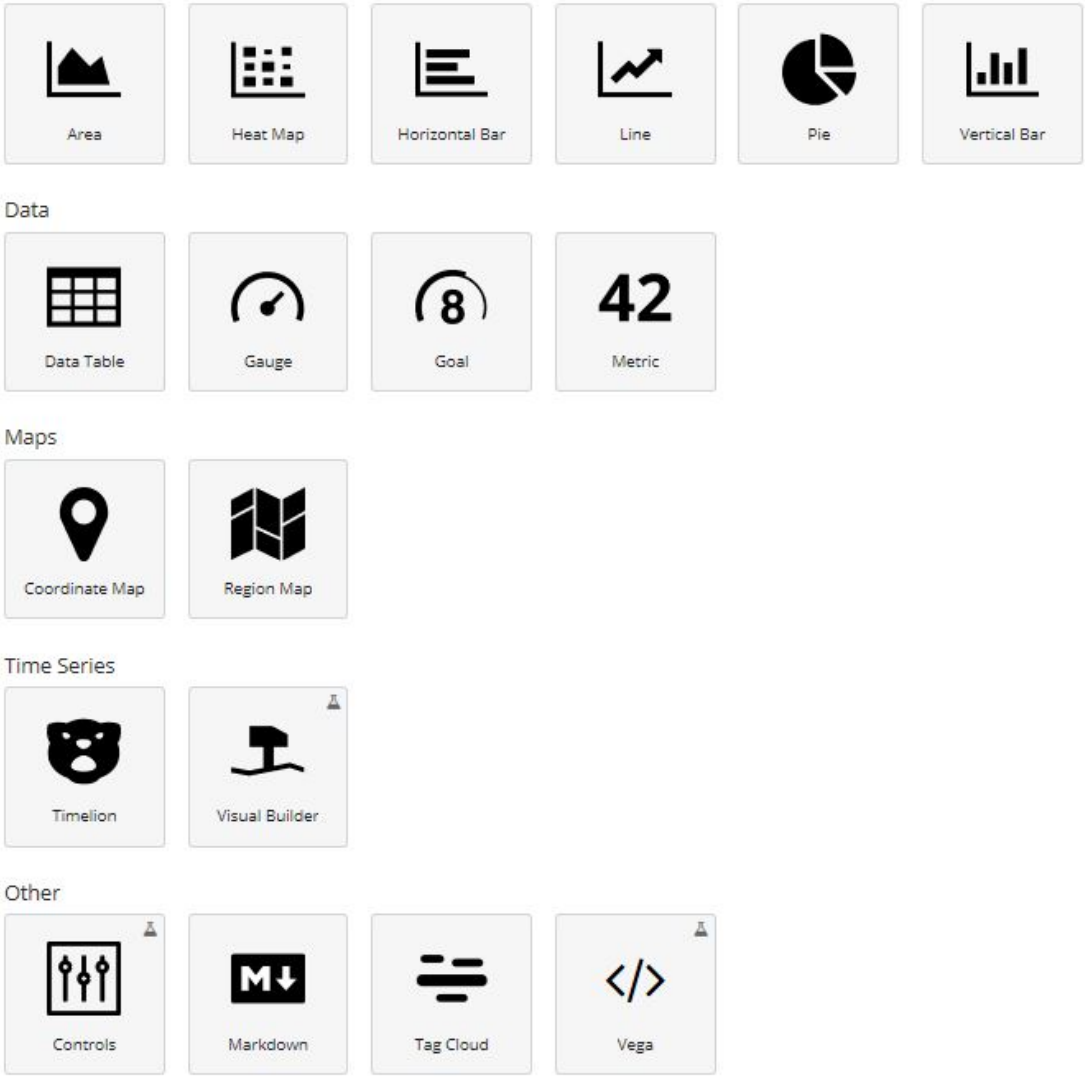
Kun kaikki näytti olevan kunnossa, voitiin kirjautua X-Pack asennuksen yhteydessä luoduilla tunnuksilla Kibanaan. Aluksi piti luoda hakemisto, jonka nimi määriteltiin aiemmin *30-outputs.conf* -tiedostossa. Tiedoston määritelmän mukaan hakemistojen nimet ovat aina logstash-{päivämäärä}. Näin ollen voidaan tehdä yksi hakemisto, joka

pystyy lukemaan kaikkia luotuja hakemistoja kerralla, kun määritellään haettavaksi hakemistoksi *logstash-** (ks Kuvio 21).



Kuvio 21. Logstash hakemiston luominen

Kun hakemisto saatiin lisättyä Kibanaan, ja sinne kirjoitettua dataakin alkoi näkymään Kibanassa, aloitettiin luomaan erilaisia visualisointeja. Visualisointeja tehtiin aluksi pääosin kokeilumielessä, testaten Kibanan erilaisia mahdollisuuksia. Kibanassa on jo vakiona huomattava määrä erilaisia työkaluja (ks. Kuvio 22), mutta myös lisää saisi tehtyä joko itse, tai ladattua Internetistä.



Kuvio 22. Kibanan työkalut

Työkalut todettiin riittäviksi, joten tarvetta alkaa luomaan uusia työkaluja ei ollut. Työkaluista hyödyllisimmiksi osoittautuivat lopulta *pie*, *data table* sekä *markdown*. Visualisointien tekemistä ei tässä työssä käydä läpi tarkemmin, sillä se on varsin triviaalia graafisen käyttöliittymän ja Internetistä löytyvien ohjeiden avulla. Lisäksi lopullisten visualisaatioiden parametrit halutaan pitää TNNet Oy:n sisäisessä tiedossa. Lopullisten visualisaatioiden ulkoasut ovat nähtävillä tämän opinnäytetyön myöhemmissä osioissa valmiiden raporttien muodossa.

Kun ensimmäiset testivisualisaatiot saatiin tehtyä, testattiin myös dashboardin luomista sekä siitä raportin muodostamista X-Packin ”*reporting”* -työkalulla. *Reporting* -työkalu luo muodostetusta dashboardista PDF-tiedoston annetulta aikaväliltä, pyr-

kien optimoimaan kaikki visualisaatiot sopivan kokoisiksi raportille. Nämä toiminnot onnistuivat hyvin suoraviivaisesti ilman ongelmia.

5 Raportti

5.1 Yleistä

Raportin tavoitetila oli olla samanaikaisesti sekä hyödyllinen, myyvä että lainvoimainen. Kuten aiemmin tässä työssä mainittiin, jälkimmäinen ehto estää mm. liian tarkan raportoinnin sisäverkon IP-osoitteiden valvomisesta. Raporttia lukemalla mitään liikennedataa ei tulisi voida yhdistää tietyn isäntälaitteen käyttäjään, ellei kyseessä ole puhdas palvelinverkko, jotta raportista ei tulisi henkilötietorekisteriä.

Jotta raportti olisi myyvä, tulisi statistiikan olla mielellään värikästä ja sisältää suuria lukuja. Värit luovat selkeyttä ja näyttävät nopealla vilkaisulla miellyttävältä, kun taas suuret luvut tuovat tunteen siitä, että palomuuri oikeasti tekee jotakin. Toisaalta pelkät suuret luvut ovat harvoin hyödyllisiä: palomuuri estää oletuksena kaiken liikenteen, jolle ei ole luotu tilallista yhteyttä sisäverkosta, tai liikenteelle ei ole erikseen tehty sallintasääntöä, joten lokitiedon mukainen suuri estettyjen yhteyksien lukumäärä ei todellisuudessa tuo mitään lisäarvoa.

5.2 Raportin ensimmäinen versio

Kun taustajärjestelmät saatiin konfiguroitua ja testattua toimiviksi, lähdettiin kokoamaan ensimmäistä versiota raportista (ks. Liite 1). Raportti levitettiin TNNetissä sisäisesti, pääasiallisena tarkoituksena saada palautetta henkilöiltä, jotka eivät työskentele aktiivisesti verkkotekniikan kanssa. Mikäli raportti todettaisiin hyväksi, voitaisiin se mahdollisesti lähettää sellaisenaan innokkaimmille asiakkaille. Raportissa itsessään ei ollut selitetekstejä millekään graafille, vaan ne olivat saatesanoina sähköpostissa, jolla raportti jaettiin. Tämän sähköpostin pääkohdat sekä raportin selitetekstit nähtävillä liitteessä 6.

Palautteen mukaan raportti näytti potentiaaliselta, ja se oli myös viitesanojen avulla luettava. Tarkkaavaisimmat silmäparit kuitenkin huomasivat pieniä bugeja raportin visualisoinnissa, esimerkiksi ”*Blocked Connections Graph*” näytti hyvin pieniä estomääriä (huippu noin 40), kun yhteenlasketun summan olisi pitänyt olla raportista löytyvä kokonaisestomäärä, 477 544. Syykin tälle löytyi: graafin x-akselin aikavälit olivat sekunnin mittaisia, eikä suinkaan koko raportin kahdentoista tunnin ajanjaksolta tasaisesti.

Myös osa ympyrädiagrammeista oli turhan yksityiskohtaisia ja ne olivat alustavasta palautteesta huolimatta hieman vaikealukuisia. Samalla todettiin, että raportissa oli paljon tarkkaa tietoa estetystä liikenteestä, vaikka tämä data ei ole käytännössä hyödyllistä; estettyä liikennettä ei kuitenkaan voida estää uudestaan.

5.3 Raportin toinen versio

Ensimmäisestä versiosta oppineena lähdettiin työstämään raportin toista versiota (ks. Liite 7). Tavoitteena oli palautteen perusteella tehdä tähän versioon tavoitteena seuraavat parannukset:

- Vähentää hyödytöntä dataa (estetty liikenne).
- Korjata visualisaatiovirheet.
- Selkeyttää kuvia.
- Lisätä selitetekstit suoraan raporttiin.

5.3.1 Hyödyttömän datan poisto

Hyödytöntä dataa lähdettiin purkamaan miettimällä uudelleen mitä dataa raporttiin oikeasti tarvitaan estetystä liikenteestä, vai tarvitaanko ollenkaan? Suurin osa muurille tulevasta liikenteestä on estettyä liikennettä, joten sitä raportoimalla päästään lähemmäksi yhtä raportin tavoitetilaa, myyvyyttä. Näin ollen estettyä liikennettä ei voitu hylätä kokonaan. Ensimmäisestä raportista saadun palautteen perusteella helppolukuista ja mielenkiintoista dataa estetystä liikenteestä olivat liikenteen suunta sekä määrä. Näitä hyödyntäen luotiin raporttiin visualisaatiot ”*Sallittu ja blokattu liikenne*” sekä ”*Sallittu ja blokattu liikenne suunnittain* (ks. Liite 7)”.

Näiden graafien visualisointiin tarvittiin ainoastaan seuraavat tiedot:

- Mikä palomuuri (asiakas) esti liikenteen?

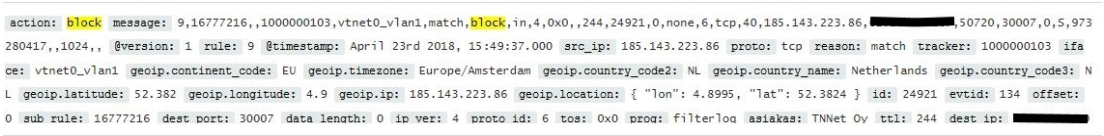
- Oliko liikenne sallittu vai estetty?
- Milloin tapahtuma havaittiin (aikaleima)?
- Mihin suuntaan liikenne oli menossa?

Tämän perusteella pystyttiin myös vähentämään lokituksen kuluttamaa datamäärä, sillä kaikki muu tieto voitiin poistaa estettyjen yhteyksien sisällöstä. Tämä tehtiin lisäämällä Logstashin *29-dropconditions.conf* -tiedostoon kuvion 23 mukainen osio.

```
#jos lokituksen syy on block, otetaan talteen vain pakolliset.
#raportissa ei tarvita tietoa siiä, mitä on blokattu. Ainoastaan
lkm riittää
 if [action] == "block" {
   prune {
    interpolate => true
    whitelist_names =>
["host","action","timestamp","direction","asiakas"]
    }
  }
```

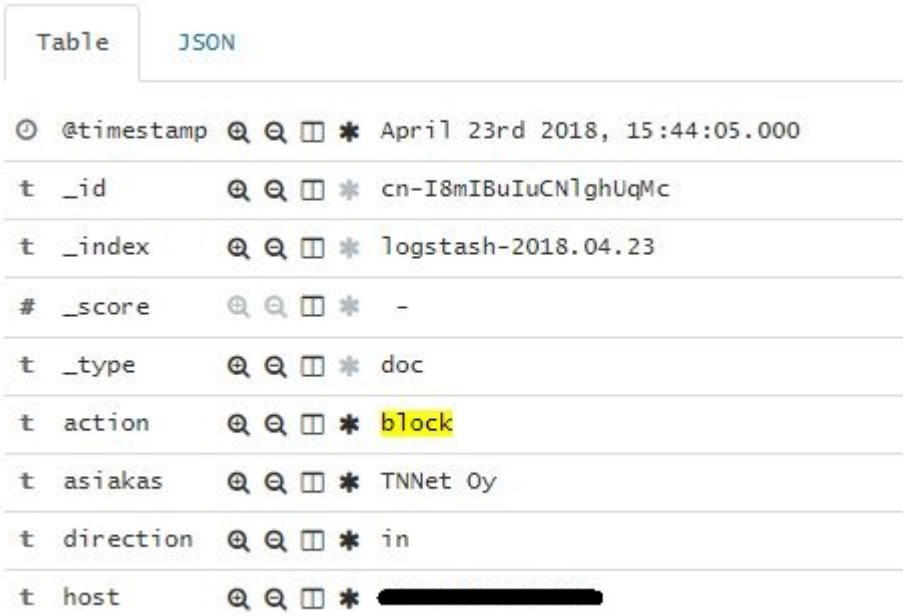
Kuvio 23. 29-dropconditions.conf estetyn tietueen siistiminen

Ennen muutosta estetyssä tietueessa oli yli neljäkymmentä kenttää, jolloin vaarana on, että isoissa hakukyselyissä estetyn tietueen datamäärä hidastaa järjestelmää huomattavasti (ks. Kuvio 24).



