Iiro Sipari

AVOIMEN LÄHDEKOODIN PALOMUURIN MÄÄRITTELYN PÄIVITYS

Tietojenkäsittelyn koulutusohjelma

2015

samk
Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

AVOIMEN LÄHDEKOODIN PALOMUURIN MÄÄRITTELYN PÄIVITYS

Sipari, Iiro
Satakunnan ammattikorkeakoulu
Liiketalouden koulutusohjelma
Maaliskuu 2015
Ohjaaja: Grönholm, Jukka
Sivumäärä: 38
Liitteitä: 1

Asiasanat: Palomuuri, FreeBSD, Packet Filter, Firewall Builder

---

Tämän opinnäytetyön aiheena oli testata Firewall Builder nimistä ohjelmaa jo valmiiksi luodun palomuurin hallintaan. Alkuperäinen palomuuri on Packet Filter ohjelmassa toimiva kirjoitettu configuration tiedosto joka toimii ainoastaan konsolista. Tarkoituksena oli saada vanhan palomuurin sääntöjä helpommin käsiteltävään muotoon ja määritellä uudelleen vanhoja sääntöjä.

Opinnäytetyö sisältää myös laajempaa tietoa palomuurien ja verkon toiminnasta. Työssä käsitellään useiden eri palomuuri tyyppien toimintaa, aliverkotusta, TCP/IP- ja OSI-mallia ja FreeBSD:tä joka toimii käyttöjärjestelmänä alkuperäiselle palomuurille. Työssä käydään läpi myös Firewall Builder ja Packet Filtter ohjelmat ja niiden toiminta.

Työn johtopäätös on, että Firewall Builder ei sovellu tässä tapauksessa korvaamaan Packet Filtteriä. Firewall Builder aiheuttaisi suuren määrän kommenttien katoamista ja myös nimeäsi jo valmiit Packet Filtterillä tehdyt taulut uudelleen. Myös uuden syntaksin opettelu veisi aikaa. Mikäli palomuuri määriteltäisiin kokonaan alusta olisi mahdollista tehdä se Firewall Builderilla, mutta tulos ei välttämättä olisi yhtä hyvä kuin alkuperäinen.

SPECIFICATION UPDATE OF OPEN SOURCE FIREWALL

Sipari, Iiro
Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences
Degree Programme in Computer Science
March 2015
Supervisor: Grönholm, Jukka
Number of pages: 38
Appendices: 1

Keywords: Firewall, FreeBSD, Packet Filter, Firewall Builder

The theme of this thesis was to test program named a Firewall Builder on an already existing firewall. This was to ease the management of the firewall. The original firewall is a Packet Filter, this firewall is basically just a configuration file that works through a console. The goal was to get a Packet Filter to work with Firewall Builder to get a graphical user interface and make it easier to manage.

This thesis also includes broader information about networks and firewalls. The thesis includes the basics of different firewall types, subnetting, introduction OSI and TCP/IP models and information about FreeBSD. This also includes information about how Firewall Builder and Packet Filter work.

The conclusion of this thesis is that Firewall Builder is not suitable to use in existing a firewall. The migration would cause a great deal of comments in the configuration file to disappear and cause table names to change which would make things difficult to manage. To get Firewall Builder to work properly, it would be best to start completely from the scratch but the result would not necessarily be better than if it was done just with Packet Filter.

TERMIT JA LYHENTEET

DD-WRT	Firmware langattomille reitittimille.
DNS	Domain Name System eli Internetin nimipalvelujärjestelmä. Muuttaa verkkotunnuksia IP-osoitteiksi.
DHCP	Dynamic Host Configuration Protocol verkkoprotokolla joka jakaa IP osoitteita lähiverkkoon kytketyille laitteille.
FTP	Fire transfer protocol. Tiedon siirto protokolla joka käyttää TCP protokollaa
FWBuilder	Firewall Builder. Ohjelma toimii graafisena ympäristönä palomuurin rakentamiselle.
HTTP	Hypertext transfer protocol. Tiedonsiirto protokolla jota selaimet käyttävät, toimii TCP välityksellä.
IMAP4	Internet Message Access Protocol. Sähköpostin lukemiseen tarkoitettu protokolla.
IPv4	Internetin protokolla. Koostuu tavallisesti neljän pisteellä erotetun luvun sarjana kuten 192.169.0.113.
IPv6	Uudempi versio protokollasta. Osoite muodostuu tyypillisesti kirjaimista ja numeroista eriteltynä kaksoispisteellä Esim: 2001:db8::ff00:42:8329
IPSec	TCP/IP perheeseen kuuluva protokolla joka turvaa kahden yhteydessä olevan osapuolen salauksen ja todennuksen. Integroitu IPv6 protokolaan.
NAT	Network address translation. IP-Osoitteen muunnos.
NGFW	Next Generation Firewall, ohjelmakohtainen palomuuri
NetId	Network ID on osa TCP/IP osoitetta jolla tunnistetaan henkilöitä tai koneita jotka ovat esimerkiksi liittyneet lähiverkossa.
OpenWRT	Linux pohjainen käyttöjärjestelmä yleisesti käytetty reitittimissä.
PAT	Port Address translation. Mahdollistaa monen sisäverkon laitteen jakaa yhden IP osoitteen.

PAR	Positive Acknowledgement Retransmission, protokolla TCP/IP sarjassa.
PF	Packetfilter, freebsd käyttöjärjestelmässä toimiva palomuuri.
Proxy	Välityspalvelin. Toimii välittäjänä datansiirrossa, voidaan käyttää osoitteiden alkuperän piilottamiseen.
Syntaksi	Varattujen sanojen ja lauseiden tunnistus ohjelmointikielessä.
SSH	Secure Shell eli salattuun tietoliikenteeseen tarkoitettu protokolla, tätä käytetään yleensä etäyhteyden muodostamiseen.
Table	Packet filterissä voidaan monia eri osoitteita asetaa yhteen, tälläistä IP osoitteiden yhdistynyttä kokoelmaa kutsutaan tabletksi, tämä helpottaa moneen osoitteeseen viittaamista.
TCP	Transmission Control Protocol. Protokollan avulla luodaan yhteyksiä tietokoneiden välille jotka ovat yhteydessä internettiin.
UDP	User Datagram Protocol. Kuten TCP mutta ei vaadi yhteyttä laitteiden välille, mahdollistaa tiedostojen siirron.
UTM	Unified Threat management, ohjelma joka sisältää paljon eri suojaus ohjelmia, yhdessä ohjelmassa.
VPN	Virtual Private Network. Tekniikka jonka avulla yksityisiä verkkoja voidaan yhdistää virtuaalisesti.

SISÄLLYS

1 JOHDANTO

Tämä opinnäytetyö on tehty Porin Opetusteknologiakeskukselle. Porin Opetusteknologiakeskus on vastuussa Porin alueella sijaitsevien koulujen ja lukioiden tietotekniikasta. Porin opetusteknologia keskus tunnetaan myös nimellä Opetek. Työn aiheena oli selvittää olisiko FreeBSD käyttöjärjestelmässä toimivalle palomuurille, Packet Filterille graafista käyttöliittymää joka auttaisi vanhan palomuurin hallinnassa.

Tarve tähän graafiseen käyttöliittymään johtui siitä, että palomuuri on ollut monen eri henkilön hallinnassa ja monissa kouluissa, jota palomuuri suojaa on tehty muutoksia. Tämä on saanut aikaan sen, että monia sääntöjä jotka eivät ole enää käytössä ja monet näistä säännöistä ovat melko sekalaisia ja pitäisi siivota ja järjestellä uudelleen. Graafinen käyttöliittymä helpottaisi asiaa huomattavasti administraation kannalta.

Palomuurin graafisen käyttöliittymän testauksen lisäksi opinnäytetyö esittelee yleisimmät palomuuri tyypit ja niiden toiminnan perusteet helposti ymmärrettävällä tavalla, työ myös käsittelee palomuureihin liittyviä asioita kuten lähiverkotusta ja OSI- ja TCP/IP mallia, koska nämä ovat tärkeitä asioita jotka tulee hallita palomuuria tehdessä tai korjatessa. Lisäksi työ käsittelee erilaisia palomuuri tyyppejä ja niiden erilaisia käyttö tarkoituksia. Näistä teoreettisista asioista johtuen tätä työtä voi myös käyttää nopeana palomuureihin tutustuttavana dokumenttina, koska opinnäytetyö käy läpi monet tärkeät asiat joita tulee ottaa huomioon palomuuria käyttöönottaessa.

Valitsin aiheekseni palomuurit, koska olen aina ollut kiinnostunut tietoturvasta ja palomuurien toimintaperiaatteista. Minua on myös aina kiinnostanut unix-pohjaiset käyttöjärjestelmät. Tämä työ antoi mahdollisuuden tutkia palomuurin toimintaa syvemmin ja opetella myös Packet Filterin syntaksia. Bonuksena sain myös hieman tutkia FreeBSD käyttöjärjestelmää, josta minulla ennen tätä työtä oli erittäin vähän tietoa.

2 PALOMUURIT

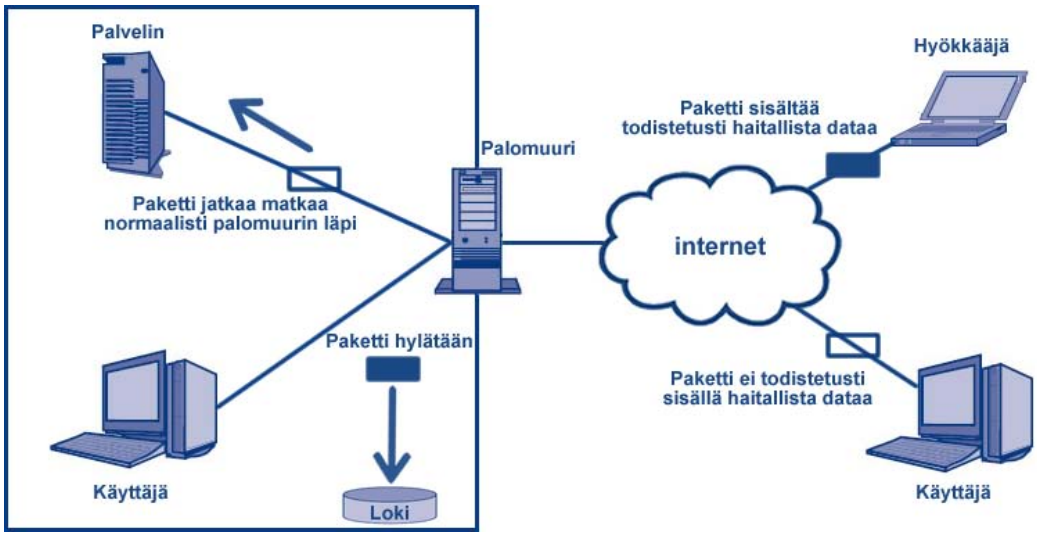
Palomuurit toimivat protokollan päätepisteenä, pakettisuodattimina, näiden hybridinä tai proxynä. Kun palomuuri toimii protokollan päätepisteenä ja releenä se mahdollistaa turvallisen osajoukon protokollalle, tekee laajoja protokollan validiteettitarkistuksia, toimii eristetyssä ympäristössä tai on jokin yhdistelmä näitä. Pakettisuodattimet joko päästävät datapaketin läpi, hylkäävät sen tai käsittelevät sen omalla tavallaan. Palomuurit tyypillisesti tekevät päätökset IP lähde- ja saapumisosoitteiden ja porttinumeroiden avulla. Palomuureja suunnitellessa on myös tärkeää, että palomuurin toiminta ei estä muiden ohjelmien toimintaa. (Freed 2000, 1-2)

Jokainen palomuurin läpi kulkeva datapaketti tutkitaan. Mikäli paketti on todistetusti haitallinen, palomuuri estää kyseisen paketin. Mikäli pakettia ei pystytä todistamaan haitalliseksi, palomuuri päästää sen kohdeosoitteeseen. Tätä kutsutaan palomuureissa pass/deny päätökseksi. On hyvä ottaa huomioon se, että mikäli pakettia ei pystytä todistamaan haitalliseksi se voi siitä huolimatta olla vaarallinen. Tämän takia palomuureja konfiguroidessa on oltava hyvin tarkka sääntöjä tehdessä. Tästä huolimatta mikään palomuuri; parhaista suojauksista huolimatta ei ole koskaan täysin turvallinen.