Kuvio 24. Ote estetystä tietueesta

Muutoksen jälkeen Logstash tallensi enää kuvion 25 mukaista dataa, jolloin sitä oli huomattavasti tehokkaampaa ja selkeämpää käsitellä.



Kuvio 25. Käsitelty estetty Logstash tietue

5.3.2 Visualisaatiovirheiden korjaus

Virheitä visualisaatiossa alettiin tutkia tarkemmin ja havaittiin, että jostakin syystä jaotellessa liikennettä suunnan mukaan, eli onko liikenne tulossa vai lähtemässä sisäverkosta, ulko- ja sisäverkon IP-osoitteet olivat sekoittuneet. Analysoimalla tämän juurisyytä tarkemmin pfSensen syslogin raakadatasta, havaittiin, että syslogissa oleva suuntaa kuvaava arvo *”direction”* ei tarkoitakaan aina samaa: WAN-rajapinnasta liikennöitäessä tämä arvo on päinvastainen kuin sisäverkon rajapinnoista liikennöitäessä. Ratkaisu ongelmaan oli luoda Logstashiin uusi parametri tekemällä uusi konfiguraatiotiedosto *”20-direction.conf”* (ks. Kuvio 26). Samalla tiedostossa luodut uudet tagit tuli lisätä aiemmin luotuun *”29-dropconditions.conf”*-tiedostoon säilytettäviksi tageiksi, jotta myös estetystä liikenteestä voitaisiin erotella rajapinta sekä liikenteen suunta.

```
#WAN interfacessa syslogin direction IN on sama kuin LAN inter-
faceissa direction OUT
filter {
#Tagataan WAN interface ja sen suunnat
if "vtnet0.1" in [iface] {
    mutate {
    add_tag => ["WAN"]
    }
   if "in" in [direction] {
   mutate {
   add_field => ["suunta", "ulkoa_sisaan"]
    }
  }
  if "out" in [direction] {
   mutate {
   add_field => ["suunta", "sisalta_ulos"]
    }
  }
}
#Tagataan LAN interface ja sen suunnat
if "vtnet0.1" not in [iface] {
    mutate {
    add_tag => ["LAN"]
    }
   if "out" in [direction] {
   mutate {
   add_field => ["suunta", "ulkoa_sisaan"]
    }
  }
  if "in" in [direction] {
   mutate {
   add_field => ["suunta", "sisalta_ulos"]
    }
  }
}
```

Kuvio 26. 20-direction.conf

Tämän muutoksen jälkeen kaikki tunnetut bugit hävisivät, pois lukien estettyjä yhteyksiä laskevan graafin virheellinen aikaleimaus. Tämä korjattiin käyttämällä Kibanassa eri työkalua, Timelionia, jolla myös saatiin graafista kattavampi.

5.3.3 Graafien yksinkertaistaminen

Graafien yksinkertaistaminen alkoi käytännössä huomaamatta jo siinä vaiheessa, kun estetystä datasta poistettiin ylimääräinen data, jolloin sama data hävisi myös graafeista. Lisäksi ympyrädiagrammeista pyrittiin poistamaan ylimääräisiä renkaita pois, jotta ne olisivat helpommin luettavissa. Myös uloimmissa renkaissa esitettävien tietueiden määrää vähennettiin, jotta pienimmät lokerikot eivät olisi liian pieniä luettaviksi. Parhaiten selkeyttämisen onnistumisen huomaa verratessa ensimmäisestä

raportista löytyvää ympyrädiagrammia *”Pass Or Block Direction IN”* toisen raportin *”Sallittu ja blokattu liikenne suunnittain”* ympyrädiagrammiin. Jälkimmäisessä on yksi rengas vähemmän, jolloin myös luettavuus on huomattavasti parempi, mutta hyötydatan määrä on molemmissa kuitenkin sama.

Graafien luettavuutta parannettiin myös lisäämällä graafien alapuolelle datataulukko, joka sisältää saman informaation ”tylsässä” numeerisessa muodossa. Taulukkoa ja visuaalista kuvaa yhdessä tulkitsemalla saatiin raportti ymmärrettävämmäksi ja selkeämmän tuntuiseksi.

5.3.4 Selitetekstien lisäys raporttiin

Selitetekstit kirjoitettiin lähes täysin uudelleen verrattuna ensimmäisen raportin mukana menneisiin viitesanoihin. Tämä johtui osittain siitä, että kaikkiin graafeihin tuli pieniä muutoksia, mutta myös siitä, että ensimmäiset viitesanat käytännössä vain kertoivat, mitä raportilla tapahtui.

Uusiin selitteksteihin pyrittiin avaamaan mikä graafi on kyseessä, mitä se tekee, miksi se on raportilla ja miten siitä saadusta tiedosta voidaan hyötyä. Esimerkiksi toisen raportin selitetekstissä *”Yhteydet ulkoverkosta sisäverkkoon”* selvennettiin kyseisen graafin todellista hyötyä seuraavasti: *”Graafista kannattaa yrittää etsiä kummallisuuksia, kuten RDP (portti 3389) tai SSH (portti 22) yhteyksiä ulkoverkon osoitteista, jotka eivät ole tunnettuja ja luotettavia. Mikäli tuntemattomasta osoitteesta tulee paljon yhteyksiä esimerkiksi porttiin 3389, voidaan olettaa, että kohdeosoitetta yritetään murtaa RDP-yhteyksien avulla ja tämä tulisi erikseen blokata palomuurilla.”*.

5.4 Raportin viimeistely

Raportti lähetetiin aiemmin tehtyjen muutosten jälkeen TNNet Oy:n asiakkaalle, joka on itsekin erikoistunut tietoturvaan. Näin ollen heillä olisi todennäköisimmin mielipiteitä ja kiinnostusta raporttia kohtaan. Asiakkaalle annettiin esimerkkiraportin lisäksi muutama apukysymys, joihin toivottiin vastauksia:

- Mikä on huonoa, mikä hyvää?
- Mitä tietoa haluaisitte lisää?
- Jääkö jokin graafi vaikeasti ymmärrettäväksi?
- Mikä olisi nähdäksenne hyvä raportin julkaisuaikaväli, päivä, viikko, kuukausi?
- Muuta huomioitavaa?

Raportti herätti asiakkaassa sen verran mielenkiintoa, että vastausten sijasta sovittiin yhteinen palaveriaika, jossa raporttia käytäisiin läpi. Samalla asiakas pyysi, että saisivatko he palaveriin raportin omalta muuriltaan. Pyynnöstä johtuen järjestelmän konfiguraation palattiin vielä kerran ennen palaveria.

Havaittiin, että visualisoinnissa olisi käytännöllistä, jos näkyvien porttien perässä olisi myös protokolla, mikäli portti on yleisesti käytetty ja tunnettu. Tämä muutos tehtiin luomalla jälleen uudet konfiguraatiotiedostot Logstashin */etc/conf.d/* -hakemistoon: 30-destports.conf sekä 35-srcports.conf (ks. Kuvio 27).

```
filter {
if [dest_port] == "20" { mutate {replace => [ "dest_port",
"%{dest_port} (FTP)" ]}}
if [dest_port] == "21" { mutate {replace => [ "dest_port",
"%{dest_port} (FTP)" ]}}
if [dest_port] == "22" { mutate {replace => [ "dest_port",
"%{dest_port} (SSH/SCP)" ]}}
if [dest_port] == "23" { mutate {replace => [ "dest_port",
"%{dest_port} (Telnet)" ]}}
if [dest_port] == "25" { mutate {replace => [ "dest_port",
"%{dest_port} (SMTP)" ]}}
```

Kuvio 27. Esimerkki 30-destports.conf -tiedostosta

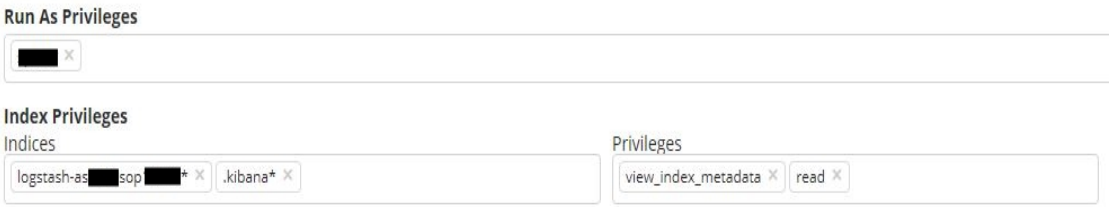
Samalla muutkin hakemiston konfiguraatiotiedostot numeroitiin uudelleen, jotta tiedostot luettaisiin oikeassa järjestyksessä. Lisäksi uusi numerointi mahdollistaisi uusien konfiguraatiotiedostojen luomisen minkä tahansa nykyisen tiedoston väliin. Uusi hakemistopolku oli kuvion 28 mukainen.

```
root@elkpfsense:/etc/kibana# ls /etc/logstash/conf.d/
[              10-syslog.conf   15-snort.conf      25-dropconditions.conf  35-srcports.conf  en
01-inputs.conf  11-pfsense.conf  20-direction.conf  30-destports.conf       40-outputs.conf   patterns
root@elkpfsense:/etc/kibana#
```

Kuvio 28. Logstash conf.d/ lopullinen hakemisto

Raportin uusinta versiota tutkiessa heräsi ajatus siitä, että asiakkaat saattaisivat haluta käyttää myös raportin web-käyttöliittymää, eivätkä tyytyä pelkkään PDF-raporttiin. Jotta tämä olisi mahdollista toteuttaa, tulisi asiakkaiden pääsy rajata siten, että he pääsevät lukemaan jaetulta palvelimelta ainoastaan omaa dataansa. Ratkaisu tähän oli nimetä Logstashin luomat hakemistot uudelleen, ja rajata Kibanan käyttäjille pääsy ainoastaan hakemistoihin, joissa asiakkaan tunniste sijaitsee.

Tämä tehtiin lisäämällä Logstashin *10-syslog.conf* -tiedostossa asiakkaan nimen lisäksi myös asiakkaan asiakasnumerotunniste (assop) tageihin, joka vastaavasti voitiin lisätä *40-outputs.conf* -tiedostossa hakemiston perään*: index => "logstash-%{[assop]}-%{+YYYY.MM.dd}" }*. Näin käyttäjälle voitiin Kibanassa luoda ryhmä, jolla on oikeudet ainoastaan omaan hakemistoonsa sekä itse Kibanaan, jotta ryhmän jäsenet voivat lukea Kibanan tuottamaa sisältöä (ks Kuvio 29). Lisäksi ryhmä tuli lisätä X-Pack lisäosan asetuksissa option *”Dashboard only users”* jäseneksi, jotta ryhmän jäsenet eivät pääse Kibanan sisällä mihinkään muuhun osioon, kuin heille tarkoitettuun Dashboardiin.



Kuvio 29. Kibanan ryhmäoikeudet

6 Lopputulos

6.1 Asiakaspalaveri

Asiakaspalaveriin mentiin asiakkaan oman palomuuriraportin kanssa hakemaan mielipiteitä, kysymyksiä ja parannusehdotuksia raporttia koskien. Hieman yllättäen, mielipiteet olivat erittäin positiivisia. Ainoa selkeä kehitysehdotus oli graafien sieventäminen niissä tapauksissa, kun jokin tapahtuma vie suuren osan (yli 70%) koko graafin pinta-alasta. Ongelmalle mietittiin ratkaisuksi aluksi sitä, että jonkin graafin osuuden ollessa tarpeeksi suuri, se voitaisiin pakottaa esimerkiksi arvoon 30% visuaalisesti.

Tämäkään ei kuitenkaan ollut aivan ideaaliratkaisu, sillä se antaa väärän kuvan tapahtumista ilman tarkempaa vilkaisua tai numeerista dataa.

Palaveri eteni siten, että ensin käytiin läpi asiakkaiden kanssa heille etukäteen lähetetty malliraportti, jonka jälkeen käytiin läpi tietokoneelta heidän oman palomuurinsa raportti reaaliajassa. Pidettyä live-esitystä ja konsultaatiota raportista pidettiin erittäin hyödyllisenä, jopa siinä määrin, että asiakas itse kysyi mahdollisuutta saada vastaavaa palveluna esimerkiksi kuukausittain.

Konsultoidun asiakasraportin sisältö osoittautui hyvinkin hyödylliseksi, sillä asiakkaan sisäverkosta löytyi asioita, joita siellä ei olisi pitänyt olla. Esimerkiksi yllättävin havainto oli mahdollinen TOR-verkon solmukone, jonka syyn asiakas halusi selvittää.

Kuukausittaisen konsultaation mahdollisuutta alettiin tuotteistaa yhdessä asiakkaan kanssa. Asiakkaan kanssa päästiin melko hyvin konsensukseen millainen palvelusta voisi muodostua: Raportin lisäksi asiakkaille voitaisiin tarjota palvelua, jossa TNNet Oy:n asiantuntija pitää kuukausittain audienssin, jossa käydään läpi raportin sisältö ja sen aiheuttamat toimenpiteet.

Lopuksi palattiin vielä aiemmin ilmenneeseen ongelmaan graafissa suuren tilan vievistä tapahtumista. Tähän ratkaisuksi päädyttiin tekemään raportin kustomointia livenä audienssin aikana, tai jopa asiakkaille mahdollisuus itselleen mahdollisuus säätää graafeja. Kaiken kaikkiaan asiakaspalaute oli hyvin myönteistä ja asiakaspalaveri oli molempia osapuolia hyödyttävä.

6.2 Järjestelmävaatimukset

Työtä aloittaessa oli käytännössä mahdotonta arvioida, paljonko lokidataa oikeasti syntyy säilytettäväksi massamuistissa. Myös muut palvelimen resurssit olivat hyvin vaikeasti arvioitavissa, sillä ELK-Stackin valmistajan antamat vaatimukset koskivat pääosin erittäin isoja, big dataa käsitteleviä klustereita.

Lokidatan syntymistä testattiin kolmella hyvin erilaisella muurilla:

- Erittäin ”rauhallinen”, yksityisen henkilön palomuuri.
- Keskitasoinen yritysmuuri alle kolmella julkisella IP-osoitteella.
- Melko vilkas muuri, jossa on paljon sääntöjä ja kymmeniä julkisia IP-osoitteita.

Havaittiin, että vilkas muuri synnyttää vuorokaudessa keskimäärin noin 350MB lokidataa, kun taas rauhallisin muuri vain 25MB samassa ajassa (ks. Kuvio 30).

Name	Status	Document Count	Data
logstash-as2[redacted]-2018.04.26	Yellow	1.3m	351.6 MB
logstash-as2[redacted]-2018.04.27	Yellow	1.1m	263.1 MB
logstash-as62[redacted]-2018.04.26	Yellow	568.3k	151.1 MB
logstash-as62[redacted]-2018.04.27	Yellow	468.2k	148.5 MB
logstash-as61[redacted]-2018.04.27	Yellow	74.9k	24.8 MB
logstash-as61[redacted]-2018.04.26	Yellow	73k	21.1 MB

Kuvio 30. Logstash lokidatan määrä

Tämän tiedon avulla voitiinkin laskea karkeasti, paljonko massamuistia tarvittaisiin palvelimelle: Asetettiin oletusarvo, että yksi asiakas tuottaa maksimissaan 500MB dataa vuorokaudessa ja dataa tulee säilöä vähintään 30 vuorokautta, jotta saadaan koostettua kuukausittainen raportti. Tästä voitiin johtaa laskukaava *500MB * 30d * asiakkaiden_lukumäärä*. Tähän kaavaan sijoitettiin realistinen luku, sata asiakasta, jolloin tarvittava muistimäärä olisi vähintään 1.5TB. Massamuistin ollessa kuitenkin huomattavan halpaa, ei palvelimen muistikapasiteettia kannata lähteä vahingossakaan alimitoittamaan, kun puhutaan verrattain pienistä muistimääristä. Voidaan siis tehdä olettamus, että palvelimelle tulisi allokoida vähintään 10TB muistia, joka riittää varmasti kaikille tuleville asiakkaille 30 vuorokauden datasäilytykseen sekä mahdollistaa myös pidemmän säilytysajan.

Palvelimelle allokoitu 16GB välimuistia tuntuu kuitenkin välillä hieman pieneltä, sillä osa isoista hauista saattaa keskeytyä. Valmistajan suositus 64GB taas tuntuu ylimitoitetulta, joten oikea määrä lienee jossakin näiden arvojen välissä. Todennäköisesti 32GB on hyvä määrä lähteä kokeilemaan, sillä se riittää varmasti nykyiselle asiakasmäärälle sekä sisältää kapasiteettia tuoda palveluun lisää asiakkaita.

Palvelimelle annettiin heti aluksi kaksi prosessoriydintä, jotka tuntuvat tekevän työnsä moitteetta, eikä niistä johtuvaa hidastelua ole ollut havaittavissa järjestelmän käytössä, eikä palvelimen *”top”* -komentoa tarkastellessa. Myös muut palvelimen resursseista tuntuvat oikein riittäviltä, sillä hidastelua ei ole esiintynyt.

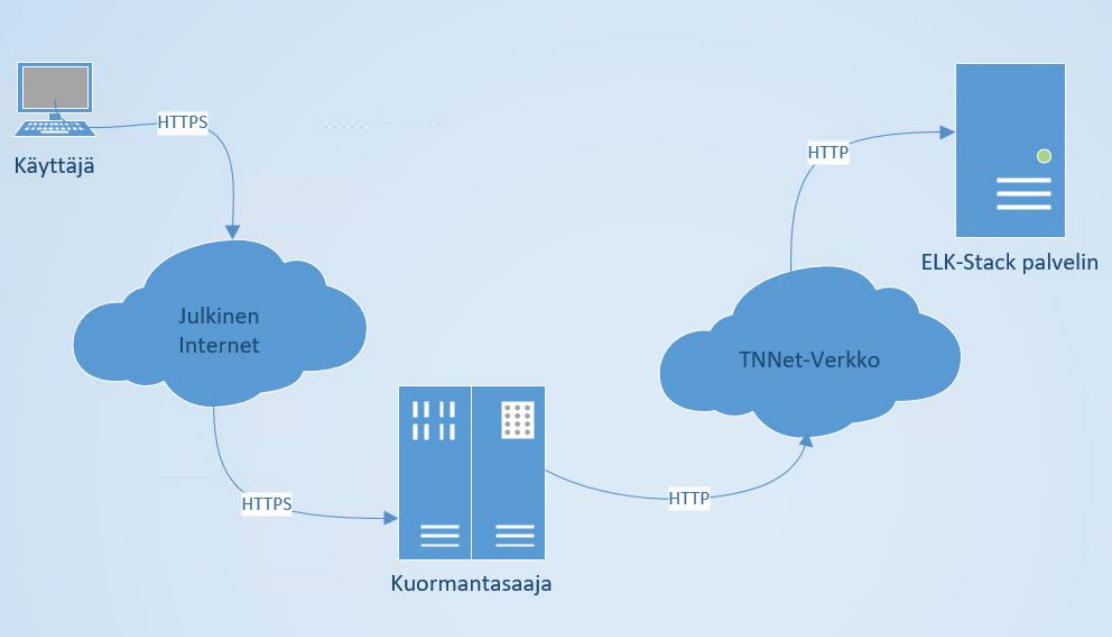
6.3 Tietoturva

PfSensen tapa lähettää syslog data verkon yli ei ole itsessään järin tietoturvallinen, sillä se käyttää ainoastaan UDP-protokollaa, joka ei ole salattua. PfSensessä ei jostakin syystä ole optiota käyttää esimerkiksi TCP protokollaa, joten palomuurijärjestelmän vuoksi ainoa tapa on UDP. Käytännössä siis kuka tahansa voisi lukea selkokielisenä tekstinä palomuurien syslogin raakadataa.

Ongelma on kuitenkin käytännössä vain teoreettinen: Dataa ei missään vaiheessa lähetetä julkisen verkon yli palvelimelta toiselle, vaan liikenne pysyy koko ajan operaattorin tuotantosisäverkossa, joka itsessään on erittäin suojattu. Näin ollen myös palomuurien ja ELK-Stack palvelimen välistä tiedonsiirtoa voidaan pitää tietoturvallisena, UDP-protokollasta huolimatta.

Palvelimelle olisi mahdollista myös asentaa SSL-sertifikaatti, esimerkiksi suojaamaan salasanakirjautumista web-käyttöliittymään pakottamalla kirjautuminen HTTPS:n yli nykyisen HTTP-protokollan sijasta. Palvelimen nykytilassa tätä tietoturvaongelmaa koskee sama sääntö kuin UDP liikennettä: Palvelimelle on rajattu pääsy vain operaattorin tuotantosisäverkosta.

Mikäli palvelimelle tulee luoda pääsy julkisesta verkosta, palvelin tullaan siirtämään kuormantasaajan taakse, jolloin palvelimen ja asiakaskoneen välinen yhteys salataan jo kuormantasaajalla (ks. Kuvio 31). Tässä skenaariossa kuormantasaajan ja ELK-Stack palvelimen välinen yhteys voi olla salaamatonta, sillä se on edelleen operaattorin tuotantosisäverkossa, mutta julkisessa netissä tapahtuva liikenne on salattua. Näin ollen SSL-sertifikaattia ei tarvitse, eikä sitä kannata lähteä asentamaan erikseen ELK-Stack palvelimelle. Kun tuotannossa on useita erilaisia palvelimia, sertifikaattien ylläpito kävisi haastavaksi ja työlääksi yksittäisillä palvelimilla.



Kuvio 31. Kuormantasaajan toiminta

7 Yhteenveto

7.1 Tulokset

Työ tehtiin melko nopeallakin aikataululla siitä sen jälkeen kun tuotteelle huomattiin tarve. Yksi syy tähän oli se, että GDPR:n voimaanastumiseen oli aikaa noin kuukausi, kuin myös se, että raportointijärjestelmän suunnittelu ja rakentaminen osoittautuivat äärimmäisen mielenkiintoiseksi, jopa addiktoivaksi tehtäväksi. Työtä lähdettiin tekemään aluksi ainoastaan keinona estää nykyisten asiakkaiden poistuminen kilpailevien raporttien johdosta. Lopputulemana työstä kehittyi vain pieni, mutta välttämätön osa isompaa palvelukokonaisuutta: generoidun raportin lisäksi voidaan esimerkiksi tarjota palvelua, jossa käydään asiakkaiden kanssa raportin sisältö läpi ja kerrotaan heille mitä toimenpiteitä tulokset aiheuttavat. Lisäksi käytetyillä työkaluilla on mahdollista toteuttaa myös palvelu, jossa valvotaan asiakkaan lokidataa, ja generoidaan hälytys kun jotakin normaalista poikkeavaa ilmenee.

Sekä raportti että sen rinnalle kehittynyt konsultaatiopalvelu toivat heti ensimmäisille testiasiakkaille selkeää lisäarvoa. Heidän sisäverkoistaan löytyi tapahtumia, jotka eivät olleet toivottuja. Samalla asiakkaat ohjeistettiin suorittamaan tarvittavia toi-

menpiteitä näiden havaintojen perusteella. Uuden asiakastuotteen lisäksi järjestelmän avulla löydettiin ongelmakohtia myös yrityksemme omasta toimistoverkosta sekä minun oman kotiliittymäni sisäverkosta.

Henkilökohtaisesti järjestelmän kehittäminen antoi myös itselleni paljon lisää: Paljon opittuja asioita ELK-Stackin komponenteista, lokituksesta sekä isojen datamäärien parsimisesta. Tämän lisäksi myös omaan työhöni tuli uusia, mielenkiintoisia toimenkuvia, kuten asiakkaiden liittäminen raporttijärjestelmään, raporttien kustomoiminen asiakkaiden toiveiden mukaiseksi, sekä asiakkaiden konsultointitehtävät raporttien pohjalta.

Opinnäytetyössä syntynyttä järjestelmää voidaan siis pitää erittäin onnistuneena: Alkuperäisen tavoitteen, raportin luomisen lisäksi TNNet Oy:lle kehittyi mahdollisuus tarjota uusia tuotteita ja minulle itselleni syntyi mahdollisuus uusiin toimenkuviin organisaatiossa. Myös raportille työn alussa asetetut tavoitteet täyttyivät: Raportti sisältää osioita täysin myyntiteknisistä syistä, esimerkiksi kauniilla väreillä visualisoidut kymmenen yleisintä maata, joista on liikennöity kohti sisäverkkoa, on selkeä myyntitekninen osio. Toisaalta raportti sisältää myös todellista hyötydataa teknisille ihmisille kuten asiakaspalaverissakin havaittiin.

Raportti lisäpalveluineen istuu myös TNNet Oy:n ajatusmalliin automatisoida verkkoa siten, että asiantuntijat voivat irtaantua muihin tehtäviin tuottamaan yritykselle ja asiakkaille lisäarvoa. Raportin tulkitseminen asiakkaille on juuri sitä lisäarvoa, jota halutaan tuottaa ja tällä hetkellä myös pystytään tuottamaan nykyisillä resursseilla.

7.2 Tulevaisuuden kehityskohteet

Raportointijärjestelmä ei suinkaan ole sellaisenaan vielä täydellinen, vaan siihen tulee varmasti lisää kehitysideoita sitä mukaa, kun uusia asiakkaita liittyy järjestelmän piiriin. Raportti on suunnattu asiakkaille ja kullakin asiakkaalla voi olla erilaisia tarpeita ja toiveita raportin sisällöstä. Myös yleinen asiakaspalaute auttaa todennäköisesti kehittämään niin raportin visuaalista ulkoasua, kuin teknistä sisältöä.

Tulevaisuudessa on myös tavoitteena saada asiakkaille web-käyttöliittymä raporttiin, sillä raporttia on huomattavasti parempi lukea interaktiivisesti hiiren avulla, kuin staattiselta PDF-raportilta. Tälle on toki jo ELK-Stackin puolesta tässä opinnäytetyössä

konfiguroitu ja testattu valmiudet luoda asiakaskohtaisia käyttäjiä ja rajata asiakkaiden pääsy ainoastaan heidän lokidataansa. Käyttöliittymäsuunnittelu tulee kuitenkin ottaa vielä huomioon web-käyttöliittymää tehdessä.

Vaikka web-käyttöliittymä kehitettäisiinkin valmiiksi, aiemmin tässä työssä mainittua mahdollisuutta antaa asiakkaille oikeudet muokata graafeja pidän kuitenkin itse huonona. Kibanassa tehdyt graafit ovat raportin oleellisin osio myyntiteknisesti, enkä pidä järkevänä antaa asiakkaille pääsyä näkemään niitä. Lisäksi livekustomoinnin suorittaminen ainoastaan konsultaation yhteydessä on oiva myyntivaltti konsulttipalvelua myydessä.