Pakettien suodatuksen lisäksi toinen palomuurien tärkeä toiminnallisuus on liikenteen kirjaaminen lokiin. Palomuuri pitää automaattisesti lokia kaikkien tiputettujen ja läpipäästettyjen pakettien tiedoista. Lokien käsittelyyn on myös omia apuja kuten Syslog, koska lokit kasvavat nopeasti hyvinkin suuriksi. Palomuurin ylläpidon olisi hyvä käydä loki läpi päivittäin tai useammin ymmärtääkseen yhtiöön kohdistetut hyökkäykset. Vaikka palomuuri estäisikin suurimman osan tietyn tyyppisistä hyökkäyksistä, se ei tarkoita että kaikki hyökkäykset olisivat epäonnistuneet. Mikäli tietystä IP osoitteesta tulee paljon hyökkäyksiä, voidaan palomuurin säännöt asettaa estämään kaikki paketit kyseisestä osoitteesta.

Tärkeä asia palomuureissa on myös sisääntulon ja ulosmenon suodatus. Sisääntulon suodatuksella tarkoitetaan tietoa joka tulee verkon ulkopuolelta, eli tyypillisesti internetistä. Suodatuksen tarkoitus on tietenkin estää haitallisia paketteja pääsemästä sisäverkkoon. Ulosmenon suodatus on yhtä tärkeää mutta vähemmän käytettyä vaikkakin se on yleistymässä. Ulosmenon suodatuksella estetään pakettien pääsy verkosta takaisin internetiin. Tämä estää sen että niin kutsutut "probe packets" eli paketit, jotka ottavat tietoa verkosta, eivät pääse takaisin viemään tietoa palomuurista ja heikkouksista takaisin hyökkääjälle. Lähtevän liikenteen suodatus estää myös infektoituneen koneen yhteyden muihin koneisiin, eli pistää saastuneet koneet karanteeniin. (Boyle & Panko 2013, 314)

Kuva on esimerkki palomuurin toiminnoista.



Kuva 1: Palomuurin toiminta

2.1 Palomuureista yleisesti

Yhtiön varustaminen riittävällä tietoturvalla on edelleen erittäin tärkeä ja haasteellinen asia tämän päivän tietoturvasta vastaavalle. Osa vaikeudesta johtuu jatkuvasti kasvavasta tietokoneen ja internetin välillä käyvästä liikenteestä. Tämä on johtanut siihen, että resurssit, tieto ja palvelut, jotka kulkevat internetin ja koneiden välillä ovat kasvaneet valtavasti, valitettavasti tämä on myös lisännyt suuresti tietoturva riskejä. Suuri määrä liikennettä ja palveluita luo hyökkääjälle paljon enemmän aukkoja

ja mahdollisuuksia päästä käsiksi liikenteeseen tai tietoihin, jotka pitäisivät olla suojattua. (Vacca 2009, 213)

Palomuurit ovat yksi tärkeimmistä suojauskomponenteista kaikkien tietoverkkojen turvallisuudelle. Toisin kuin yleisesti luullaan, palomuuri ei ole yksittäinen systeemi, vaan oikeastaan se muodostuu useammasta komponentista. Palomuuri asetetaan yleensä kahden tai useamman verkon välille ja toimii täten porttina näiden verkkojen välillä tutkien tietoa, joka kulkee sen läpi. Palomuurin pääasialliset tarpeet voidaan selittää lyhyesti seuraavasti.

1. Kaiken verkon liikenteen on kuljettava palomuurin läpi.
2. Palomuuri päästää läpi vain valtuutetun liikenteen.
3. Palomuurin tulisi olla immuuni tunkeutumiselle ja muille vaaroille.

Perussääntönä yhtiön ei koskaan pitäisi liittää omaa verkkoaan mihinkään ulkoiseen verkkoon ilman palomuuria. Laitteistosta ja ohjelmista, joista palomuuri koostuu suojaa yhtiön koneita ja verkkoa ulkoisilta hyökkäyksiltä. Palomuuria toteuttaessa sisäiseen verkkoon, tulee olla varma, että toimenpiteet ovat varmasti tarpeeksi hyviä ottaen huomioon vaarat, jotka ovat jo olemassa.

Suojatussa ympäristössä on tärkeä ylläpitää datan ja resurssien yksityisyyttä, yhtenäisyyttä ja saatavuutta. Yksityisyydellä tarkoitetaan tiedon kulkemista niille tarkoitetuilla kanavilla ja täten estäen sen pääsyn tahoille joille se ei kuulu. Yhtenäisyydellä viitataan tiedon tarkkuuteen ja sen luotettavuuteen. Tähän liittyy myös tiedon eheys, jolloin tietoa ei ole muutettu ilman lupaa sekä lähteen eheys eli tieto on alkuperäisesti juuri sieltä mistä se väittää olevansa. Saatavuudella tarkoitetaan tietenkin datan ja resurssien valmiutta ja niiden saatavuutta niille jotka, näitä tietoja tarvitseva. Nämä perussäännöt voivat hyvinkin määritellä yhtiön maineen, joka on yhtiölle hyvin tärkeä asia. (Vacca 2009, 213)

2.2 Henkilökohtainen palomuuri

Henkilökohtaiset palomuurit ovat tarkoitetut suojaamaan vain yhtä laitetta luvattomilta yhteyksiltä. Vuosien saatossa tämä on kehittynyt suuresti ja nykyään palomuuri on sisällytetty myös viirustorjuntaan ja IDS:ään. Pienyrityksissä ja kotikäytössä nämä ovat erittäin järkeviä päätöksiä. Ne antavat tarvittavan suojan ja ne ovat helposti hallittavissa. Suuremmissa yrityksissä ongelmat ovat monimutkaisempia ja tämän tyylinen suojaus ei ole riittävä suuren organisaation suojaamiseen. Laajassa verkossa, jossa on monia koneita eri aliverkoissa; palomuurit jotka eivät pysty erottelemaan sisäistä liikennettä eivät ole tarpeeksi turvallisia. Yrityksissä on myös monesti useampia palomuureja hallinnon helpottamiseksi. (Noonan & Dubrawsky 2006, Firewall Products)

2.3 Verkkokerroksen palomuuri

Verkon palomuurit ovat suunniteltu siten, että ne suojaavat koko verkkoa hyökkäyksiltä toisin kuin henkilökohtainen joka suojaa vain muutamaa konetta. Nämä palomuurit tulevat yleensä kahdessa eri muodossa, ne ovat joko laitepohjaisia tai tulevat ohjelmistopakettina osana asennettua käyttöjärjestelmään eli ohjelmapohjaisina. Esimerkkinä laitepohjaisista palomuureista voidaan pitää esimerkiksi, Cisco ASA ja Junipers Netscreen palomuureja. Ohjelmapohjaisista palomuureista taas esimerkkinä voidaan pitää Linux pohjaisia palomuureja kuten IPtables ja BSD projektin packet filtteriä. (Noonan & Dubrawsky 2006, Firewall Products)

Verkon palomuurit ovat erittäin tärkeitä turvallisen ympäristön ylläpitämisessä ja ne ovat etulinjassa hyökkäyksiä vastaan. Palomuurit ovat vastuussa kaikesta tiedon pääsystä laitteiden välillä, tähän kuuluu verkko, tietokoneet ja palvelimet. Palomuurien rooli on suuresti kasvanut ja osittain vastauksena kattavampien palveluiden kasvuun internetin kautta, kuten multimedia ja salatut yhteydet. Useimmat palomuurit voivat myös suorittaa osoitteenmuunnoksen (NAT), joka mahdollistaa useamman koneen käyttävän samaa IP-osoitetta täten säästäen rajoitettua määrää osoitteita. Palomuuri voi myös huolehtia verkon liikenteestä, antaen erilaisen liikenne prioriteetin eri pal-

veluille. Yksi tällainen palvelu on VoIP eli Voice over IP, joka huolehtii äänen siirrosta IP:n kautta reaaliajassa.(Vacca 2009, 213-214)

3 PALOMUURIEN TYYPIT

Palomuurit voidaan luokitella neljään eri pääluokkaan: Pakettisuodattimet, tilalliset, tilattomat palomuurit ja proxyt. Jokainen palomuuri tarjoaa omanlaisensa suojauksen ja sopii paremmin tietynlaiseen verkkoon. Näiden vanhojen palomuuri tyyppien lisäksi on kehitetty uusi palomuurityyppi Next-Generation Firewall, joka on useasti lyhennetty NGFW.

Näiden eri palomuurityyppien lisäksi on vielä olemassa Unified Threat Management (UTM) mutta nämä eivät varsinaisesti ole vain palomuureja, vaan paremminkin yhdistelmä Anti-virus, palomuuri ja IPS sovelluksia luotuna yhdeksi kokonaisuudeksi. UTM ei ole aina välttämättä paras vaihtoehto sen laajuuden takia mikä voi hidastaa verkkoliikennettä. UTM käyttää useasti myös enemmän aikaa pakettien tarkistukseen, koska paketti tarkistetaan mahdollisesti moneen kertaan monen eri UTM:ään kuuluvan ohjelman kautta ja tämä saattaa aiheuttaa viivytyksiä verkkoliikenteessä. Tämä tarkoittaa sitä, että tietoturvavastaavien on löydettävä jonkunlainen kompromissi käytettävyyden ja suojauksen välillä.(Ohlhorst 2013)

3.1 Pakettisuodattimet

Pakettisuodatin eli englanniksi packet filter, vaikka tässä tapauksessa ei tarkoiteta FreeBSD:ssä toimivaa saman nimistä ohjelmaa. Tämä palomuurityyppi on ehkä kaikkein yksinkertaisin palomuuri, koska se suodattaa toimintaa vain verkko ja kuljetuskerroksissa. Pakettisuodatin toimiikin reitittimen tapaisesti. pakettisuodatin saa datapaketin verkosta ja tutkii paketin, jonka jälkeen paketti joko hylätään tai päästetään läpi palomuurista. Pakettisuodattimen toiminta datapaketin kanssa määräytyy kuljetus ja verkkokerroksien datan ja informaation perusteella. Tämä tarkoittaa sitä,

että pakettisuodatin ottaa huomioon vain IP osoitteet ja porttien numerot tehdessään pass/deny päätöksiä. Tämä myös johtaa siihen, että pakettisuodatin ei tutki paketin sisäistä payloadia eli dataa, koska kaikki informaatio, jota se tarvitsee sijaitsee paketin headerissä, eli otsikossa. Joissain tapauksissa pakettisuodattimet tutkivat myös datan välityskerroksen informaatiota. Pakettisuodattimet eivät ylläpidä tietoa saapuneista tai lähteneistä paketeista.(Vacca 2009, 222)

3.2 Tilalliset palomuurit

Tilalliset palomuurit tekevät samoja operaatiota kuin pakettisuodattimet, mutta toisin kuin pakettisuodattimet, tilalliset palomuurit pitävät niin sanotun yhteyden tai tilan saapuneista paketeista. Tämän lisätyn funktionalisuuden myötä on mahdollista rakentaa palomuurin sääntöjä, jotka mahdollistavat sessiota, joissa kaksi konetta, lähettäjä ja vastaanottaja voivat keskustella keskenään. Tämä on kriittinen osuus pääteohjelmien ja palvelimien toimintaa, koska pääteohjelman ja palvelimen on pystyttävä keskustelemaan keskenään, eli kun paketteja lähetetään yleensä odotetaan myös vastausta kyseiseen lähetykseen. Terminologiassa pakettisuodatin lakkaa olemasta pakettisuodatin ja puhutaan palomuurista kun siihen lisätään tila, mutta tämä on täysin mielipide kysymys.