Teknisesti järjestelmää tulee ylläpitää, kuten mitä tahansa muutakin yrityksen järjestelmää. Palvelinta ei voi jättää päivittämättä ja varmuuskopioimatta, vaan nämä tulee hoitaa myös jatkossa. Lisäksi palvelimen kapasiteettia tulee seurata ja mahdollisesti lisätä resursseja sitä mukaa, kun asiakkaita tulee lisää ja lokidatan käsittely käy raskaammaksi. Tämänhetkiset resurssit riittävät kohtalaisesti tämän hetken asiakasmäärälle, mutta ideaalitapauksessa nykyiset resurssit jäävät hyvinkin nopeasti pieniksi asiakasmäärän kasvun johdosta. Mahdollista on myös se, että järjestelmää tulee skaalata usealle eri palvelimelle, eli klusteroida. Lähitulevaisuudessa tullaan kuitenkin ainakin lisäämään RAM-muistin määrä 32GB:n sekä kasvattamaan massamuistin määrää vähintään useisiin teratavuihin. Vaikka palvelin on virtualisoitu ja täten muistimäärien kasvattaminen käytännössä on hyvin triviaali toimenpide, ei palvelimelle tule yliallokoida resursseja. Massamuistin pienentäminen saattaa tuottaa ongelmia, jos dataa on kirjoitettu hajautetusti levyjärjestelmälle. Massamuistin kasvattaminen onkin huomattavasti helpompaa ja todennäköisesti tulevaisuudessa myös tarpeellista.

Järjestelmän ylläpito ja käyttö tulee kouluttaa organisaatiossa myös muille työntekijöille, jotta koko järjestelmä ei ole yhden henkilön varassa. Tämä opinnäytetyö on toki hyvä pohja henkilöille ilman aiempaa kokemusta, mutta opinnäytetyössäkään ei ole voitu kertoa aivan kaikkea. Lisäksi henkilökohtainen opastus on varmasti tehokkaampaa, kuin kirjallisten ohjeiden lukeminen. Ylläpidollisen koulutuksen lisäksi koulutusta tulee antaa myös organisaation myyjille, jotta he osaavat myydä tuotetta mahdollisimman tehokkaasti.

Itse raportin tuottaminen tulee vielä sekä tuotteistaa, että hinnoitella kunnolla. Tämä sisältää muun muassa tuotteen hinnoittelun sekä palvelukuvauksen muodostamisen. Sama toimenpide tulee tehdä myös raportin tukipalveluille, joita todennäköisesti ovat ainakin konsultaatio ja automaattiset hälytykset. Todennäköisesti raportin ympärille on mahdollista rakentaa muitakin lisäpalveluita, mutta nämä eivät ole vielä tulleet esille, vaan ne ilmenevät tulevaisuudessa tämän ja muiden tuotteiden kehittyessä.

Lähteet

Amon, C, Amon, R & Shimonski, J. 2003. The Best Damn Firewall Book Period. Viitattu 29.4.2018 https://docstore.mik.ua/cisco/pdf/security/Syngress%20-%20Best%20Damn%20Firewall%20Book%20Period.pdf

Cholakian, A. 2013. Exploring Elasticsearch. Viitattu 28.4.2018 http://exploringelasticsearch.com

Data Visualization. SAS verkkosivut. Viitattu 23.4.2018 https://www.sas.com/en_us/insights/big-data/data-visualization.html

Eaton, I. 2003. The Ins and Outs of System Logging Using Syslog. Viitattu 29.4.2018 https://www.sans.org/reading-room/whitepapers/logging/ins-outs-system-logging-syslog-1168

Elasticsearch Reference. N.d. Elastic Co, verkkosivut. Viitattu 28.4.2018 https://www.elastic.co/guide/en/elasticsearch/reference/

Elk-Stack. N.d. Elastic Co, verkkosivut. Viitattu 23.4.2018 https://www.elastic.co/elk-stack

GDPR - EU:n uusi tietosuoja-asetus. N.d. Nettiartikkeli. Viitattu 6.5.2018 https://fakta.tietosuojamalli.fi/aihe/gdpr

Gerhards, R. 2009. The Syslog Protocol. Request for Comments: 5424. Network Working Group. Viitattu 29.4.2018. http://tools.ietf.org/html/rfc542

Kibana Reference. N.d. Elastic Co, verkkosivut. Viitattu 27.4.2018. https://www.elastic.co/guide/en/kibana/6.x/introduction.html

Logstash Reference. N.d. Elastic Co, verkkosivut. Viitattu 27.4.2018. https://www.elastic.co/guide/en/logstash/current/first-event.html

Lokiohje. 2009. Julkisen hallinnon digitaalisen turvallisuuden johtoryhmän (VAHTI) ohjesivusto. Viitattu 21.4.2018**.** https://www.vahtiohje.fi/c/document_library/get_file?uuid=775179cb-6c54-4dfb-b65d-e925d47c61d2&groupId=10229

Lokitusohje. 2016. Viestintäviraston verkkosivut. Viitattu 21.4.2018. https://www.viestintavirasto.fi/attachments/tietoturva/Lokitusohje.pdf

Mitä jokaisen kuuluu tietää GDPR:stä. N.d. Nettiartikkeli. Viitattu 6.5.2018 https://findwise.com/en/gdpr-fi

Pfelk. 2017. Käyttäjän a3ilson github-repository. Viitattu 23.4.2018 https://github.com/a3ilson/pfelk/

Shiner, D. 2005. Understanding how an intrusion detection system (IDS) works. Viitattu 28.4.2018 https://www.techrepublic.com/article/solutionbase-understanding-how-an-intrusion-detection-system-ids-works/

Snort Official Documentation. 2017. Snort-ohje. Viitattu 23.4.2018 https://www.snort.org/documents/snort-users-manual

What is a firewall. N.d. Cisco verkkolaitevalmistajan verkkosivut. Viitattu 23.4.2018
https://www.cisco.com/c/en/us/products/security/firewalls/what-is-a-firewall.html

What is Kibana. N.d. Amazon AWS verkkosivut. Viitattu 27.4.2018
https://aws.amazon.com/elasticsearch-service/kibana/

Liitteet

Liite 1. 10-syslog.conf

```
filter {
  if [type] == "syslog" {
      mutate {
        add_tag => ["PFSense", "Ready"]
      }
    }
    if "Ready" not in [tags] {
      mutate {
        add_tag => [ "syslog" ]
      }
    }
}
filter {
  if [type] == "syslog" {
    mutate {
      remove_tag => "Ready"
    }
  }
}
#Tagataan asiakkuus palomuurin IP-osoitteen perusteella
filter {
  if [host] == "###.###.###.###" {
    mutate {
    add_field => [ "asiakas", "##########"]
    }
  }
  if [host] == "###.###.###.###" {
    mutate {
    add_field => [ "asiakas", "#########"]
    }
  }
  if [host] == "###.###.###.###" {
    mutate {
    add_field => [ "asiakas", "#########"]
    }
  }
}
filter {
  if "syslog" in [tags] {
    grok {
      match => { "message" => "%{SYSLOGTIMESTAMP:syslog_timestamp}
%{SYSLOGHOST:syslog_hostname}
%{DATA:syslog_program}(?:\[%{POSINT:syslog_pid}\])?:
%{GREEDYDATA:syslog_message}" }
      add_field => [ "received_at", "%{@timestamp}" ]
      add_field => [ "received_from", "%{host}" ]
    }
    syslog_pri { }
    date {
      match => [ "syslog_timestamp", "MMM  d HH:mm:ss", "MMM  dd
HH:mm:ss" ]
      locale => "en"
    }
```

```
    if !("_grokparsefailure" in [tags]) {

      mutate {
        replace => [ "@source_host", "%{syslog_hostname}" ]
        replace => [ "@message", "%{syslog_message}" ]
      }
    }
    mutate {
      remove_field => [ "syslog_hostname", "syslog_message",
"syslog_timestamp" ]
    }
  }
}
```

Liite 2. 11-pfsense.conf

```
filter {

 if "PFSense" in [tags] {
    grok {
      add_tag => [ "firewall" ]
      match => [ "message", "<(?<ev-
tid>.*)>(?<datetime>(?:Jan(?:uary)?|Feb(?:ruary)?|Mar(?:ch)?|A
pr(?:il)?|May|Jun(?:e)?|Jul((?:y)?|Aug(?:ust)?|Sep(?:tember)?|
Oct(?:ober)?|Nov(?:ember)?|Dec(?:ember)?)\s+(?:(?:0[1-
9])|(?:[12][0-9])|(?:3[01])|[1--9]) (?:2[0123]|[01]?[0-
9]):(?:[0-5][0-9]):(?:[0-5][0-9])) (?<prog>.*?): (?<msg>.*)" ]
    }
    mutate {
      gsub => ["datetime","  "," "]
    }
    date {
      match => [ "datetime", "MMM dd HH:mm:ss" ]
      timezone => "Europe/Helsinki"
    }
    mutate {
      replace => [ "message", "%{msg}" ]
    }
    mutate {
      remove_field => [ "msg", "datetime" ]
    }
  }
if [prog] =~ /^filterlog$/ {
    mutate {
      remove_field => [ "msg", "datetime" ]
    }
    grok {
      patterns_dir => "/etc/logstash/conf.d/patterns"
      match => [ "message",
"%{PFSENSE_LOG_DATA}%{PFSENSE_IP_SPECIFIC_DATA}%{PFSENSE_IP_DA
TA}%{PFSENSE_PROTOCOL_DATA}",
                 "message",
"%{PFSENSE_LOG_DATA}%{PFSENSE_IPv4_SPECIFIC_DATA_ECN}%{PFSENSE
_IP_DATA}%{PFSENSE_PROTOCOL_DATA}",
                 "message",
"%{PFSENSE_LOG_DATA}%{PFSENSE_IPv6_SPECIFIC_DATA}"]
    }
    mutate {
      lowercase => [ 'proto' ]
    }
if "_geoip_lookup_failure" not in [tags] {
    geoip {
      add_tag => [ "GeoIP" ]
      source => "src_ip"
      database => "/etc/logstash/GeoLite2-City.mmdb"
      }
    }
  }
}
```

Liite 3. pfsense2-4.grok

```
# GROK match pattern for logstash.conf filter:

# pfsense 2.4 modification: sub_rule is optional.
PFSENSE_LOG_DATA
(%{INT:rule}),(%{INT:sub_rule})?,,(%{INT:tracker}),(%{DATA:iface})
,(%{WORD:reason}),(%{WORD:action}),(%{WORD:direction}),(%{INT:ip_v
er}),
PFSENSE_IP_SPECIFIC_DATA
(%{PFSENSE_IPv4_SPECIFIC_DATA}|%{PFSENSE_IPv6_SPECIFIC_DATA})
PFSENSE_IPv4_SPECIFIC_DATA
(%{BASE16NUM:tos}),,(%{INT:ttl}),(%{INT:id}),(%{INT:offset}),(%{WO
RD:flags}),(%{INT:proto_id}),(%{WORD:proto}),
PFSENSE_IPv4_SPECIFIC_DATA_ECN
(%{BASE16NUM:tos}),(%{INT:ecn}),(%{INT:ttl}),(%{INT:id}),(%{INT:of
fset}),(%{WORD:flags}),(%{INT:proto_id}),(%{WORD:proto}),
PFSENSE_IPv6_SPECIFIC_DATA
(%{BASE16NUM:class}),(%{DATA:flow_label}),(%{INT:hop_limit}),(%{WO
RD:proto}),(%{INT:proto_id}),
PFSENSE_IP_DATA (%{INT:length}),(%{IP:src_ip}),(%{IP:dest_ip}),
PFSENSE_PROTOCOL_DATA
(%{PFSENSE_TCP_DATA}|%{PFSENSE_UDP_DATA}|%{PFSENSE_ICMP_DATA}|%{PF
SENSE_CARP_DATA})
PFSENSE_TCP_DATA
(%{INT:src_port}),(%{INT:dest_port}),(%{INT:data_length}),(%{WORD:
tcp_flags}),(%{INT:sequence_number}),(%{INT:ack_number}),(%{INT:tc
p_window}),(%{DATA:urg_data}),(%{DATA:tcp_options})
PFSENSE_UDP_DATA
(%{INT:src_port}),(%{INT:dest_port}),(%{INT:data_length})
PFSENSE_ICMP_DATA (%{PFSENSE_ICMP_TYPE}%{PFSENSE_ICMP_RESPONSE})
PFSENSE_ICMP_TYPE
(?<icmp_type>(request|reply|unreachproto|unreachport|unreach|timee
xceed|paramprob|redirect|maskreply|needfrag|tstamp|tstampreply)),
PFSENSE_ICMP_RESPONSE
(%{PFSENSE_ICMP_ECHO_REQ_REPLY}|%{PFSENSE_ICMP_UNREACHPORT}|
%{PFSENSE_ICMP_UNREACHPROTO}|%{PFSENSE_ICMP_UNREACHABLE}|%{PFSENSE
_ICMP_NEED_FLAG}|%{PFSENSE_ICMP_TSTAMP}|%{PFSENSE_ICMP_TSTAMP_REPL
Y})
PFSENSE_ICMP_ECHO_REQ_REPLY
(%{INT:icmp_echo_id}),(%{INT:icmp_echo_sequence})
PFSENSE_ICMP_UNREACHPORT
(%{IP:icmp_unreachport_dest_ip}),(%{WORD:icmp_unreachport_protocol
}),(%{INT:icmp_unreachport_port})
PFSENSE_ICMP_UNREACHPROTO
(%{IP:icmp_unreach_dest_ip}),(%{WORD:icmp_unreachproto_protocol})
PFSENSE_ICMP_UNREACHABLE (%{GREEDYDATA:icmp_unreachable})
PFSENSE_ICMP_NEED_FLAG
(%{IP:icmp_need_flag_ip}),(%{INT:icmp_need_flag_mtu})
PFSENSE_ICMP_TSTAMP
(%{INT:icmp_tstamp_id}),(%{INT:icmp_tstamp_sequence})
PFSENSE_ICMP_TSTAMP_REPLY
(%{INT:icmp_tstamp_reply_id}),(%{INT:icmp_tstamp_reply_sequence}),
(%{INT:icmp_tstamp_reply_otime}),(%{INT:icmp_tstamp_reply_rtime}),
(%{INT:icmp_tstamp_reply_ttime})

PFSENSE_CARP_DATA
(%{WORD:carp_type}),(%{INT:carp_ttl}),(%{INT:carp_vhid}),(%{INT:ca
rp_version}),(%{INT:carp_advbase}),(%{INT:carp_advskew})
```