Esimerkkinä tästä voidaan pitää tilannetta, jossa käyttäjä sijaitsee suojatussa sisäverkossa ja haluaa ottaa yhteyden web palvelimeen, joka sijaitsee internetissä. Pyyntö lähetettäisiin käyttäjältä palvelimeen ja palvelin lähettäisi vastauksen pyydettyine tietoineen käyttäjälle. Pakettisuodattimet tarvitsevat tähän kaksi sääntöä, ensimmäinen sääntö antaisi luvan käyttäjälle lähettää paketteja palvelimelle ja toinen sääntö antaisi käyttäjälle luvan vastaan ottaa paketteja kyseiseltä palvelimelta. Tässä lähestymistavassa on muutama ongelma, koska on vaikea ennalta määritellä mihin web palvelimiin käyttäjän on tarkoitus yhdistää. Tässä tapauksessa olisi tarve lisätä uusi sääntö jokaiseen web palvelimeen mihin käyttäjän ottaa yhteyttä.

Tilallinen palomuuri pystyy jäljittämään yhteyksiä, jotka mahdollistavat saapuneiden datapakettien hyväksymisen niiden lähtöosoitteen perusteella. Kaikki informaatio

joka kulkee näiden kahden kohteen välillä, voidaan pitää yhtenä keskusteluna käyttäjän ja palvelin välillä. Tätä web-palvelin esimerkkiä käyttämällä voidaan helposti esittää miten yksi tilallinen sääntö voidaan tehdä niin, että se hyväksyy kaikki web-pyynnöt ja data paketit suojatusta verkosta. Yksinkertainen tapa lisätä nämä valmiudet on asettaa palomuuriin uusi sääntö, joka antaa luvan palauttaa paketteja, tämä sääntö olisi tietenkin eliminoitava yhteyden loppuessa. Yleensä on vaikea arvioida yhteyden loppumisaikaa, tämän takia ajastimia käytetään usein säännön lopettamiseen. Kaikesta huolimatta tilalliset säännöt ovat olleet merkittävä edistys palomuuri tekniikassa. (Vacca 2009, 222-223)

3.3 Proxyt

Proxyt toimivat hieman eri tavalla muihin palomuureihin verrattuna, mutta voivat tarjota hyvää suojausta verkolle siitä huolimatta. Proxyt toimivat välittäjinä verkko yhteyksille. Tästä voidaan käyttää esimerkkinä vaikka sisäverkossa olevan käyttäjän ottamaa yhteyttä palvelimeen ulkoverkosta. Tämä yhteys pysähtyy palomuuriin ja palomuuri tekee uuden yhteyden palvelimeen. Tämä muutos tapahtuu saumattomasti.

Proxyyn voidaan esimerkiksi sijoittaa palvelimella oleva nettisivu, tämä estää sen että käyttäjät, jotka käyttävät sivua eivät ole oikeastaan palvelimella ollenkaan vaan proxyssä. Tämä estää käyttäjiä tietämästä palvelimen oikeaa osoitetta. Tämä toimii myös toiseen suuntaan ja jos lokia kerätään palvelimelta eikä proxystä kaikki sivulla vierailevat osoitteet näyttävät vain proxyn osoitteen.

Uuden yhteyden seurauksena palomuuri voi tarkastella datapakettien sisältöä IDS:n tapaan. Tämä on tärkeää, koska sovelluksien määrä kasvaa jatkuvasti, kuten myös käytäjien ja ohjelmien, jotka käyttävät standardista poikkeavia porttinumeroita datan siirtoon. (Vacca 2009, 223)

3.4 Next-Genertation Firewall

Next-Generation Firewall on nimensä mukaisesti palomuureista kaikkein viimeisin palomuuri tyyppi. Tämä palomuuri eroaa traditionaalisimmista palomuureista monel-

la tavalla ja ollessaan uusi, siitä on myös erittäin vähän käyttäjien kokemuksia. Tämä palomuuri tunnetaan myös nimellä Application Firewall eli Sovellus palomuuri, NGFW, eroaa muista palomuureista olemalla tietoinen eri sovelluksista ja niiden toiminnasta.

Toisin kuin tilallinen palomuuri, NGFW pystyy tunnistamaan ja estämään sovelluksia tiettyjen sovellusrakenteiden ja toiminnallisuuden perusteella, jotka ovat tyypillisiä tietylle sovellukselle. Tämä suojaus paradigma on tehty sitä varten, että käyttäjät eivät pysty ohittamaan suojausta yhdisteltyjen menetelmien kautta, kuten mappaamalla haitallisia ohjelmia portteihin, jotka tiedetään olevan auki tai käyttämällä anonyymejä proxyjä kuten TOR. Toisin kuin perinteinen palomuuri, joka kontrolloi liikennettä IP osoitteiden ja porttien kautta, NGFW valvoo käyttäjiä ja ohjelmia perinteisen palomuurin tapaan. NGFW perinteisen palomuureille tyypilliseen tapaan käyttää Pass/Deny menetelmää monitoroidessa ohjelmia ja sovelluksia. Nämä säännöstelyt voidaan myös asettaa niin, että ne koskevat vain tiettyjä käyttäjiä, jotka yrittävät käyttää tiettyjä ohjelmia.(Passeri, 2011)

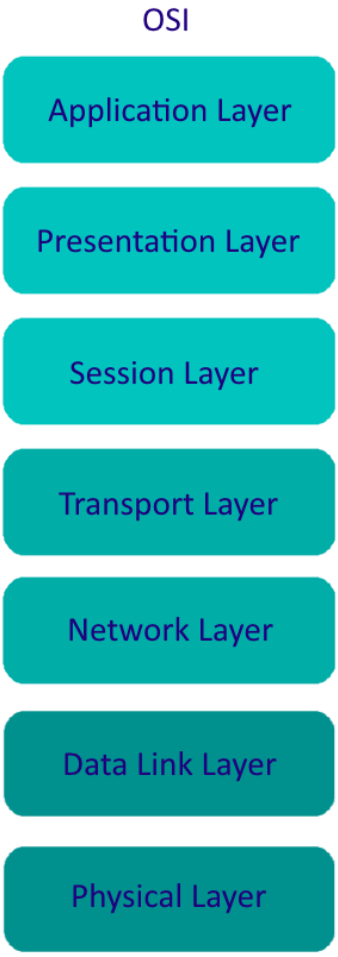
4 TCP/IP DATA-ARKKITEHTUURI JA OSI

Olkoon sitten langallinen tai langaton yhteys, suurin osa datan liikkumisesta IP-protokollan mukaisesti, käy näinä päivinä datapakettien välityksellä, jotka kulkevat useamman verkon kautta. Mutta mikäli näiden verkkojen on kommunikoitava keskenään niiden täytyy molempien käyttää yleistä protokollaa pakettien lähettämiseen ja vastaan ottamiseen. Vaikka monia protokollia on tehty, yksi kaikkein käytetyimmistä on TCP/IP eli Transmission Control Protocol / Internet protocol. Myös geneerinen protokolla malli Open Systems Interconnection eli OSI on erittäin käytetty, kun halutaan selittää protokollien toiminta periaatteita. (Frenzel, 2013)

4.1 OSI

OSI malli eroaa paljon IP/TCP:stä. Ensinnäkin OSI on vain malli, sitä ei ole pakko noudattaa sanasta sanaan, tosin monet protokollat ja systeemit seuraavat sitä erittäin

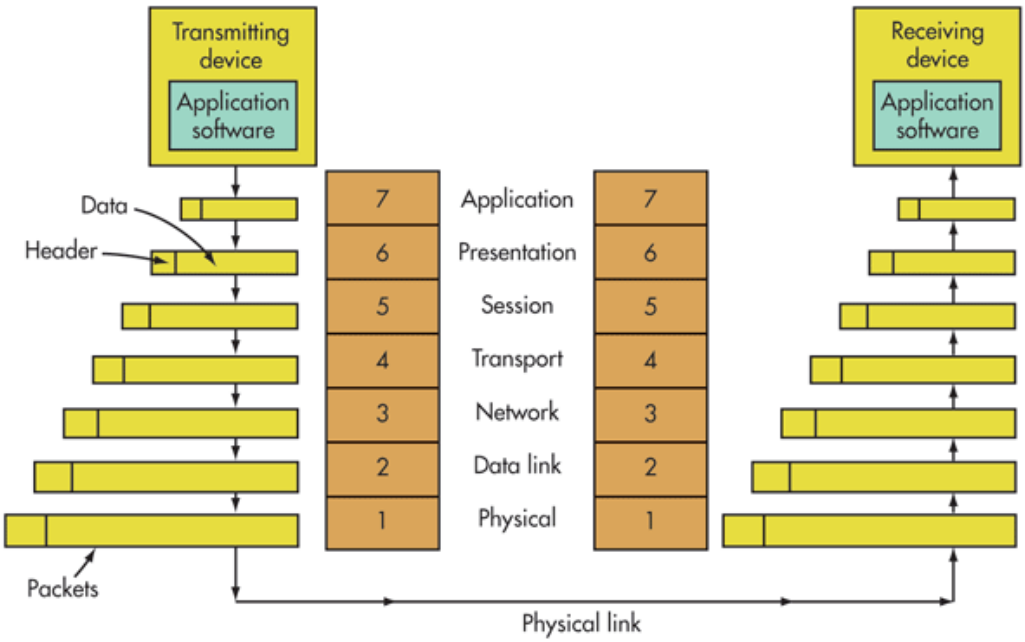
läheisesti. OSI:n hyödyllisyys tulee siitä, että sillä on helppo verrata ja selittää verkkojen eri funktiota. Toiseksi kaikki OSI:n kerrokset eivät ole välttämättä käytössä yksinkertaisemmissa ohjelmistoissa, vain kerrokset yksi, kaksi ja kolme ovat välttämättömät dataliikenteeseen.



Kuva 2: OSI

Datan siirron tapahtuessa jokainen kerros lisää otsikon dataan, joka ohjaa datapakettia. Tätä prosessia kutsutaan kapseloinniksi. Otsikko ja data muodostavat yhdessä paketin, joka taas johdetaan seuraavaan kerrokseen, jossa se saa uuden otsikon, lopuksi kapseloitu paketti on valmis lähetettäväksi ja vastaan otettavaksi. Tietokone, johon paketti saapuu, tekee sitten kapseloinnin käänteisenä eli de-kapseloi paketin. Otsikko auttaa konetta kapseloinnin kääntämisessä. Tämä toistetaan kaikelle datalle, joka kulkee koneiden välillä. (Frenzel, 2013)

Seuraava kuva selventää miten kapselointi toimii käytännössä.