Liite 4. 15-snort.conf

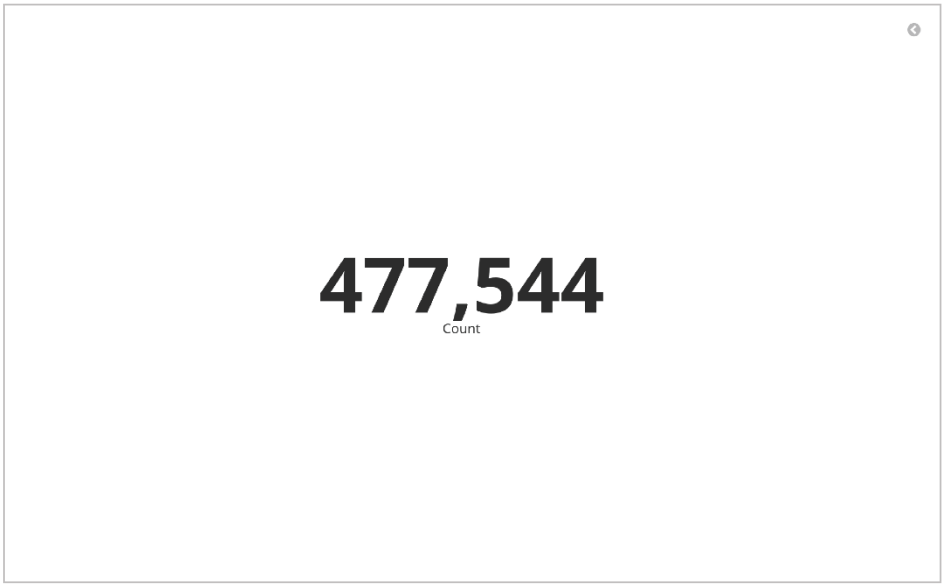
```
filter {
  if "snort" in [prog] {
    mutate {
     add_tag => ["snort"]
     remove_tag => ["PFSense"]
    }
   grok {
    match => ["message",
".%{INT:ids_gid}\:%{INT:ids_sid}\:%{INT:ids_rev}\]%{DATA:ids_re
ason}\[Classification:\s%{DATA:ids_cl$assification}\]\s.Priorit
y:\s%{INT:priority}\]\s.%{WORD:ids_proto}.\s%{IP:src_ip}(.%{INT
:src_port})?\s\-\>\s%{IP:dst_ip}(.%{INT:dst_port})?"]
    }
   if [priority] == "1" {
      mutate {
        add_field => { "severity" => "High" }
      }
    }
    if [priority] == "2" {
      mutate {
        add_field => { "severity" => "Medium" }
      }
    }
    if [priority] == "3" {
      mutate {
        add_field => { "severity" => "Low" }
      }
    }
    if [ids_proto] {
      if [ids_proto] =~ /^GPL/ {
        mutate {
          add_tag => [ "Snort-ET-sig" ]
          add_field => [ "ids_rule_type", "Emerging Threats" ]
        }
      }
      if [ids_proto] =~ /^ET/ {
        mutate {
          add_tag => [ "Snort-ET-sig" ]
          add_field => [ "ids_rule_type", "Emerging Threats" ]
        }
      }
      if "Snort-ET-sig" not in [tags] {
        mutate {
          add_tag => [ "Snort-sig" ]
          add_field => [ "ids_rule_type", "Snort" ]
        }
      }
    }
if "Snort-sig" in [tags] {
      if [ids_gid] == "1" {
        mutate {
          add_field => [ "Signature_Info",
"http://rootedyour/.com/snortsid?sid=%{ids_sid}" ]
        }
      }
```

```
if [ids_gid] != "1" {
        mutate {
          add_field => [ "Signature_Info",
"http://rootedyour.com/snortsid?sid=%{ids_gid}-%{ids_sid}" ]
        }
      }
    }
if "Snort-ET-sig" in [tags] {
      mutate {
        add_field => [ "Signature_Info",
"http://doc.emergingthreats.net/bin/view/Main/%{ids_sid}" ]
      }
    }
  if "_geoip_lookup_failure" not in [tags] {
    geoip {
      add_tag => [ "GeoIP" ]
      source => "src_ip"
      database => "/etc/logstash/GeoLite2-City.mmdb"
      }
    }
  }
}
```

Liite 5. TNNet IDS-Raportti versio 1

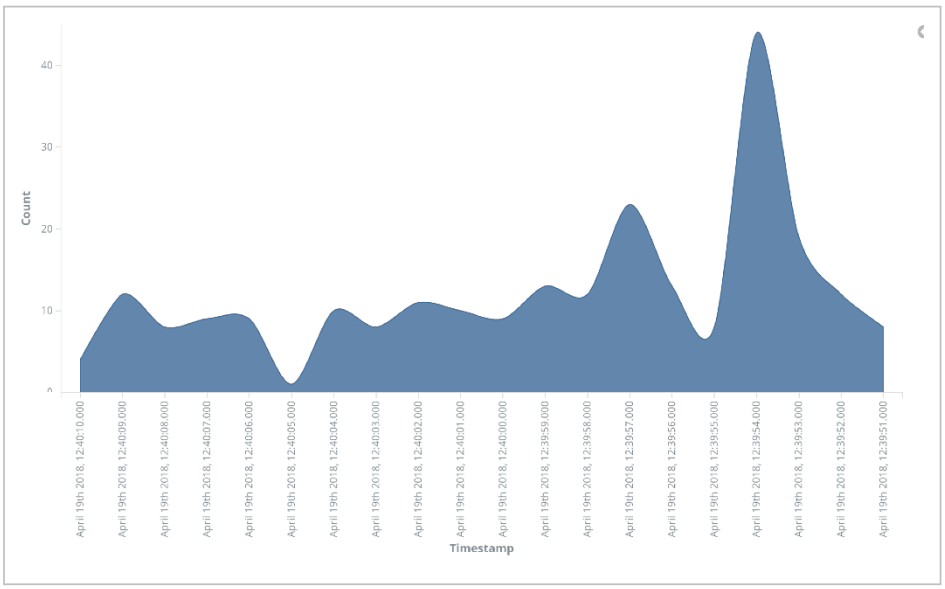
IDS Raportti — Thu, Apr 19, 2018 12:40 AM to Thu, Apr 19, 2018 12:40 PM

Blocked Connections

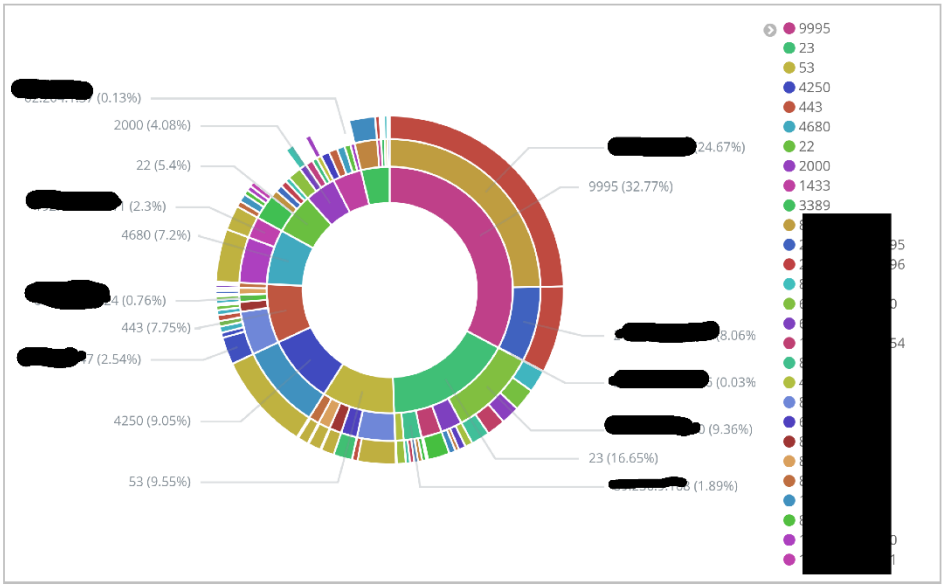


Country Traffic direction in

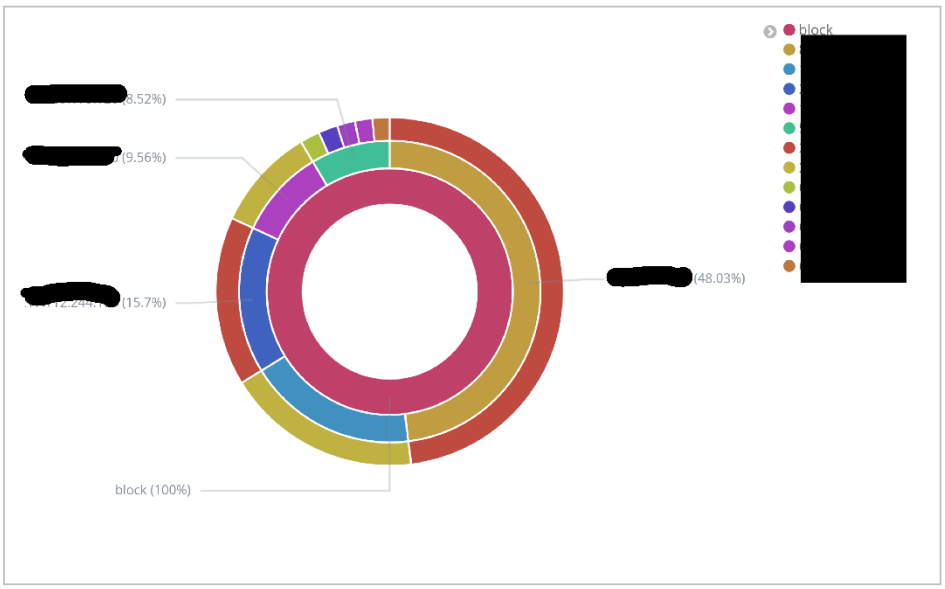
Blocked Connections Graph



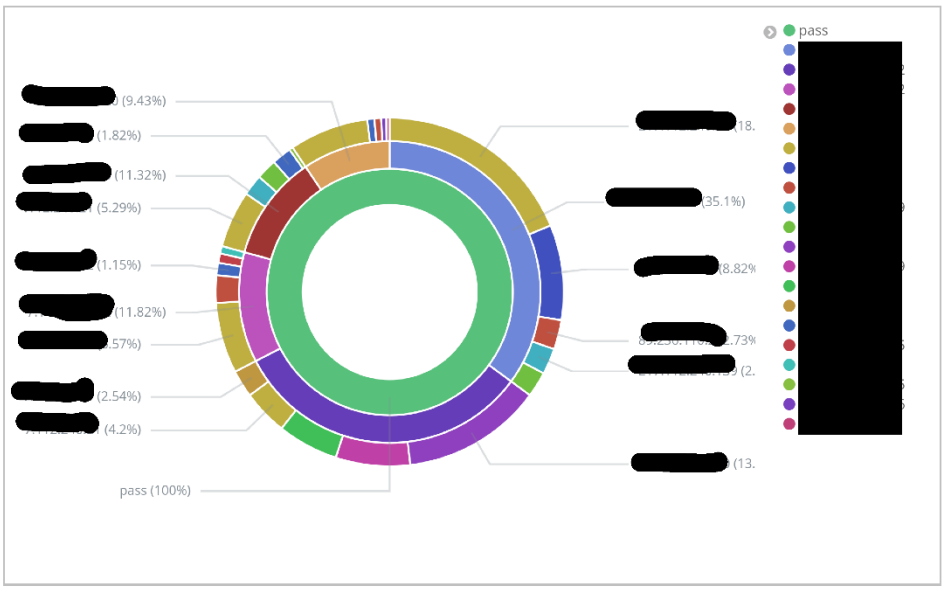
Destination Ports from Source IP direction IN