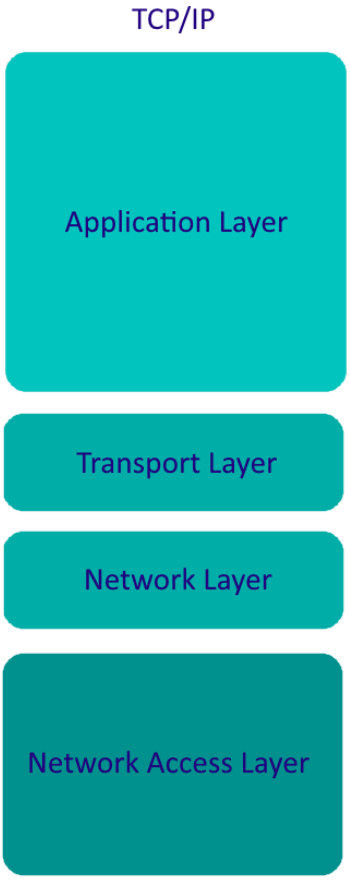
Kuva 3: Kapselointi (Frenzel, 2013)

Seitsemäs kerros eli sovelluskerros, kommunikoi sovelluksen kanssa ja myös varmistaa yhteyden toimivuuden ja resurssien tarpeellisen määrän. Tämä kerros työskentelee myös loppu sovellusten kanssa, kuten DNS, FTP, HTTP ja IMAP4. Kuudes kerros tarkistaa, että data on yhteensopiva tietoliikenteen resurssien kanssa ja varmistaa yhteensopivuuden eri data formaattien kanssa, jotka ovat sovelluskerroksella tai alemmilla kerroksilla. Kuudes kerros hoitaa myös kaiken datan formatoinnin tai koodin muuntamisen, kuten myös datan pakkaamisen ja salaamisen. Viides kerros huolehtii autentikoinnista ja valtuutuksista sekä huolehtii myös yhteydestä laitteiden välillä, kuten yhdistämisen, ylläpitämisen ja lopuksi yhteyden terminoinnin. Lisäksi tämäkin kerros huolehtii datan tarkistamisesta. Neljäs kerros huolehtii QoS eli Quality of Servicestä, joka pitää huolen siitä, että tietoliikenne on priorisoitu ja siitä ,että kaikki data on onnistuneesti siirretty yhtenäisenä. Kerros kolme eli verkkokerros huolehtii paketin reitityksestä. Kerros kaksi eli siirtoyhteyskerros purkaa datapaketin ja ensimmäinen kerros eli fyysinen kerros määrittelee logiikka tason, datan määrän, fyysisen median ja datan toiminnallisuuden muuntamisen, jotka muodostava bittivirran siirrättäessä paketteja toisesta laitteesta toiseen. (Frenzel, 2013)

4.2 TCP/IP

Paras tapa visualisoida ja selittää TCP/IP protokolla sarja, on ajatella sitä OSI mallin tapaisesti kerroksina. Toisin kuin OSI mallissa TCP/IP protokollassa on vain neljä kerrosta, sovellus-, transportaatio-, verkko- ja fyysinenkerros.

Jokainen kerros on vastuussa asetetusta määrästä palveluita sekä valmiuksia, joita pystytään tarjoamaan ylemmille ja alemmille kerroksille. Tämä kerrosmalli mahdollistaa kehittäjien ja insinöörien moduloida kerroksen toiminnallisuutta minimoiden muutosten vaikutuksia muihin kerroksiin. OSI mallin tapaan jokainen kerros lisää otsikon dataan, kun se siirtyy kerroksesta kerrokseen ja ennen kuin data on valmis verkkosiirtoon tai ennen kuin se vastaan otetaan verkosta. Tapa jolla nämä toiminnallisuudet suoritetaan sisäisesti, on piilossa muilta kerroksilta ja niin kauan kuin data ylläpitää edellytettyjä sääntöjä siitä, miten datan tulee kulkea kerrosten läpi, kaikki sisäiset toiminnallisuudet ovat täysin eristettyjä muista kerroksista.



Kuva 4: TCP/IP

Sovelluskerros eli application layer huolehtii sovelluksista ja prosesseista, mukaan lukien prosessit, joita käyttäjä mahdollisesti käyttää kuten verkkoselain, email ja muut verkkotietoiset ohjelmat. On myös mahdollista, että monia muita ohjelmia, jotka ovat koneella, suoritetaan sovelluskerroksessa, koska ne ovat yhteydessä verkkoon. Käyttäjä ei kuitenkaan ole välttämättä edes tietoinen näistä, koska käyttäjällä on erittäin vähän vuorovaikutusta näiden ohjelmien kanssa, esimerkkinä tästä voidaan käyttää vaikka reititys protokollaa.

Transport layer eli kuljetuskerros tai transportaatiokerros on vastuussa datan siirron käsittelystä eri verkon hostien eli isäntien välillä. On olemassa kaksi eri siirto protokollaa TCP/IP sarjassa ja nämä ovat Transmission Control Protocol eli TCP ja User Datagram Protocol eli UDP. TCP on yhteys tai istunto suuntautunut protokolla, joka tarjoaa monia palveluita sovelluksille, kuten esimerkiksi luotettavan pakettien kuittauksen, bufferin hoidon ja virhekäsittelyn Positive Acknowledgement Retransmission eli PAR:n kautta. Toisin kuin TCP, UDP on kevyt, yhteydetön protokolla, joka ei tee toimituksen kuittausta tai muita istunnon palveluita. Kaikki pakolliset sovellusluotettavuudet on oltava sovelluksessa itsessään. TCP:ssa sovelluksen ei tarvitse huolehtia pakettien lähettämisestä. Kumpikin protokolla palvelee tiettyä tarkoitusta ja mahdollistaa mahdollisimman suuren joustavuuden sovellusten kehittäjille.

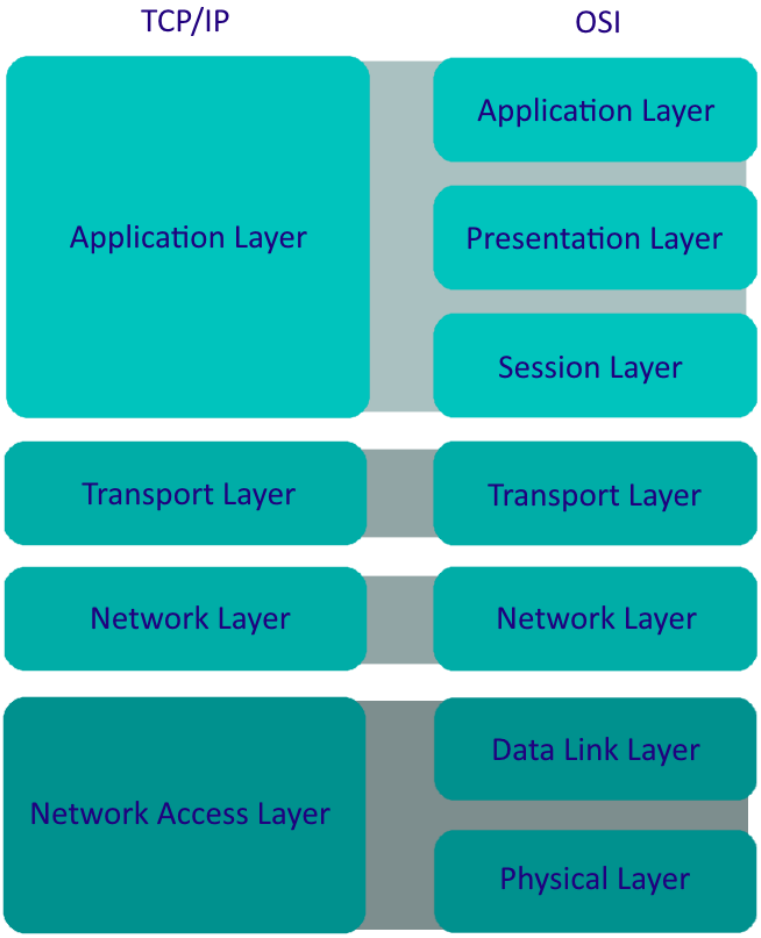
Koneella voi olla monia verkkosovelluksia käynnissä samaan aikaan, joista jokainen on joko rakennettu TCP:n tai UDP:n päälle, ja joissain tapauksissa ohjelma voi käyttää molempia, joten molemmat protokollat käyttävät portteja, joiden avulla määritellään verkkopalveluita ja dataa. Esimerkkinä tästä voidaan käsitellä palvelinta, jossa on verkkopalvelin, tälle tarpeelliset verkkopalvelut on tarjottu TCP portissa 80, Samalla palvelimella on myös sähköpostipalvelut, joka käyttää Simple Main Transfer Protocol eli SMTP protokollaa, joka on normaalisti TCP portissa 25 ja DNS palvelin, joka kuuntelee UDP porttia 53 ja TCP porttia 58. Kuten esimerkistä huomataan portit mahdollistavat monien eri UDP ja TCP palveluiden toiminnan samaan aikaan ilman, että ne häiritsevät toistensa suorituksia. Yleisesti portit ovat suljettuja ja ne pitää avata, jotta niitä voi käyttää. Esimerkiksi Linuxilla pitää määritellä avatut portit iptables ohjelmalla, jos palvelu on esimerkiksi portissa 22, mutta portti on kiinni ohjelma ei saa yhteyttä verkkoon. On myös hyvä huomioida, että esimerkiksi portissa 80 ei ole

pakollisesti verkkopalveluita vaan periaatteessa missä tahansa portissa voi olla mikä tahansa palvelu.

Verkkokerros on pääasiallisesti vastuussa pakettien käsittelystä ja reitityksestä verkon läpi. Internet Protocol eli IP johtaa tätä prosessia TCP/IP perheen sisällä. Yksi erittäin tärkeä osa IP:tä on IP osoitteen käsite. Jokaisella koneella, joka toimii TCP/IP verkkossa on oltava ainakin yksi uniikki IP osoite, johon muut koneet voivat suunnata liikennettä. IP osoite koostuu 32-bittisestä numerosta, joka on yleisesti esitettynä neljänä kokonaislukuna, jotka voivat olla mitä tahansa 0 ja 255 väliltä. Nämä kokonaisluvut ovat eroteltuna pisteillä toisistaan esimerkiksi 192.168.1.254. IP osoitteessa on itse asiassa kaksi tärkeää informaatiosaa: verkkoosoite ja node (host) osoite. Jotta tiedetään mihin verkko osoite loppuu ja mistä node osoite alkaa, käytetään aliverkon peitettä eli subnet maskia, joka osoittaa kuinka monta bittiä IP osoitteesta on verkko osoitteessa, tätä merkitään yleensä kenoviivalla ja numerolla, kuten /24. Mikäli esimerkki IP osoitteella 192.168.1.254 on aliverkon peite /24, tiedetään, että verkon osoite on 24 bittiä eli 192.168.1 ja node osoite on tällöin 254, jos taas aliverkon peite olisi /16 verkko osoite olisi 192.168 ja node olisi 1.254. Aliverkon peite antaa mahdollisuuden verkon suunnittelijoille tehdä eri kokoisia aliverkkoja vaihdellen kahdesta (subnet mask /30) miljooniin eri nodeihin (subnet mask /8) tai mitä tahansa tältä väliltä. Aliverkotus itsessään on laaja ja monimutkainen asia ja sillä on suuri vaikutus osoitteisiin ja reititykseen.(kts. Kuvio 4)

Peruskerros eli Network Access Layer on vastuussa vuorovaikutuksesta fyysisen verkon kanssa. Datan kuljetuksesta riippuen, tässä kerroksessa voi olla erilaisia funktiota. Näihin voi kuulua törmäyksen välttäminen, pakettien ja tietosähkeiden lähettäminen ja vastaanottaminen, perus virheiden tarkistus ja niin edespäin. Peruskerros hoitaa kaikki rajapinnan tiedot ja eristää kaiken fyysisen tiedon ylemmistä kerroksissa.

Kuten OSI mallissakin TCP/IP sarjassa käytetään hyväksi myös kapselointia. Datan kulkiessa alaspäin kerroksissa jokainen kerros lisää osoitteen dataan. Kolme tärkeintä osoitetta ovat IP osoite, TCP osoite ja UDP osoite, nämä osoitteet ovat tärkeitä jotta kerrokset pääsevät käsiksi kapseloituun informaatioon. (Vacca 2009, 128-130)



Kuva 5: TCP/IP ja OSI

5 IPV6 JA ALIVERKOTUS

5.1 IPv6 tietoturva

IPv6 on uusi versio internet protokollasta, joka on suunniteltu seuraajaksi IPv4:lle. Suurimmat muutokset IPv6:ssa on suuresti laajennettu osoiteavaruus, joka on laajennettu 32 bitistä 128 bittiin, koska IP osoitteista alkaa olla pulaa. Toinen muutos on tunnisteiden formaatin yksinkertaistaminen, jotkin IPv4 tunnistekentät poistettu tai tehty vaihtoehtoisiksi paketinhallinnan yksinkertaistamiseksi ja IPv6 tunnisteen kaistanleveyden pienentämiseksi. Kolmantena muutoksena IPv6 asetuksien koodauksen muutos, joka on tehnyt asetuksien muuttamisesta ja niiden lisäämisestä helpompaa.