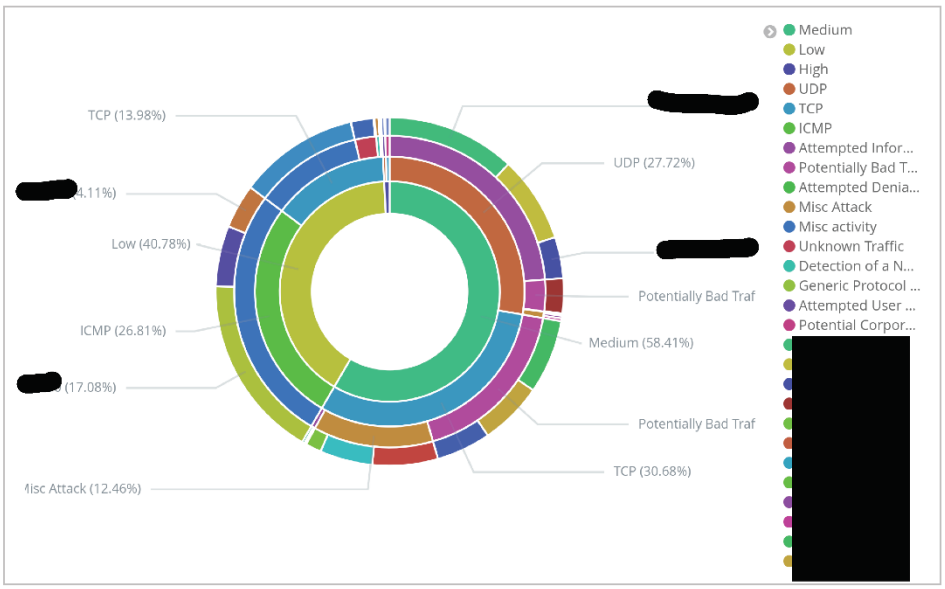
Pass or Block direction IN



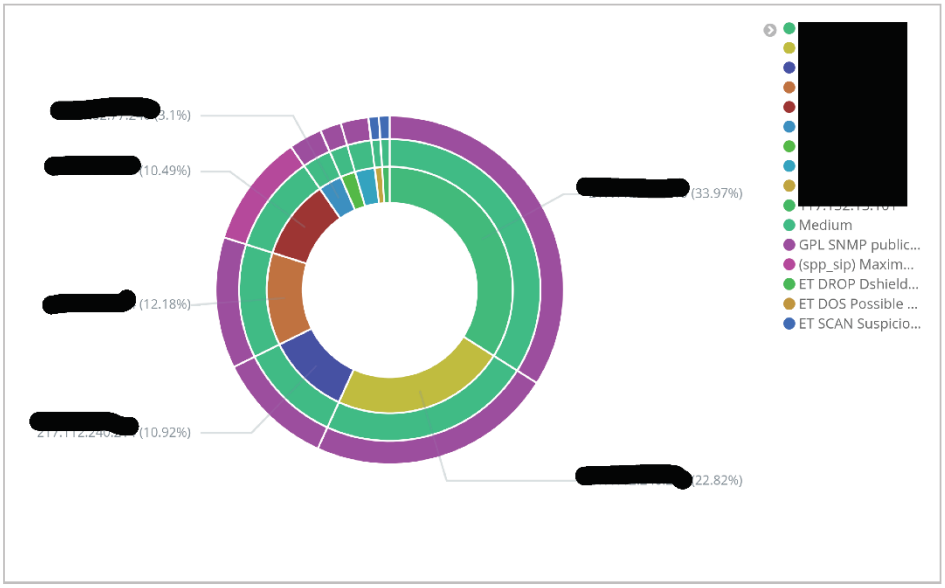
Pass or Block direction OUT



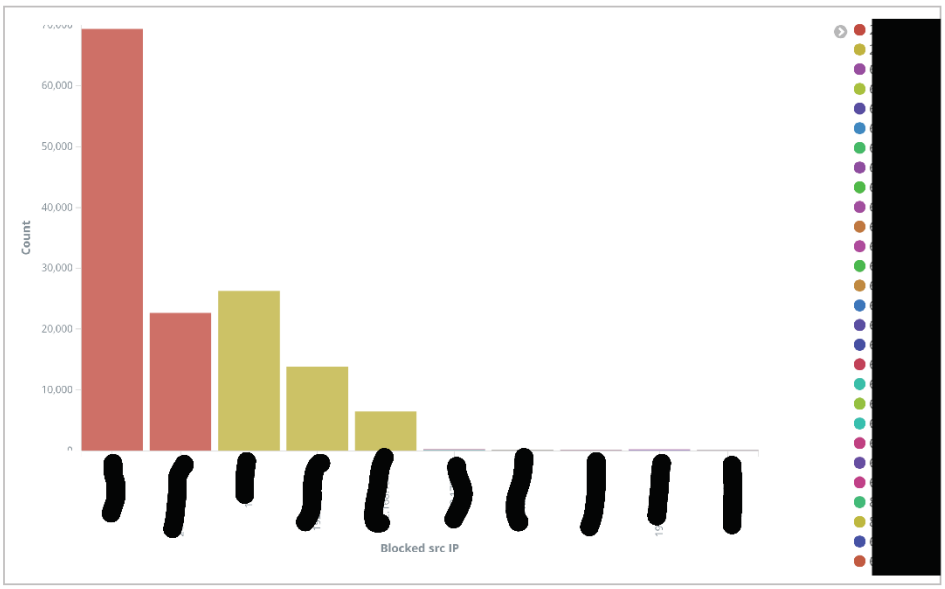
Snort Severity



Severity Med or High snort alert



Blocked IPs to IP direction IN



Liite 6. Ensimmäisen raportin jakelun pääkohdat

Palomuurimme estää oletuksena kaiken liikenteen, jota ei erikseen olla sallittuna. Näin ollen ulkoapäin tulevaa liikennettä on hieman tympeää mitata, kun käytännössä kaikki liikenne estyy. Tästä huolimatta on kuitenkin mielenkiintoista tietää, estääkö muuri oikeasti jotakin ja jos kyllä, niin mitä. Tästä syystä raportista löytyykin blokattujen yhteyksien määrä, maat joista estot ovat tulleet sekä hieman tarkempaa dataa lähdeosoitteista ja -porteista.

Muurin sisäverkon puoli onkin mielenkiintoisempaa tutkittavaa, ja sinne meillä onkin IDS-valvonta, joka haistelee mahdollisia tietoturvauhkia, kuten liikennettä tunnettuihin bottiverkkoihin tai porttiskannauksia. Näistä havainnoista otetaan ylös myös lähdeosoite, jotta mahdollisesti saastunut kone on mahdollista paikantaa. Muutoin emme sisäverkon koneiden liikennettä valvo, sillä se rikkoo käyttäjien yksityisyyttä ja on jopa itseasiassa Suomen lain vastaista.

Blocked Connections - Yksinkertaisesti lokitetaan jokainen muurin estämä yhteysyritys. Raportti on 12 tunnin ajalta, ja kuten huomata saattaa, siinäkin ajassa muurimme blokkaa ~500 000 yhteysyritystä.

Country Traffic direction in – Kymmenen yleisintä maata, joista yritetään liikennöidä tai joihin liikennöidään. Ei koske siis sisäverkosta ulkoverkkoon menevää liikennettä

Blocked Connections Graph - Aikajana siitä, millä aikavälillä on estettyjä yhteyksiä ja kuinka paljon.

Destination Ports from Source IP direction IN - Lähinnä mielenkiintoista dataa siitä, mihin portteihin, mihin osoitteeseen ja mistä ulkoverkon osoitteesta liikennöidään tai yritetään liikennöidä sisäverkkoa kohti.

Pass or Block direction IN - Ulkoverkosta sisäänpäin tulevaa liikennettä, onko päästetty muurin läpi vai ei. Jälleen näkyy lähde- ja kohdeosoitteet. Muurin tiukan määrittelyn vuoksi tässä kuvassa muuri ei ole päästänyt yhtään ulkoverkon yhteysavauksia läpi, joten block on 100%

Pass or Block direction OUT - Sama kuin äskeinen, mutta toiseen suuntaan. Tällä kertaa tosin kaikki on ollut sallittua liikennettä

Snort Severity - Sisäverkon haitallisen liikenteen haistelua. Severity, eli havainnon vakavuus kolmiportaisesti Low - Medium, High. Low on hyvin usein merkityksetöntä ja saatetaan jättää lopullisesta raportista pois. Lisäksi kuvasta näkyy Protokolla, jota on käytetty ja tarkempi selvitys mikä haitallisen liikenteen syy on ollut (esimerkiksi Attempted information leak, joka sisältää esimerkiksi SNMP haistelun ja SQL-injektio yritykset)

Severity Med or High snort alert - Jos Vakavuus on Medium tai High, otetaan se tarkempaan tarkastelluun kyseisessä graafissa, jotta mahdolliset saastuneet koneet olisi helpompi havainnoida ja löytää.

Blocked IPs to IP direction IN - Käytännössä sama, kuin "Pass or Block direction IN", mutta eri muodossa. Tässä näkyy lähde- ja kohdeosoitteet sisäänpäin tulevalle liikenteelle, joka on jäänyt blokkiin.

Liite 7. TNNet IDS-Raportti versio 2

TNNet Oy Palomuuriraportti

Raportista löytyy statistiikkaa palomuurille asti kulkeutuvasta liikenteestä, sekä jokaiseen osioon oma selitekenttä, jossa kerrotaan, mitä tietoa graafeista löytyy, sekä miten siitä voidaan hyötyä.

Statistiikassa lasketaan avattuja yhteyksiä ja yhteysyrityksiä lukumäärittäin. Tämä ei välttämättä korreloi liikkuneeseen datamäärään. Lisäksi statistiikka koskee pääsääntöisesti ainoastaan sallittua liikennettä, ellei raportissa erikseen mainita myös blokattua liikennettä.

Raportista löytyvät seuraavat kohdat:

- Maakohtainen liikenne sisäänpäin.
- Sallittu ja blokattu liikenne.
- Sallittu ja blokattu liikenne suunnittain (sisään vai ulos).
- Liikenne ulkoverkosta sisäverkkoon.
- Liikenne sisäverkosta ulkoverkkoon.
- IDS Hälytykset.

Maakohtainen liikenne sisäänpäin

Listaukseen sisällytetty ainoastaan muurin läpi päässyt liikenne ulkoverkosta kohti sisäverkkoa. Listaus ei siis huomioi, mihin päin sisäverkosta liikennöidään, vaan ainoastaan ulkoverkosta tulevaa liikennettä.

Country	Count
Finland	162,061
China	59,305
Russia	53,269
Hong Kong	33,699
United Kingdom	30,412
United States	29,044
Netherlands	25,924
Germany	18,199
Republic of Korea	13,597
Vietnam	11,651
	437,161

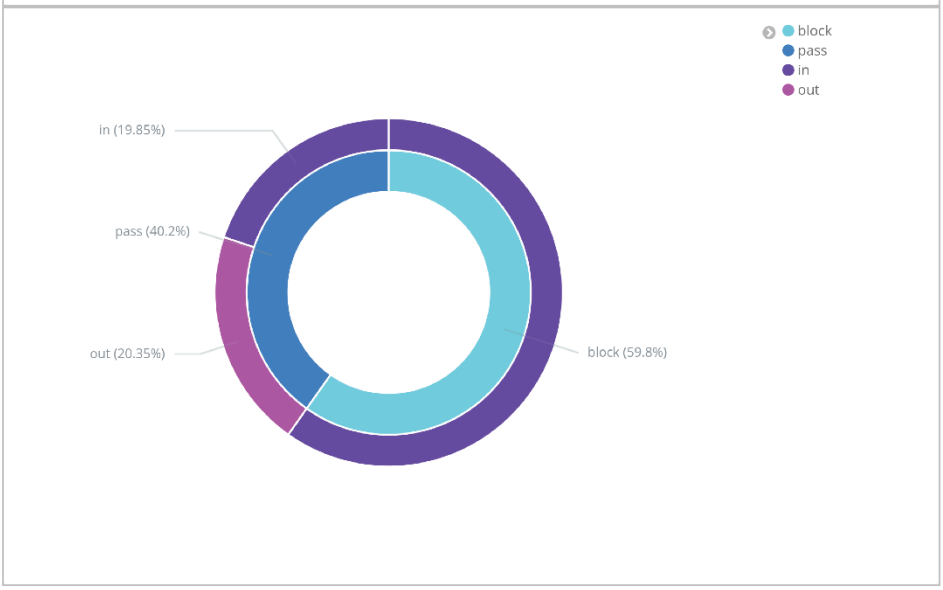
Sallittu ja blokattu liikenne

Graafiin piirtyy kolme viivaa: Sallittujen ja blokattujen yhteysyritysten määrä, sekä näiden summa. Graafi piirretään neljän tunnin keskiarvoa laskien, jotta siitä voitaisiin havaita vielä jotakin poikkeamia, mutta ei samaan aikaan olisi liian vaikeasti luettavissa.

Sallittu ja blokattu liikenne suunnittain

Ympyrägraafista voidaan lukea muurin molemmat liikennesuunnat yksitellen, kuinka paljon liikennettä osuu muurisääntöihin sallittuna ja blokattuna. Liikenne mitataan yhteysyrityksinä, ei itse liikutetun datan määränä.

Tyypillisesti muuri sallii huomattavasti enemmän ulospäin, kuin sisäänpäin ja vastaavasti taas blokkaa liikennettä enemmän ulkoverkosta. Usein on myös mahdollista, että sisäverkosta kaikki liikenne on sallittua, jolloin kentän Direction: Out kenttä Action: Pass saa arvon 100%.

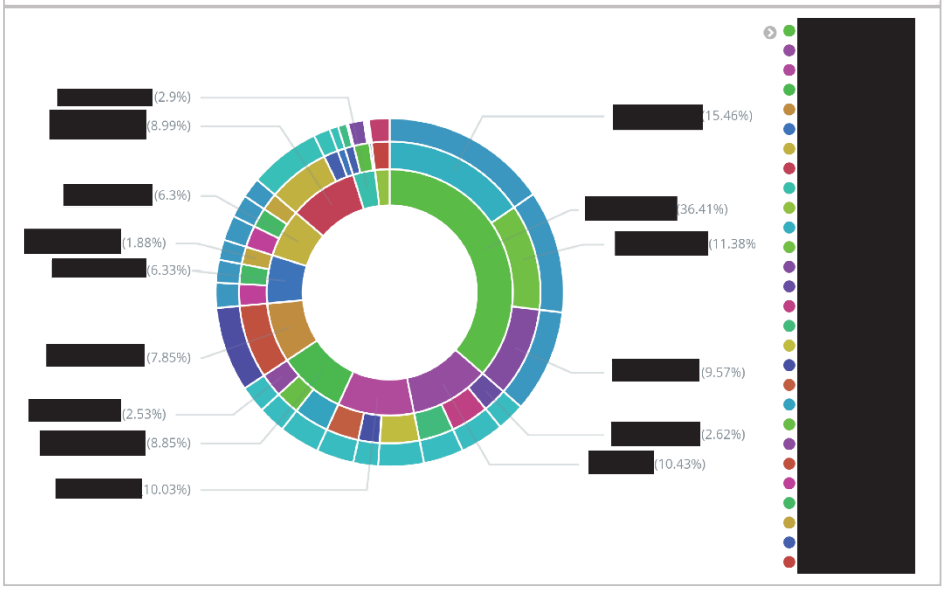


Yhteydet ulkoverkosta sisäverkkoon

Ympyrägraafiin on melko tarkastikin kuvattu sallittua liikennettä ulkoverkosta kohti sisäverkkoa. Sisärenkaalla on kymmenen yleisintä kohdeosoitetta (eli sisäverkon IP-osoitetta). Toisella renkaalla on lähdeosoite (eli ulkoverkon osoite) ja viimeisellä renkaalla on liikenteen kohdeportti.