Neljänneksi on lisätty pakettien merkintä, jolla paketit pystytään tunnistamaan kuuluvaksi tiettyyn liikennevirtaukseen. Viimeiseksi on vielä lisätty autentikointi ja yksityisyys mahdollisuudet, tämä sisältää laajennuksia, jotka tukevat ja mahdollistavat datan yhtenäisyyden ja vaihtoehtoisesti yksityisyyden autentikoinnin. (Deering & Hinden 1998, 1-2)

IPv6 on samantyylinen IPv4:ään verrattuna, mutta niissä on silti eroja. IPv6 on integroituna IPSec. Yleinen käsitys on, että IPv6 on turvallisempi, koska IPSec on pakollinen, tämä ei silti ole ihan täysin totta. Vaikka kaikissa IPv6 hosteissa on IPSec, sen käyttö ei ole pakollista, ja toiseksi jos kaikki kommunikaatio kahden IPv6 hostin välillä on salattua niin, verkko tulee sokeaksi eikä pysty tutkimaan liikenteet sisältöä, täten aiheuttaen turvallisuusriskin. Täten IPSec:n käyttöä suositellaan käytettäväksi vain tilanteissa johon se sopii, kuten VPN yhteyksissä.

IPv6 on myös käytössä NDP:ssa (Neighbour discovery protocol), joka toimii OSI mallin toisessa kerroksessa. Se määrittelee automaattisesti nodeja ja niiden osoitteita, mutta siinä on samoja ongelmia kuin IPv4 ARP:ssa ja DHCP:ssä eli IP spoofing. Tämä tarkoittaa sitä, että hyökkääjän IP osoite näkyy erilaisena kuin se oikeasti on, tämä tarkoittaa sitä, että kaikki hyökkäysyritykset esimerkiksi lokitiedostossa eivät paljasta hyökkääjän oikeaa osoitetta. Hyökkääjä voi myös käyttää tätä hyödykseen, mikäli hän haluaa päästä sisäverkkoon, esittämällä konetta, joka on oikeasti sisäverkossa. Näitä hyökkäyksiä voidaan välttää määrittämällä IP väärennykseen (Spoofing) liittyviä sääntöjä. Koska IP osoitteiden väärennys on yhtä helppoa kuin IPv4:ssa on käytössä samat keinot niiden estämiseen, myös lievennys tekniikat ovat samanlaisia eli estetään väärät lähteet ja osoitteet.

Reititys protokollat, joita IPv6 käyttää, ovat joko identtisiä IPv4 kanssa tai paranneltuja versiota IPv4 protokollista. Tähän kuuluu autentikointi mekanismi, joka pitää konfiguroida, että voidaan estää reitin kaappaus. Lähdereititys on pois käytöstä automaattisesti IPv6:ssa. Hyökkääjä ei voi käyttää lähdereititystä suojauksen läpipääsyyn tai DDoS hyökkäykseen. Kaikki sovelluskerroksen haavoittuvuudet kuten SQL injection tai cross-site scripting ovat edelleen ongelma, koska nämä eivät oikeastaan ole riippuvaisia verkkokerroksesta ja näiden ongelmien kanssa toimitaan samalla tavalla, kuin jos hyökkäys kohdistuisi IPv4:een. Kuten IPv4 myös IPv6 on alt-

tiina DDoS (Distributed Denial-of-Service) hyökkäykselle ja tähän ei ole vielä keksitty tapaa, jolla suojautua kyseistä hyökkäystä vastaan.

Vaikka IPv6:lla on paljon samanlaisuuksia ja samoja ongelmia kuin IPv4:ssa, on IPv6 silti monella tavalla myös erilainen. Osoitteet ovat paljon pidempiä, suuremman osoitetilan takia. Vaikka luulisi, että suurempi osoite tekisi vaikeammaksi hyökkääjälle löytää uhreja, tämä ei ole totta. IPv6 ei myöskään suojaa hyökkäyksiltä, jotka tulevat verkkokerroksen ulkopuolella kuten virukset ja sähköpostimadot. Toisaalta verkkokerroksessa lisääntyvät madot eivät toimi IPv6:ssa ilman muutoksia, koska IPv6 käyttää eri osoiteskannausta kuin IPv4. IPv4:lla on tietokantoja esimerkiksi sähköpostien yleisestä roskapostista, tätä ei IPv6:lla vielä ole, koska IPv6:n liikenne on vielä vähäistä IPv4:ään verrattuna. IPv6:lla on myös mahdollisuus käyttää yksityisyys laajennusta. Tämä laajennus muuntaa hostin osoitetta ajoittain, joten osoitetta ei pystytä jäljittämään. Tämä toimii hyvin yksityisellä käyttäjällä, mutta organisaation hosteille, joita tietoturva vastaavien ja verkkovastaavien on pakko pystyä jäljittämään, tämä ei ole niinkään hyödyllinen.(Cisco 2011. 1-3)

5.2 Aliverkotus

Aliverkot englanniksi subnets, lyhenne sanasta sub-networks, ovat pienempiä verkkoja isojen verkkojen sisällä. Aliverkko, jossa ei ole enempää osia, pidetään broadcast-osoitteena, mikä tarkoittaa, että kyseinen lähiverkko on kiinni yhdessä ethernet kytkimessä. Broadcast-osoite on tärkeä, koska se toimii paikkana verkossa, jossa laitteet voivat keskustella toistensa kanssa käyttäen MAC osoitteita. MAC osoitteen kommunikaatio on rajoitettu pienempään verkkoon, koska MAC osoitteet käyttävät ARP lähetyksiä löytääkseen määränpäänsä, ja lähetyksiä voidaan skaalata pienemmiksi vain rajallisesti, ennekuin niiden liikenne kaataa koko verkon. Tästä syystä yleisin pieni aliverkko on 8-bittinen. MAC osoitteet määräytyvät koneiden verkko adapterien perusteella. Tämän takia MAC osoitteet tunnetaan myös joskus nimellä "hardware address" tai "physical address". Jokainen MAC osoite on uniikki ja muodostuu kahdestatoista heksadesimaali numerosta.

Aliverkoilla on aloitus ja lopetus osoite, aloitus osoite on aina parillinen ja lopetus osoite on aina pariton luku. Aloitus osoite on network ID ja lopetus osoite on broadcast ID. Näitä osoitteita ei saa käyttää normaalisti, koska niillä on erikoistarkoitus. Network ID eli verkon ID on virallinen nimi aliverkolle ja lopetus osoite on broadcast osoite, jota kaikki laitteet aliverkossa kuuntelevat. Jos haluaa viitata aliverkkoon, tulisi viitata sen network ID:n ja subnet maskiin, joka määrittää aliverkon laajuuden. Mikäli halutaan lähettää dataa kaikille aliverkossa, esimerkiksi multicast, se lähetetään broadcast ID:n.

Subnet mask, eli aliverkon maski täyttää tärkeää roolia aliverkotuksessa. Aliverkon maski on suuressa roolissa aliverkon koon määrittelemisessä. Kun työskennellään aliverkkojen kanssa, kahdeksan lukua toistuvat jatkuvasti ja ne olisi hyvä muistaa, nämä luvut ovat: 255, 254, 252, 248, 240, 224, 192 ja 128.

Subnet mask quick reference							
Host Bit length	math	Max hosts	Subnet mask	Mask octet	Binary mask	Mask length	Subnet length
0	2^0=	1	255.255.255.255	4	11111111	32	0
1	2^1=	2	255.255.255.254	4	11111110	31	1
2	2^2=	4	255.255.255.252	4	11111100	30	2
3	2^3=	8	255.255.255.248	4	11111000	29	3
4	2^4=	16	255.255.255.240	4	11110000	28	4
5	2^5=	32	255.255.255.224	4	11100000	27	5
6	2^6=	64	255.255.255.192	4	11000000	26	6
7	2^7=	128	255.255.255.128	4	10000000	25	7
8	2^8=	256	255.255.255.0	3	11111111	24	8
9	2^9=	512	255.255.254.0	3	11111110	23	9
10	2^10=	1024	255.255.252.0	3	11111100	22	10
11	2^11=	2048	255.255.248.0	3	11111000	21	11
12	2^12=	4096	255.255.240.0	3	11110000	20	12
13	2^13=	8192	255.255.224.0	3	11100000	19	13
14	2^14=	16384	255.255.192.0	3	11000000	18	14
15	2^15=	32768	255.255.128.0	3	10000000	17	15
16	2^16=	65536	255.255.0.0	2	11111111	16	16
17	2^17=	131072	255.254.0.0	2	11111110	15	17
18	2^18=	262144	255.252.0.0	2	11111100	14	18
19	2^19=	524288	255.248.0.0	2	11111000	13	19
20	2^20=	1048576	255.240.0.0	2	11110000	12	20
21	2^21=	2097152	255.224.0.0	2	11100000	11	21
22	2^22=	4194304	255.192.0.0	2	11000000	10	22
23	2^23=	8388608	255.128.0.0	2	10000000	9	23
24	2^24=	16777216	255.0.0.0	1	11111111	8	24

Kuva 6: Subnet mask quick reference (Ou, 2006)

Kuvasta käy ilmi, hostien määrä, ja subnetin ja subnetin maskin pituus ja maskin binääri. Tästä kuvasta on helppo nähdä miten subtnet maskin binääri luvussa nollat menevät oikealta vasemmalle. Nollien määrä on aina sama kuin aliverkon pituus. Kuvasta näkyy vain oktettin kiinnostava osa, kokonaisuus esimerkiksi 11 bitiä pitkällä aliverkolla olevalla maskilla näyttäisi tältä:
11111111.11111111.11111000.00000000. Tämä aliverkon maski olisi käännettynä base-256 muotoon 255.255.248.0.

Aliverkon maski ei ainoastaan määrittele aliverkon kokoa, vaan myös auttaa paikantamaan aliverkon lopetus pisteet, mikäli tiedossa on aliverkossa oleva IP osoite. Aliverkon maskia kutsutaan maskiksi, koska se piilottaa hostin bitit ja jättää ainoastaan network ID:n, josta aliverkko alkaa näkyville. Mikäli saadaan selville aliverkon aloitus osoite ja kuinka suuri aliverkko on, sen loppu eli broadcast ID pystytään päättelemään. (Ou, 2006)

6 VERKON RAKENNE

6.1 Lokitiedostot

Riippumatta siitä minkälainen palomuuri tai suojaus on käytössä, lokit ovat kriittinen osa tietoturvaa, tämän takia olisi tärkeää, että palomuuri tai mikä tahansa muu suojaus ja virusturva pitävät tarkkaa lokia tapahtumista. Vaikka lokit pitäisi käydä useasti läpi, lokien koko on yleensä niin suuri ja niissä on niin paljon tietoa, että lokin läpikäyminen kestäisi aivan liian pitkään. Tämän takia aina, kun uusi palomuuri otetaan käyttöön sille määritellään säännöt, joiden mukaan palomuuri tallentaa sisäisen ja ulkoisen liikenteen lokia.

Lokeissa itsessään on paljon tärkeää tietoa palomuurin tapahtumista. Tärkeimpiä ovat tietenkin palomuurin tiputtamat IP osoitteet ja paketit, koska nämä ovat luvattomia yhteysyrityksiä. Hyökkääjät käyttävät yleensä probing tyylistä taktiikkaa etsiessään haavoittuvuuksia verkosta, eli kun lokissa on sama IP osoite, yritetään päästä

sisään monesta eri paikasta on todennäköistä, että joku yrittää löytää haavoittuvuuksia verkossa. Kyseinen IP olisi siinä tapauksessa hyvä kieltää kokonaan. On myös hyvä kiinnittää huomiota mihin yritetään sisään palomuurissa. Toinen tärkeä asia mihin tulisi kiinnittää huomiota on epäonnistuneet sisäänkirjautumisyritykset. On hyvä tietää jos joku yrittää päästä sisään kriittisiin salasanalla suojattuihin systeemeihin. Kolmantena tärkeänä asiana on lähtevä liikenne sisäisistä palvelimista. On hyvä tietää miten sisäinen palvelin toimii normaalisti, jotta pystytään helpommin huomaamaan, kun palvelin ei toimi kunnolla, tai jos joku on päässyt muuttamaan asetuksia. Viimeisenä vielä lähdereititetyt paketit, lähdereititys voi olla merkki siitä että joku yrittää päästä sisään sisäiseen verkkoon, koska monilla verkoilla on osoite jonne ei pääse internetistä (10.x.x.x). Lähdereititettyjä paketteja voidaan käyttää pääsyn saamiseen koneelle, jossa on yksityinen osoite, koska verkossa voi olla kone, jolla on pääsy yksityiselle osoitetaajuudelle.(Willard 2002, 2-4)

Yksi ratkaisu lokien pitämiseen on syslog palvelinmen käyttäminen. Unix palvelimet ja monet palomuurit, printterit ja jopa jotkin web palvelimet kuten apache pystyvät tuottamaan syslog dataa. Windows pohjaiset palvelimet taas eivät normaalisti tue syslogia. Syslog on hyvä tapa kerätä lokitiedot yhteen paikkaan monesta lähteetstä. Tämän tekee mahdolliseksi muutama syslogin komponentti. Syslog listener, ensimmäinen komponentti tekee mahdolliseksi viestien vastaan ottamisen verkon kautta. Tämä kuuntelu prosessi kerää dataa UDP portista 514, koska UDP viestit eivät ole sadan prosentin varmuudella tunnistettu tai taattu saapuvan, tämän takia jotkin laitteet käyttävät TCP porttia 1468 tiedon saapumisen varmistamiseksi. Seuraava tärkeä osa syslogia on tietokanta. Suuret verkot generoivat suuria määriä syslog dataa ja syslog serverin tietokanta mahdollistaa datan tallentamisen ja sen nopean palauttamisen tarpeen tullen. Datan suodatuksen ja järjestämisen taas hoitavat suodatus- ja hallintaohjelmat. Suodatus mahdollistaa tärkeän datan erottamisen muusta datasta ja hallinta mahdollistaa erillaisten syslog varoituksien saamisen, mikä helpottaa ongelman paikantamista.

Vaikka syslogissa on paljon tarpeellisia ominaisuuksia, ei se silti ole täysin ilman ongelmia. Syslog protokolla ei määrittele standardia sille minkälaisena rakenteella viesti saapuu perille, jokin osa datasta on luettavissa, mutta kaikki data ei saavu ihmiselle ymmärrettävässä muodossa. Toinen ongelma on taas edellä mainittu paketti-

en menetys UDP porttien käytön takia, koska UDP on yhteydetön eikä sillä ole sisään rakennettua varmennusta, se saattaa menettää dataa siirron aikana toisin kuin TCP. Suljetussa verkossa tämä ei tosin ole ongelma. Lopuksi vielä syslogissa on muutamia tietoturvaan liityviä riskejä. Syslog viesteissä ei ole autentikointia, joten jonkin koneen on mahdollista imitoida toista konetta ja lähettää virheellisiä lokeja. Syslog on myös haavoittuvainen niin kutsutulle "Replay Attack" hyökkäykselle, joka voi joko viivyttää tai toistaa data siirtoa. (Leskiw, A)

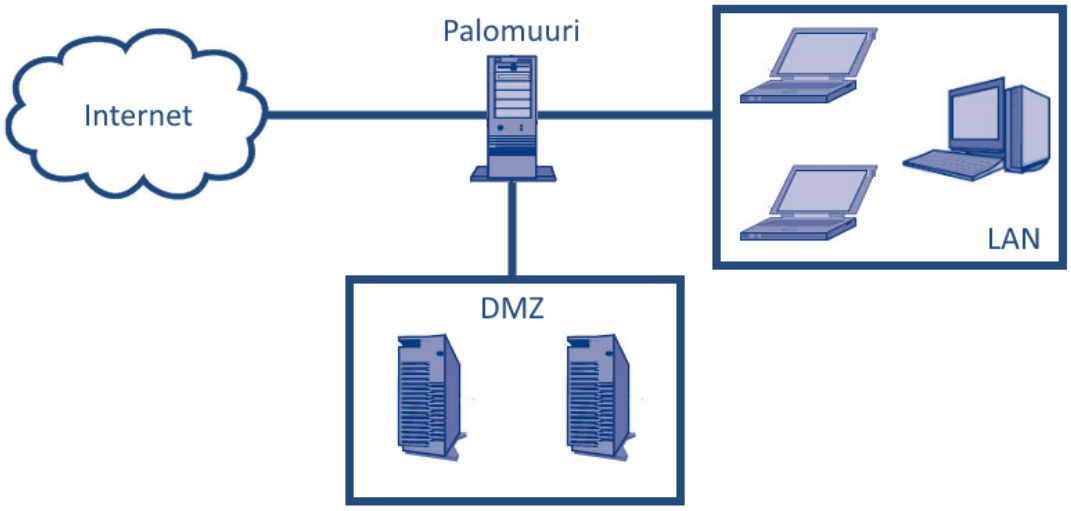
Lokien yhteydessä on myös noudatettava tiettyjä lainsäädäntöjä. Lainsäädäntö asettaa lokien käsittelylle tietyt vaatimukset, jotka poikkeavat oman hallintoalan toiminnan erityislakien sisällöstä ja vaatimuksista riippuen. Muutamia tärkeitä lakeja, jotka asettavat vaatimuksia ovat esimerkiksi henkilötietolaki, julkisuuslaki ja työelämän tietosuojalaki. Erityisesti silloin, kun lokeihin tallentuu henkilökohtaisia tietoja tai tunnistamistietoja, lainsäädäntö asettaa rajoituksia lokien käsittelyyn. Tämän takia lokien käsittelyä suunniteltaessa olisi hyvä kiinnittää huomiota eri lokeihin tallentuviin tiedotoihin. Jos loki sisältää henkilöä koskevaa tunnistettavaa tietoa lokista muodostuu henkilörekisteri, jonka seurauksena tulee huomioida kaikki henkilörekisteriä koskevat vaatimukset ja määrittelyt. (Lokiohje 2009, 20-21)

6.2 Demilitarisoidut vyöhykkeet

Demilitarisoitu vyöhyke eli demilitarized zone (DMZ) johtaa juurensa armeijan käyttämästä samasta sanasta, jolla tarkoitetaan aluetta, jossa sotilas operaatiota ei saa suorittaa. Demilitarisoidulla vyöhykkeellä tarkoitetaan tietotekniikassa liikennettä, joka erottaa sisäisen verkon ja ulkoisen verkon toisistaan.
Toisin kuin termin geopoliittinen vastine, demilitarisoitu vyöhyke ei ole oikeasti neutraali alue, vaan on kokonaan yhtiön hallittavissa. On silti hyvä muistaa, että se on myös eristetty muusta verkosta ja se käyttää IP osoitteita eri network ID:stä. (Shinder, 2005)

Mikäli oletetaan, että palomuurilla on kolme eri yhteyttä, yksi ulkoiseen verkkoon, yksi palvelinverkkoon ja yksi sisäverkkoon ja jokaisella näistä on yhteys palomuuriin. Jokaiselle yhteydelle voidaan asettaa omat säännöt ja toteuttaa eri vahvuiset suo-

jaukset jokaiseen yhteyteen. Yhteyttä web-palvelimeen kutsutaan DMZ:ksi ja se estää pääsyn ulkoisesta verkosta suoraan sisäverkkoon. (Vacca 226, 2009)



Kuvio 7: Demilitarisoitu vyöhyke

Tällä saavutetaan se että koneet, kuten web-palvelin ja muut julkiset palvelimet pystyvät olemaan suorassa yhteydessä internetiin DMZ:n kautta sen sijasta, että ne olisivat sisäisessä verkossa. Koneet DMZ:n sisällä ovat suojattu palomuurilla, mutta niihin on paljon helpompi murtautua kuin sisäverkkoon, koska ne ovat suorassa yhteydessä ulkoisiin koneisiin. Mutta vaikka hyökkääjä pääsee sisään koneeseen, joka on DMZ:n sisällä, se ei vaaranna itse sisäverkkoa, koska ne ovat kokonaan eri verkoissa.

Jossain tapauksissa koneen on oltava yhteydessä internetiin, koska muuten se ei pysty suorittamaan annettua tehtävää, tästä esimerkkinä julkinen web-palvelin, mutta tulisi aina ottaa huomioon, ettei sisäverkko olisi suorassa yhteydessä ulkoverkkoon, tämän takia DMZ vähentää suuresti hyökkäyksiä sisäverkkoon. Kaikesta tästä huolimatta olisi hyvä pitää DMZ:ssä olevat koneet minimissä, koska ne ovat aina suuremmassa vaarassa tulla murretuiksi kuin sisäverkon koneet.(Shinder, 2005)

7 FREEBSD

FreeBSD, joskus tyylitelty FreeBSD Projektiksi, on unix tyylinen käyttöjärjestelmä jota käytetään usein palomuureissa, palvelimissa ja reitittimissä. FreeBSD projektin päämäärä on pitää käyttöjärjestelmä vapaana kaikista maksullisista ohjelmista ja lisensseistä, täten se sopii täydellisesti esimerkiksi pienyrityksille jotka tarvitsevat palomuuria tai palvelinta. FreeBSD sopii eritoten palomuuriksi ja palvelimeksi, koska siinä on valmiiksi rakennettu TCP/IP verkotus, tämä tekee siitä ideaalisen hoitamaan verkon toimintaa. FreeBSD:n lähdekoodi on myös avoin monen muun Unix tyylisen distribuution tapaan. FreeBSD on myös melko yksinkertainen käyttöjärjestelmä ja suurin osa sen komennoista on lyhyitä ja yksinkertaisia, tehden esimerkiksi palomuurin tai palvelimen ylläpidosta helpompaa. FreeBSD tukee omien ohjelmiensa lisäksi kolmannen osapuolen ohjelmia. (Gavin, 2014, 1-8)

Vaikka FreeBSD:llä on paljon hyviä puolia, se on hyvin tarkka tietokoneen laitteistosta. FreeBSD ei tue läheskään samaa määrää laitteita kuin esimerkiksi Windows tai Ubuntu. Tämä tarkoittaa sitä, että käyttäjän olisi hyvä tarkistaa koneen laitteisto ennen FreeBSD:n asentamista esimerkiksi palomuuriksi. FreeBSD sopii myös hyvin palvelimeksi muiden ilmaisten distrojen tapaan.