Graafista kannattaa yrittää etsiä kummallisuuksia, kuten RDP (portti 3389) tai SSH (portti 22) yhteyksiä ulkoverkon osoitteista, jotka eivät ole tunnettuja ja luotettavia. Mikäli tuntemattomasta osoitteesta tulee paljon yhteyksiä esimerkiksi porttiin 3389, voidaan olettaa, että kohdeosoitetta yritetään murtaa RDP-yhteyksien avulla ja tämä tulisi erikseen blokata palomuurilla.

Graafin alapuolella taulukko, johon on merkitty eniten liikennettä vastaanottavat kohdeosoitteet, sekä kyseiselle osoitteelle yleisin portti yleisimmistä lähdeosoitteesta.



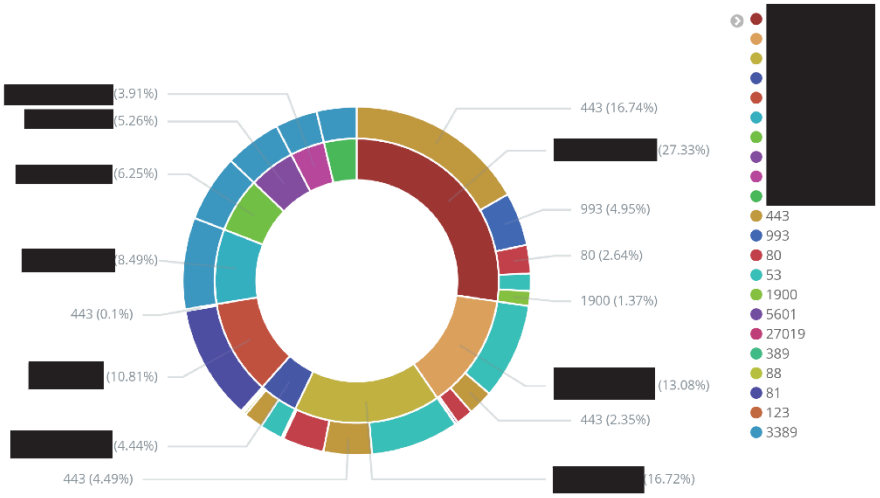
Destination IP	Destination Port	Source IP	Count
[redacted]	3389	[redacted]	23,164
	ICMP		348
	ICMP		320
	ICMP		7,442
	81		29,032
	3389		1,346
	3389		1,341
	53		18,502
	49157		3,966
	771		7,082
			92,543

Liikenne sisäverkosta ulkoverkkoon

Ympyrägraafin ensimmäisellä kerroksella on kymmenen yleisintä sisäverkon osoitetta, joista liikennöidään ulkoverkkoon sallitusti. Toisella kerroksella on kyseisille lähdeosoitteille yleisimmät portit, joihin laitteet yrittävät liikennöidä.

Graafiin ei olla laitettu kohdeosoitetta, sillä emme voi Suomen lakia noudattaen valvoa käyttäjien liikennettä liian tarkasti. Tästä huolimatta graafista voidaan havainnoida mahdollisesti saastunut sisäverkon laite, jos havaitaan, että jokin laite käyttää paljon kohdeportteja, joita ei tiedetä laitteen käyttäjän käyttävän. Näitä ovat esimerkiksi SNMP portit 161 ja 162, SSH portti 22 ja RDP portti 3389. Näitä portteja voidaan käyttää esimerkiksi kun saastunut kone yrittää kirjautua sisäverkossa muihin laitteisiin.

Alapuolella taulukko, johon listattu nämä yleisimmät lähdeosoitteet, sekä jokaiselle lähdeosoitteelle yleisin kohdeportti.



Source IP	Destination Port	Count
[redacted]	443	40,595
	53	24,177
	53	22,067
	53	5,457
	81	29,032
	3389	23,167
	3389	17,046
	3389	14,350
	3389	10,676
	3389	10,132
		196,699

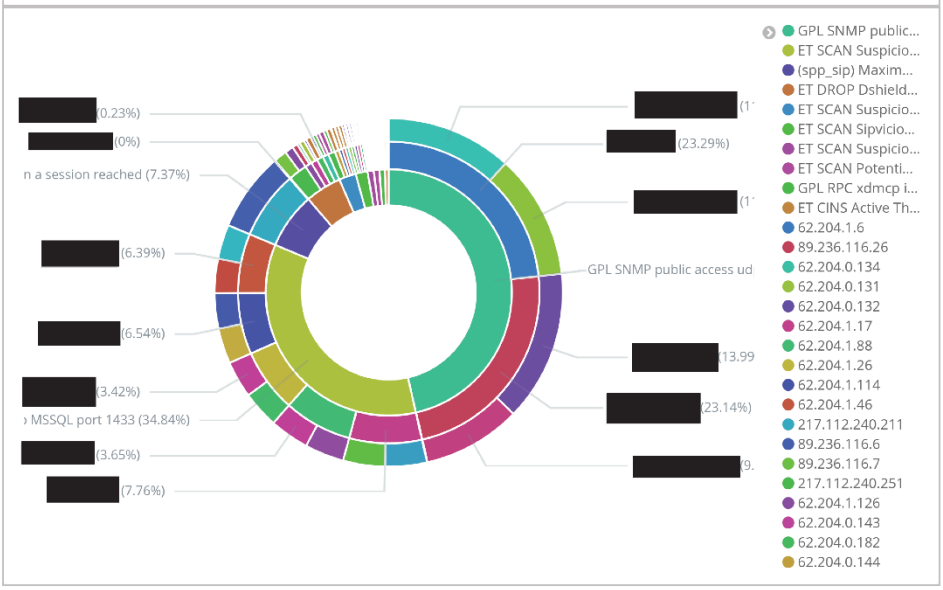
IDS Hälytykset

Intrusion Detection Service (IDS) analysoi liikennettä ja pyrkii havainnoimaan haittaliikenteen, kuten esimerkiksi SQL-injektiot, porttiskannaukset ja bottiverkkoon saastuneet laitteet. IDS ei kuitenkaan suoraan estä tätä liikennettä, sillä todellisuudessa haitallinen liikenne on hyvinkin sisäverkkokohtaista ja tämän vuoksi varsinkin alussa saattaa oikeatakin liikennettä joutua blokkiin paljon.

Käyttämämme IDS työkalu SNORT jaottelee uhat kolmeen eri vakavuusasteeseen, low, medium ja high. Raporttiin on otettu mukaan vain vakavuudet medium ja high. Graafin ensimmäisellä kierroksella on yleisimmät syyt IDS:n hälytykselle, toisella renkaalla kohdeosoite ja kolmannella renkaalla lähdeosoite. Liikenteen suunta on pääteltävissä näillä osoitteilla, sekä hälytyksen syyllä.

Hälytyksille löytyy hyvin tarkempaa tietoa hakukoneisiin raportista löytyvän syyn syöttämällä. Valitettavasti näitä ei voida yksitellä tässä raportissa, sillä erilaisia syitä on satoja.

Graafin alapuolella listattuna taulukkoon yleisimmät syyt, sekä jokaiselle syylle yleisin kohde- ja sille yleisin lähdeosoite.



Reason	Destination IP	Source IP	Count
GPL SNMP public access udp	[redacted]	[redacted]	7,283
(spp_sip) Maximum dialogs within a session reached			4,786
ET SCAN Sipvicious User-Agent Detected (friendly-scanner)			22
ET SCAN Suspicious inbound to mySQL port 3306			6
ET CINS Active Threat Intelligence Poor Reputation IP TCP group 4			5
ET DROP Dshield Block Listed Source group 1			5
ET SCAN Potential SSH Scan			4
GPL RPC xdmcp info query			2
ET SCAN Suspicious inbound to MSSQL port 1433			2
ET SCAN Suspicious inbound to PostgreSQL port 5432			1
			12,116

Liite 8. Valmis raportti

TNNet Oy Palomuuriraportti

Raportista löytyy statistiikkaa palomuurille asti kulkeutuvasta liikenteestä, sekä jokaiseen osioon oma selitekenttä, jossa kerrotaan, mitä tietoa graafeista löytyy, sekä miten siitä voidaan hyötyä.

Statistiikassa lasketaan avattuja yhteyksiä ja yhteysyrityksiä lukumäärittäin. Tämä ei välttämättä korreloi liikkuneeseen datamäärään. Lisäksi statistiikka koskee pääsääntöisesti ainoastaan sallittua liikennettä, ellei raportissa erikseen mainita myös blokattua liikennettä.

Raportista löytyvät seuraavat kohdat:

- Maakohtainen liikenne sisäänpäin.
- Sallittu ja blokattu liikenne.
- Sallittu ja blokattu liikenne suunnittain (sisään vai ulos).
- Liikenne ulkoverkosta sisäverkkoon.
- Liikenne sisäverkosta ulkoverkkoon.
- IDS Hälytykset.

Maakohtainen liikenne sisäänpäin

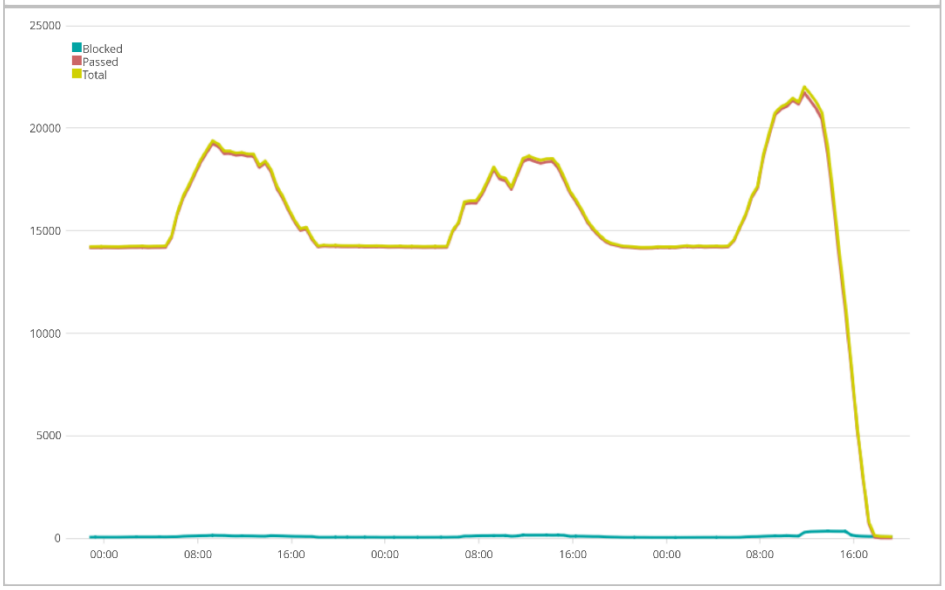
Listaukseen sisällytetty ainoastaan muurin läpi päässyt liikenne ulkoverkosta kohti sisäverkkoa. Listaus ei siis huomioi, mihin päin sisäverkosta liikennöidään, vaan ainoastaan ulkoverkosta tulevaa liikennettä.

Finland

Country	Count
Finland	103
	103

Sallittu ja blokattu liikenne

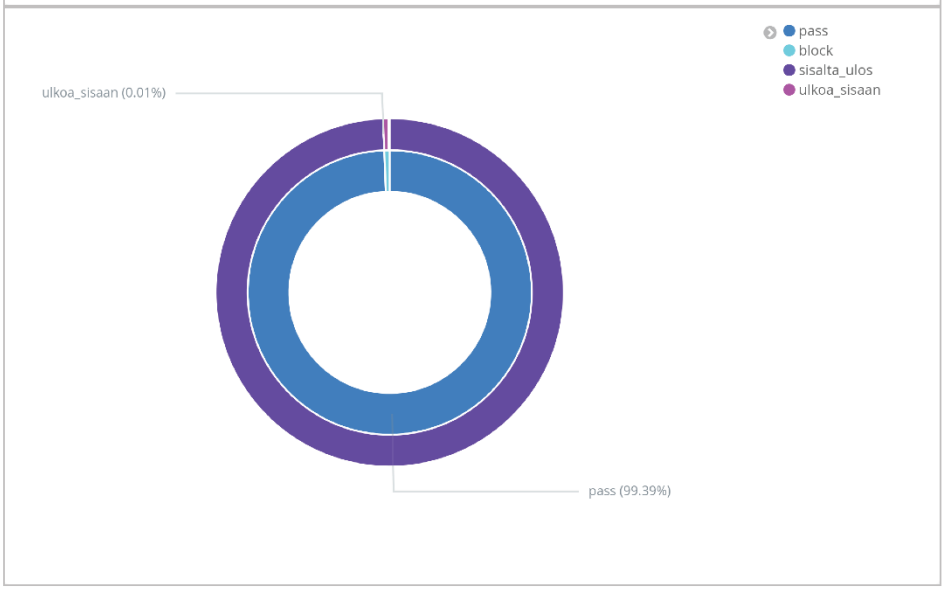
Graafiin piirtyy kolme viivaa: Sallittujen ja blokattujen yhteysyritysten määrä, sekä näiden summa. Graafi piirretään neljän tunnin keskiarvoa laskien, jotta siitä voitaisiin havaita vielä jotakin poikkeamia, mutta ei samaan aikaan olisi liian vaikeasti luettavissa.



Sallittu ja blokattu liikenne suunnittain

Ympyrägraafista voidaan lukea muurin molemmat liikennesuunnat yksitellen, kuinka paljon liikennettä osuu muurisääntöihin sallittuna ja blokattuna. Liikenne mitataan yhteysyrityksinä, ei itse liikutetun datan määränä.