Vaikka Linux ja BSD ovat molemmat Unix-pohjasia käyttöjärjestelmiä ja niiden komennot ovat jokseenkin samoja, on niissä silti joitakin eroja, jotka olisi hyvä muistaa. Linux ja BSD käyttävät eri nimitys sääntöjä verkkoliitännöille. Linux nimeää kaikki koneen verkot järjestyksessä eth0, eth1, jne. Joissakin tapauksissa langattomat verkot ovat muodossa wlan0, wlan1, jne. BSD taas nimeää liitännät ajurin mukaan esimerkiksi vanhemmat 3Com kortit näkyvät muodossa ep0, ep1, jne. kun taas Intel Gigabit kortit näkyvät muodossa em0, em1, jne. Tämä systeemi on melko looginen ja tekee dokumentoinnista helpompaa, kun esimerkiksi koneen kernel ilmoittaa, että sinulla on liitäntä nimellä em0 tarvitsee vain kirjoittaa man em komento nähdäkseen mitä siirto nopeuksia se tukee, mitä ominaisuuksia sillä on ja tarvitseeko se esimerkiksi firmwarea. Toinen huomattava asia on eri komentojen lisäksi se, että BSD asetustiedostot ovat eri paikassa linuxiin verratessa. (Hansteen 2011, 6)

7.1 Packet Filter

FreeBSD:n mukana tulee kolme erilaista palomuuria, Packter Filter lyhyesti PF, IPFW tilallinen palomuuri ja IPFILTER, joka tutkii liikennettä porttien kautta. PF on ollut FreeBSD projektissa integroituna ohjelmana jo versiosta 5.3 lähtien, joka julkaistiin vuona 2004 tämä tarkoittaa, että tätä kirjoittaessa PF on ollut FreeBSD:ssä perus sovelluksena jo yli 10 vuotta. PF itsessään on täysin varusteltu palomuuri, joka sisältää kaikki tilalliselle palomuurille tyypilliset piirteet. (Gavin 2014, 850-852)

Packet Filterin pääasiallinen tarkoitus on nimensä mukaisesti suodattaa paketteja Packet Filterin rakenteen määrittämällä tavalla. PF käy läpi jokaisen paketin, joka kulkee sen läpi ja tarkistaa, että se on rakenteessa olevien suodatus sääntöjen mukainen. PF on myös vastuussa pakettien pudottamisesta tai läpipääsystä. Läpäisyn ja pudottamisen sijasta PF:lle voidaan myös määritellä muita sääntöjä, esimerkiksi antaa komento tietynlaisen paketin antamiselle ulkoiselle ohjelmalle tutkittavaksi.

PF:lle voidaan määrittää monia eri sääntöjä pakettien suodatukseen liittyen. Verkon liikenteen valvontaan voidaan asettaa melkein mitä tahansa sääntöjä, jotka liittyvät verkkoliikenteen tai tulevan paketin ominaisuuksiin. PF voi suodattaa paketteja muun muassa: verkko osoitteen perheen, lähtö- tai saapumisosoitteen, rajapinnan, portin ja paketin suunnan mukaan. Näiden kriteerien perusteella PF toimii käyttäjän sille asettamalla tavalla. PF pystyy pitämään ei tahdotun liikenteen poissa verkosta, mutta tämän lisäksi se pystyy myös hallitsemaan verkon sisäistä liikennettä. Vaikka molemmat funktiot ovat palomuurissa tärkeitä nämä eivät silti ole ainoita asioita, joita palomuuri pystyy tekemään. Suodatusta voidaan myös käyttää pakettien ohjaamiseen tiettyihin hosteihin, asettamaan verkkoliikenteeseen jonoja ja hallitsemaan muutenkin verkon liikennettä. Myös eri ohjelmille voidaan asettaa niiden omia sääntöjä mikäli halutaan rajata tiettyjen ohjelmien toimintaa. Kaikki tämä prosessointi tapahtuu verkkotasolla pakettien ja yhteyksien ominaisuuksien perusteella. PF toimii käyttöjärjestelmän kernelin tasolta koska tämä nopeuttaa sen toimintaa huomattavasti, sen sijasta että se toimisi käyttäjän tilasta. (Hansteen 2011, 3)

8 FIREWALL BUILDER

Firewall Builder on ilmainen, avoimen lähdekoodin ohjelma, joka toimii graafisena käyttöliittymänä moniin eri palomuuriehin. Firewall Builder yksinkertaistaa palomuurien hallintaa palomuureissa, jotka ovat muuten hieman hankalia tai hitaita hallita. Firewall Builder tukee muun muassa Netfilter/iptables, ipfw, PF ja Cisco PIX palomuureja. Tämä helpottaa varsinkin PF:n ja muiden unix pohjaisten palomuurien hallintaa, joissa ei ole graafista käyttöliittymää, koska Firewall Builder tarjoaa mahdollisuuden käytää sitä palomuurien hallintaan.

Firewall Builderilla pystyy hallitsemaan palomuurin suojaus sääntöjä pienemmällä vaivalla ja helpommin kuin esimerkiksi packet filterissä, jossa ainoa käyttöliittymä on vain konsoli. Ohjelma eroaa myös monella tapaa siitä miten palomuurin komentoja ja parametrejä käytetään, koska ohjelma ei ole tekstipohjainen, komennot ja parametrit toteutetaan graafisen käyttöliittymän kautta eri objekteilla. Firewall Builderissa on myös sisään rakennettu kirjasto, jossa on valmiita palomuuri sääntöjä. Valmiin tuotoksen, joka on tehty Firewall Builderilla voi sen jälkeen compilata haluttuun palomuuri muotoon ja sen jälkeen vain siirtää halutulle koneelle jonne palomuuri on saatava.

Firewall Builder tekee monen palomuurin kokoonpanon hallinnan ja syntaksin muuntamisen helpommaksi. Firewall Builder pystyy kirjoittamaan muun muassa iptables shellscriptejä, pf.conf tiedostoja ja Ciscon reitittimen pääsyluetteloita. Kokoonpanojen kirjoittamisen jälkeen tiedoston voi suoraan kopioda omiin scripteihin tai antaa Firewall Builderin itse kopioida teksti alkuperäisiin scripteihin. Firewall builder mahdollistaa myös samojen osoite- ja palveluasetuksien käytön monissa palomuureissa. Tämä helpottaa palomuurin asetuksien kopiointia toisiin palomuureihin mikäli organisaatiossa on käytössä useita palomuureja, jotka tarvitsevat samoja sääntöjä.

Useiden eri palomuurien kokoonpanot käyttävät eri syntakseja, jotka voivat olla hankalia muistaa, varsinkin jos työskentelee monen eri tyyppisen palomuurin kanssa. Firewall Builder tekee monen eri palomuurin kokoonpanon kirjoittamisesta helpompaa. Firewall Builder tukee iptables, pf ja Cisco reitittimien asetuksien syntakseja.

Vaikka Firewall Builder tukee useita eri palomuureja sen graafinen käyttöliittymä näyttää jokaisessa palomuurissa samalta. Tämä on hyvä muistaa varsinkin jos on tottunut täysin pelkistettyyn työympäristöön. Graafisen käyttöliittymän takia ohjelmaan on ensin tutustuttava, koska se pyörittää samoja configuraatio tiedostoja niiden graafinen ulkomuoto on erillainen. Toisaalta Firewall Builder tukee monia eri palomuureja ja antaa monia eri työkaluja, jotka helpottavat palomuurien hallintaa ja muokkausta. Firewall builder tukee myös mahdollisuutta muuntaa IPv4 ympäristöön tehdyn palomuurin toimivaksi IPv6 ympäristössä. Firewall Builder mahdollistaa myös uusien IPv6 sääntöjen lisäämisen jo valmiina olevaan palomuuriin, tekemällä IPv6 objekteja ja lisäämällä vain IPv4 säännöt näihin kyseisiin objekteihin.

Firewall Builder pystyy myös kehittämään kokoonpanoja eri laitteisiin kuten Linksys, D-link ja muihin reitittimiin, jotka käyttävät DD-WRT ja OpenWRT tekniikoita. Kokoonpanoja voi myös kehittää Linuxin tai FreeBSD koneilla toimiviin palomuureihin kuten myös normaalien Windows koneiden ja Ciscon reitittimien ohjelmistoihin. (Firewall Builder 5 User guide 2011, 1-3)

9 TOTEUTUS

Toteutuksessa tutkin miten Firewall Builderia voisi soveltaa jo valmiiksi tehdyn PF asetuksien muokkaukseen ja korjaamiseen. Kyseessä oleva PF configuration tiedosto on melko vanha ja sisältää monia sääntöjä, jotka eivät ole enää käytössä. Minulle on annettu osa alkuperäistä PF configuration tiedostoa, jota käytän Firewall Builderin kanssa testaamiseen, mutta koska tiedosto ei ole kokonainen sen toiminnasta minulla ei ole varmuuksia, koska jotkin säännöistä voivat vaikuttaa epäsuorasti palomuurin toimintaa, on voinut jäädä kokonaan pois configuration tiedostosta.

Lähtökohtaisena ongelmana palomuurissa oli se, että sillä on vuosien aikana ollut useita ylläpitäjiä ja siihen on kerääntynyt suuri määrä sääntöjä näiden vuosien aikana. Säännöistä toiset ovat vuosien saatossa jääneet täysin käyttämättömiksi, mikä aiheuttaa hämmennystä palomuurin sääntöjä katsoessa ja uusia sääntöjä lisätessä. Tar-

koituksena oli pääasiallisesti löytää tapa, jolla saataisiin palomuuri helposti siivottua ylimääräisistä säännöistä ja saada palomuuri helpommin hallittavaksi.

Alkuperäinen päämäärä oli saada jonkinlainen graafinen käyttöliittymä packet filtteriin. Packet Filterille on ohjelma nimeltä PFsense, joka olisi periaatteessa täydellinen vaihtoehto tarvittavalle graafiselle käyttöliittymälle. Ongelmana PFsensessä on kuitenkin sen rajoittuneet sääntöjen määrittelyt. PFsense ei pysty asettamaan sääntöjä sisäverkossa oleville eri kohteille vaan pystyy ainoastaan määrittelemään ne sisään ja ulos menevälle liikenteelle, tämä rajoittaa sen käyttöä laajassa käyttöympäristössä. PFsense olisi toisaalta erittäin hyvä vaihtoehto mikäli haluaa rakentaa palomuurin Packet Filterillä esimerkiksi omalle koti koneelle. Kun PFsense ei sopinut tähän tarkoitukseen edellä mainituista syistä, piti löytää toinen vaihtoehto ja päädyttiin Firewall Builderiin, jolla pystytään rakentamaan useita eri palomuureja.

Toteutusvaihetta tehdessä ensimmäinen haaste oli opetella alkuperäisen palomuurin eli Packet Filterin syntaksi, koska palomuurin alkuperäinen conf tiedosto on PF syntaksin mukainen, oli tärkeä ymmärtää sen toiminta ja säännöt. Syntaksin opettelu auttoi myös ymmärtämään suuren osan asetuksista ja siitä miten PF conf kuuluu kirjoittaa. Alkuperäisen configuration tiedoston lukeminen auttoi myös ymmärtämään hieman kuinka suuresta verkosta on kyse ja kuinka monia eri osoitteita se oikein piti sisällään. Packet Filterin Syntaksin läpikäynnin jälkeen piti opetella Firewall Builderin käyttöä ja ymmärtää sen rajoitukset, että tietäisin, ettei se PFsensen tapaan olisi liian rajoittunut.

Saatuani configuraatio tiedoston Opetekin kontaktiltani, Firewall Builder luki kyllä Packet Filtterin configuration tiedoston, mutta ennen kuin se antoi koota tiedoston se valitti monesta eri säännöstä. Kokoaminen eli compiling on tapahtuma jossa ohjelman koodi muunnetaan toisella alustalla sopivaan muotoon. FWBuilder on erittäin tarkka kaikista säännöistä eikä se anna koota palomuurin tiedostoa mikäli siinä on yhtään niin kutsuttua "shadow rule" sääntöä. Shadow rule tarkoittaa sellaista sääntöä, joka ei koskaan tule käytäntöön, koska se jää jonkin toisen säännön varjoon, eli siihen ei koskaan päästä. Tämä tapahtuu muun muassa jo pelkästään aloitus säännössä "block drop all", joka kieltää kaiken liikenteen. FWbuilder ymmärtää tämän säännön

sellaisena, että mikään ei ikinä kulje verkon läpi. En tiedä johtuiko shadow rule tapaus vain siitä, että minulla on vain osa palomuuria vai onko jokin muu ongelmana.

Saatuani vihdoin configuration tiedoston koottua FWBuilderilla, sen jälkeen, kun poistin kaikki kokoojan säännöistä ilmoittavat virheet, tuli uusi ongelma. Kokooja nimittäin ei tallentanut mitään alkuperäisen tiedoston kommentteja. Tämä on hankalaa siksi, koska kommentit olivat erittäin tärkeitä ja auttoivat minua ja varmasti muitakin ymmärtämään tiedostoa paljon paremmin. Kommenttien puuttuminen ei ollut ainoa ongelma. FWBuilder myös uudelleen nimesi kaikki jo valmiit taulut configuration tiedostosta, joka hämmensi minua vielä enemmän. Tämä sai aikaan sen että esimerkiksi alkuperäisessä tiedostossa oleva

table <fsecure> { 172.16.16.50, 172.16.16.60 }

Muuttui muotoon

table <tbl.r42.s> { 172.16.16.50 , 172.16.16.60 }

Tämä myös tietenkin aiheutti sen, että kaikki viittaukset <fsecure> tableen viitattiin nyt <tbl.r42.s> merkinnällä. Kaikkien kommenttien puuttuminen ja taulujen eri nimet vaikeuttaisivat huomattavasti palomuurin käyttöä. Myös kaikki muut nimetyt asiat, kuten verkko-alueet ja muut ovat menettäneet nimensä ja ne on korvattu FWBuilderin omilla nimillä. Muissa tapauksissa säännöt kyllä ovat aivan syntaksin mukaisia ja varmaan toimisivatkin Packet Filterissä, mutta kaikkien kommenttien puuttuminen ja jokaisen taulun uudelleen nimeäminen tuottaisi luultavasti ongelmia tai vaatisi vähintäänkin kaikkien taulujen nimien uudelleen opettelun ja kommenttien uudelleen kirjoittamisen.

Ratkaisuksi tämän kyseisen palomuurin siistimiseen en voi FWBuilderia suositella. Kaikkien kommenttien uudelleen kirjoittaminen FWBuilderiin olisi erittäin työlästä ja koska FWBuilder nimeää myös tablet uudestaan tämä hankaloittaisi enemmän sen käyttöä kuin auttaisi. Graafinen käyttöliittymä on ihan hieno, mutta kokooja on erittäin tarkka ja valitti paljon myös varjo säännöistä joita en saanut ratkaistua. Myös kommenttien poistuminen configuration tiedostosta vaikeutti sen lukemista, koska

kaikki säännöt olivat nyt vain nimetty "Rule 61, Rule 62" jne. Kommentteja voi kyllä lisätä Firewall Builderiin mutta nekään eivät näy configuration tiedostossa yhtä hyvin kuin alkuperäiset kommentit.

Ratkaisuksi olisi todennäköisesti paras aloittaa täysin alusta määrittelemällä palomuuri uudelleen ja päättää haluaako käyttää FWBuilderia vai ei. En oli varma olisiko tulos yhtä hyvä kuin ilman FWBuilderia mutta koska FW Builder haluaa käyttää omia taulu merkintöjään ja kommentit tulevat sen, kautta olisi palomuuri tehtävä joko heti alusta sen kanssa tai ei lainkaan sillä. Omasta mielestäni alkuperäinen configuration tiedosto on paljon selvempi, vaikka ehkä työläämpi koota, kuin jos sen tekisi Firewall Builderilla. Palomuurin uudistamisessa olisi myös mahdollisuus uusia FreeBSD ja saada siitä uusin versio käyttöön, mutta tämä saattaisi aiheuttaa ongelmia mikäli tietokoneen osat eivät enää sopisi uusimmalle versiolle FreeBSD:stä.

10 LOPPUSANAT

Työn toteutus osuus jäi hieman vähäiseksi, johtuen enimmäkseen Opetekin migraatiosta Porin kaupungin IT-palveluiden alaisuuteen, mutta olin silti tyytyväinen yhteistyöhön ja apuun, jota sain Opetekin henkilöstöltä. Alkuperäisenä tarkoituksenani oli työskennellä Opetekin henkilöstön kanssa ja yrittää auttaa ja oppia lisää palomuurin hallinnasta, mutta monesta syystä johtuen päädyimme hieman pienempään toteutus osuuteen. Toteutus osuuden koosta johtuen päätin kirjoittaa laajemmin palomuureista ja niiden toiminnasta. Tämä myös tarkoitti sitä, että otin huomioon muitakin asioita kuin pelkästään palomuurin toiminnan, kuten TCP/IP ja OSI mallin ja aliverkotuksen. Kirjoitin myös yleisesti eri palomuuri tyyppien toiminnasta.

Itselleni tämä opintonäytetyö oli hyvä mahdollisuus oppia itseäni kiinnostavasta aiheesta ja samalla tehdä jotain joka auttaa myös muita asiasta kiinnostuneita ymmärtämään palomuurien toimintaa. Samalla opein myös paljon lisää verkon toiminnasta jonka merkitystä palomuurien määrittelemisen kannalta en ollut vielä työn aloitus vaiheessa täysin ymmärtänyt. Tätä työtä tehdessä opein myös asioita joita en olisi ehkä muussa tapauksessa oppinut ja työn tekemisen myötä tajusin myös kuinka vä-

hän tiesin palomuuriin liittyvistä asioista. Tietoni palomuurien toiminnasta ja verkon toiminnasta yleensä on lisääntynyt huomattavasti tämän opinnäytetyön ansiosta.

LÄHTEET

Boyle, R & Panko, R 2013. Computer Corporate Security (3rd Edition). Pearson Education Inc.

Cisco 2011. IPv6 Security Brief Viitattu: 11.9.2014 Saatavissa: http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/enterprise-ipv6-solution/white_paper_c11-678658.pdf

Deering, S & Hinden, R 1998. RFC 2460 Internet Protocol, Version 6 (IPv6) Specification. The Internet Society .Viitattu: 1.10.2014. Saatavissa: https://www.ietf.org/rfc/rfc2460.txt

Firewall Builder 5 User Guide 2011. NetCitadel, LLC

Freed, N 2000. RFC 2979 Behavior of and Requirements for Internet Firewalls Viitattu: 2.10.2014. Saatavissa: https://www.ietf.org/rfc/rfc2979.txt

Frenzel, L 2013. What's The Difference Between The OSI Seven-Layer Network Model And TCP/IP Viitattu: 17.7.2014. Saatavissa: http://electronicdesign.com/what-s-difference-between/what-s-difference-between-osi-seven-layer-network-model-and-tcpip

Gavin 2014. FreeBSD Handbook. The FreeBSD Documentation Project.

Hansteen, P 2011. The Book of PF 2nd Edition. No Starch Press, Inc.

Leskiw, A Understanding Syslog: Servers, Messages & Security Viitattu: 22.9.2014 Saatavissa: http://www.networkmanagementsoftware.com/what-is-syslog

Lokiohje 2009. Valtiovarainministeriö. Viitattu: 25.9.2014 Saatavissa: http://www.vm.fi/vm/fi/04_julkaisut_ja_asiakirjat/01_julkaisut/05_valtionhallinnon_tietoturvallisuus/20090511Lokioh/Vahti_3_NETTI.pdf

Noonan, W & Dubrawsky, I 2006. Firewall Fundamentals. Cisco Press.

Ohlhorst, F 2013. Next-Generation Firewalls 101 Viitattu: 21.10.2014 Saatavissa: http://www.networkcomputing.com/careers-and-certifications/next-generation-firewalls-101/a/d-id/1234097?

Ou, G 2006. IP subnetting made easy Viitattu: 27.10.2014 Saatavissa: http://www.techrepublic.com/article/ip-subnetting-made-easy/

Passeri, P 2011. Next Generation Firewalls and Web applications firewalls Q&A Viitattu 21.10.2014 Saatavissa: http://hackmageddon.com/2011/10/07/next-generation-firewalls-and-web-applications-firewall-qa/

Shinder, D 2005. SolutionBase: Strengthen network defenses by using a DMZ. Viitattu: 12.7.2014. Saatavissa: http://www.techrepublic.com/article/solutionbase-strengthen-network-defenses-by-using-a-dmz/

Vacca, J 2009 Managing Information Security. Elsevier Inc.

Willard, M 2002 Getting most out of your Firewall Logs Viitattu: 11.9.2014. Saatavissa: http://www.sans.org/reading-room/whitepapers/firewalls/firewall-logs-811

Packet Filter

#---#
admin_if (vlan199, em0)
#---#
pass in quick on vlan199 proto { icmp, tcp, udp } from any to any keep state

pass out quick on vlan199 proto { icmp, tcp, udp } from 212.86.17.64/27 to 192.168.95.253/32 keep state
pass out quick on vlan199 proto { icmp, tcp, udp } from any to 192.168.95.25/32 keep state
pass out quick on vlan199 proto { icmp, tcp, udp } from any to 192.168.95.30/32 keep state
pass out quick on vlan199 proto { icmp, tcp, udp } from any to 192.168.95.31/32 keep state
pass out quick on vlan199 proto { tcp, udp } from any to 192.168.95.252/32 port = 3389 keep state
pass out quick on vlan199 proto { tcp } from 212.86.17.75/32 to 192.168.95.254/32 port = 22 keep state
pass out quick on vlan199 proto { icmp, tcp, udp } from <valvonta> to 192.168.95.0/24 keep state
pass out quick on vlan199 proto { icmp } from <wsus> to any keep state

Firewall Builder

Rule 9 (vlan199)
pass in quick on vlan199 inet proto tcp from any to any label "RULE 9 -- ACCEPT "
pass in quick on vlan199 inet proto udp from any to any label "RULE 9 -- ACCEPT "
#
Rule 10 (vlan199)
pass out quick on vlan199 inet proto tcp from 212.86.17.64/27 to 192.168.95.253 label "RULE 10 -- ACCEPT "
pass out quick on vlan199 inet proto udp from 212.86.17.64/27 to 192.168.95.253 label "RULE 10 -- ACCEPT "
#
Rule 11 (vlan199)
pass out quick on vlan199 inet proto tcp from any to 192.168.95.25 label "RULE 11 -- ACCEPT "
pass out quick on vlan199 inet proto udp from any to 192.168.95.25 label "RULE 11 -- ACCEPT "
#
Rule 12 (vlan199)
pass out quick on vlan199 inet proto tcp from any to 192.168.95.30 label "RULE 12 -- ACCEPT "
pass out quick on vlan199 inet proto udp from any to 192.168.95.30 label "RULE 12 -- ACCEPT "
#
Rule 13 (vlan199)
pass out quick on vlan199 inet proto tcp from any to 192.168.95.31 label "RULE 13 -- ACCEPT "

```
pass out  quick on vlan199 inet proto udp  from any  to 192.168.95.31  label "RULE 13 -- ACCEPT "
#
# Rule  14 (vlan199)
pass out  quick on vlan199 inet proto tcp  from any  to 192.168.95.252 port 3389  label "RULE 14 -- ACCEPT "
pass out  quick on vlan199 inet proto udp  from any  to 192.168.95.252 port 3389  label "RULE 14 -- ACCEPT "
#
# Rule  15 (vlan199)
pass out  quick on vlan199 inet proto tcp  from 212.86.17.75  to 192.168.95.254 port 22  label "RULE 15 -- ACCEPT "
#
# Rule  16 (vlan199)
pass out  quick on vlan199 inet proto tcp  from 212.86.17.76  to 192.168.95.0/24  label "RULE 16 -- ACCEPT "
pass out  quick on vlan199 inet proto udp  from 212.86.17.76  to 192.168.95.0/24  label "RULE 16 -- ACCEPT "
#
# Rule  17 (vlan199)
pass out  quick on vlan199 inet  from <tbl.r17.s>  to any  label "RULE 17 -- ACCEPT "
#
```