Tyypillisesti muuri sallii huomattavasti enemmän ulospäin, kuin sisäänpäin ja vastaavasti taas blokkaa liikennettä enemmän ulkoverkosta. Usein on myös mahdollista, että sisäverkosta kaikki liikenne on sallittua, jolloin kentän Direction: Out kenttä Action: Pass saa arvon 100%.

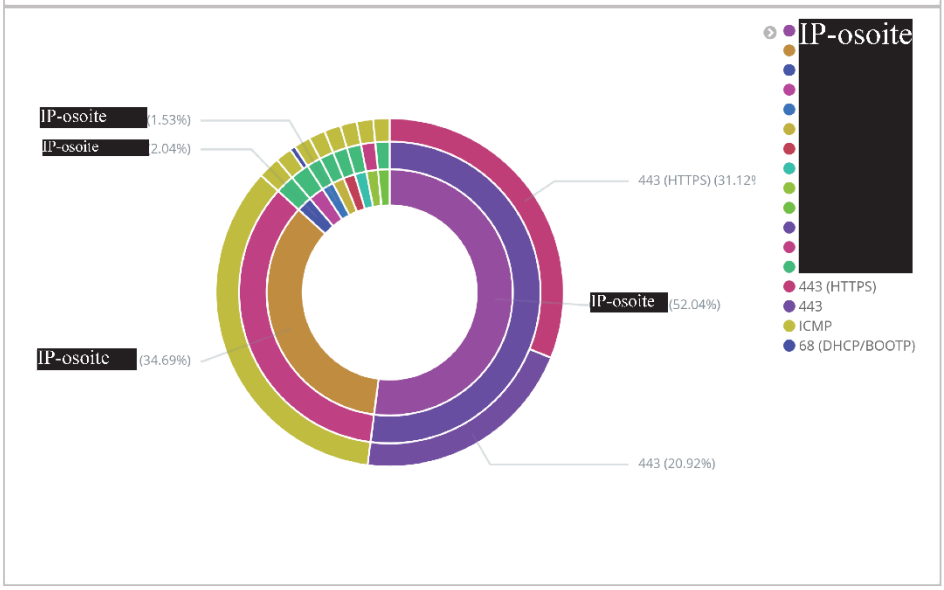


Yhteydet ulkoverkosta sisäverkkoon

Ympyrägraafiin on melko tarkastikin kuvattu sallittua liikennettä ulkoverkosta kohti sisäverkkoa. Sisärenkaalla on kymmenen yleisintä kohdeosoitetta (eli sisäverkon IP-osoitetta). Toisella renkaalla on lähdeosoite (eli ulkoverkon osoite) ja viimeisellä renkaalla on liikenteen kohdeportti.

Graafista kannattaa yrittää etsiä kummallisuuksia, kuten RDP (portti 3389) tai SSH (portti 22) yhteyksiä ulkoverkon osoitteista, jotka eivät ole tunnettuja ja luotettavia. Mikäli tuntemattomasta osoitteesta tulee paljon yhteyksiä esimerkiksi porttiin 3389, voidaan olettaa, että kohdeosoitetta yritetään murtaa RDP-yhteyksien avulla ja tämä tulisi erikseen blokata palomuurilla.

Graafin alapuolella taulukko, johon on merkitty eniten liikennettä vastaanottavat kohdeosoitteet, sekä kyseiselle osoitteelle yleisin portti yleisimmistä lähdeosoitteesta.



Destination IP	Source IP	Destination Port	Count
IP-osoite	IP-osoite	443 (HTTPS)	61
		ICMP	68
		ICMP	4
		ICMP	3
		ICMP	3
		ICMP	3
		ICMP	3
		ICMP	3
		ICMP	3
			154

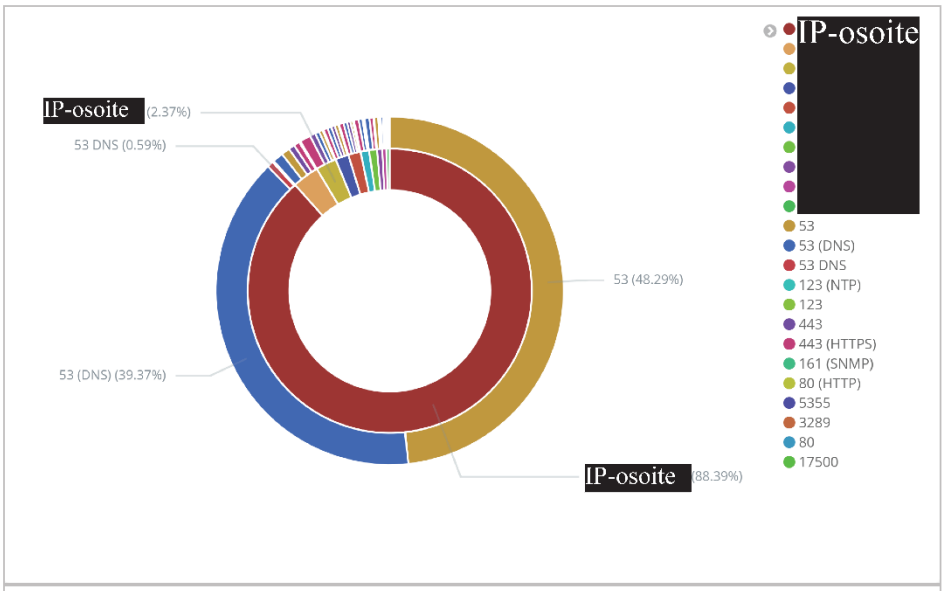
Liikenne sisäverkosta ulkoverkkoon

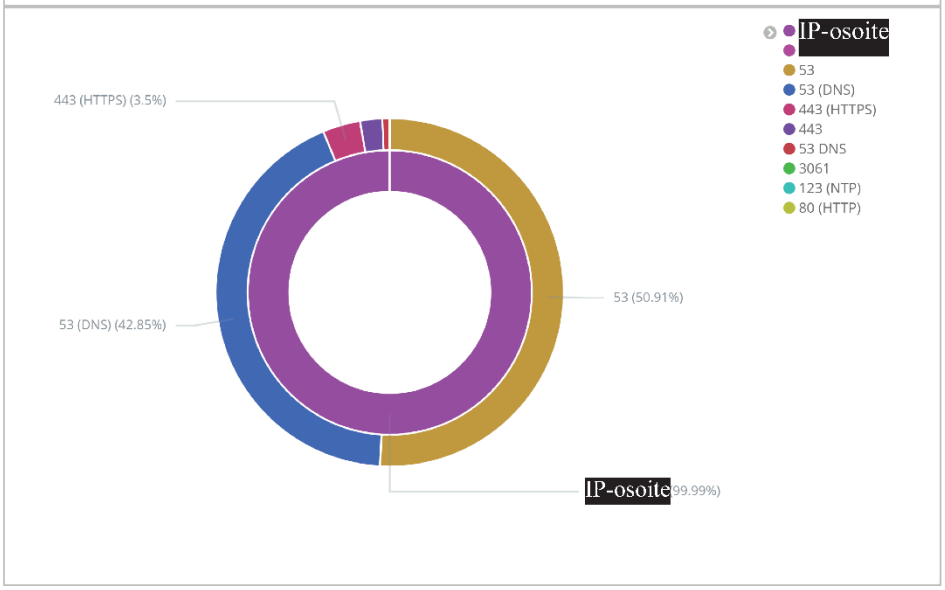
Ympyrägraafin ensimmäisellä kerroksella on kymmenen yleisintä sisäverkon osoitetta, joista liikennöidään ulkoverkkoon sallitusti. Toisella kerroksella on kyseisille lähdeosoitteille yleisimmät portit, joihin laitteet yrittävät liikennöidä.

Graafiin ei olla laitettu kohdeosoitetta, sillä emme voi Suomen lakia noudattaen valvoa käyttäjien liikennettä liian tarkasti. Tästä huolimatta graafista voidaan havainnoida mahdollisesti saastunut sisäverkon laite, jos havaitaan, että jokin laite käyttää paljon kohdeportteja, joita ei tiedetä laitteen käyttäjän käyttävän. Näitä ovat esimerkiksi SNMP portit 161 ja 162, SSH portti 22 ja RDP portti 3389. Näitä portteja voidaan käyttää esimerkiksi kun saastunut kone yrittää kirjautua sisäverkossa muihin laitteisiin.

Osiossa on kaksi ympyrägraafia, jotka mittaavat samaa asiaa, mutta eri rajapinnasta: Ensimmäinen graafi kuvastaa pelkkiä sisäverkon osoitteita, kun taas toinen graafi kuvastaa saman liikenteen, mutta lähdeosoitteena on sisäverkon osoitteiden 1:1 NAT:n jälkeinen, ulkomaailmaan näkyvä julkinen osoite.

Alapuolella taulukko, johon listattu nämä yleisimmät sisäverkon lähdeosoitteet, sekä jokaiselle lähdeosoitteelle yleisin kohdeportti. Taulukko siis vastaa ensimmäistä ympyrägraafia.


IP-osoite
IP-osoite (2.37%)
53 DNS (0.59%)
53 (48.29%)
53 (DNS) (39.37%)
IP-osoite (88.39%)
53
53 (DNS)
53 DNS
123 (NTP)
123
443
443 (HTTPS)
161 (SNMP)
80 (HTTP)
5355
3289
80
17500


IP-osoite
443 (HTTPS) (3.5%)
53 (50.91%)
53 (DNS) (42.85%)
IP-osoite (99.99%)
53
53 (DNS)
443 (HTTPS)
443
53 DNS
3061
123 (NTP)
80 (HTTP)

Source IP	Destination Port	Count
IP-osoite	53	513,254
	53 (DNS)	9,037
	443 (HTTPS)	10,190
	443 (HTTPS)	3,802
	443 (HTTPS)	3,545
	443 (HTTPS)	4,119
	53 (DNS)	4,919
	53	3,768
	53 (DNS)	2,064
	53 (DNS)	1,005
		555,703

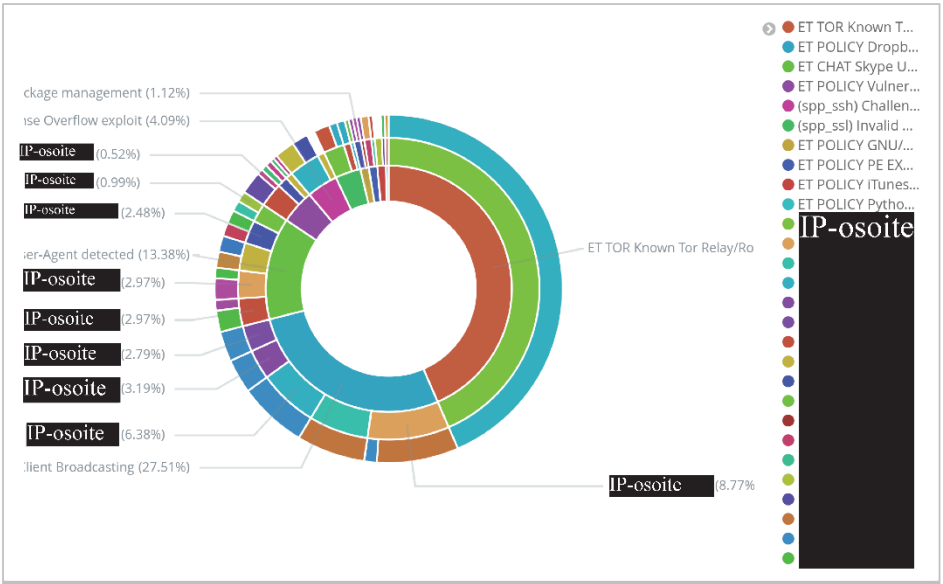
IDS Hälytykset

Intrusion Detection Service (IDS) analysoi liikennettä ja pyrkii havainnoimaan haittaliikenteen, kuten esimerkiksi SQL-injektiot, porttiskannaukset ja bottiverkkoon saastuneet laitteet. IDS ei kuitenkaan suoraan estä tätä liikennettä, sillä todellisuudessa haitallinen liikenne on hyvinkin sisäverkkokohtaista ja tämän vuoksi varsinkin alussa saattaa oikeatakin liikennettä joutua blokkiin paljon.

Käyttämämme IDS työkalu SNORT jaottelee uhat kolmeen eri vakavuusasteeseen, low, medium ja high. Raporttiin on otettu mukaan vain vakavuudet medium ja high. Graafin ensimmäisellä kierroksella on yleisimmät syyt IDS:n hälytykselle, toisella renkaalla kohdeosoite ja kolmannella renkaalla lähdeosoite. Liikenteen suunta on pääteltävissä näillä osoitteilla, sekä hälytyksen syyllä.

Hälytyksille löytyy hyvin tarkempaa tietoa hakukoneisiin raportista löytyvän syyn syöttämällä. Valitettavasti näitä ei voida yksitellä tässä raportissa, sillä erilaisia syitä on satoja.

Graafin alapuolella listattuna taulukkoon yleisimmät syyt, sekä jokaiselle syylle yleisin kohde- ja sille yleisin lähdeosoite.



Reason	Destination IP	Source IP	Count
ET TOR Known Tor Relay/Router (Not Exit) Node UDP Traffic group 221	IP-osoite	IP-osoite	117
ET POLICY Dropbox Client Broadcasting			13
(spp_ssh) Challenge-Response Overflow exploit			5
ET POLICY Vulnerable Java Version 1.8.x Detected			4
(spp_ssl) Invalid Client HELLO after Server HELLO Detected			4
ET CHAT Skype User-Agent detected			2
ET POLICY PE EXE or DLL Windows file download HTTP			1
ET POLICY GNU/Linux YUM User-Agent Outbound likely related to package management			1
ET POLICY iTunes User Agent			1
			148