jamk.fi

BGP:n tietoturvalaajennusten vertailu

Erne Ruisaho

Opinnäytetyö
Joulukuu 2015

Tietotekniikan koulutusohjelma
Tekniikan ja liikenteen ala

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

Kuvailulehti

Tekijä(t)	Julkaisun laji	Päivämäärä
Ruisaho, Erne	Opinnäytetyö	11.12.2015
	Sivumäärä 117	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty: (X)

Työn nimi
BGP:n tietoturvalaajennusten vertailu

Koulutusohjelma
Tietotekniikan (tietoverkkotekniikan) koulutusohjelma

Työn ohjaaja(t)
Karo Saharinen
Antti Häkkinen

Toimeksiantaja(t)
JYVSECTEC
Marko Vatanen

Tiivistelmä

Opinnäytetyössä vertailtiin JYVSECTECille neljää eri BGP:n tietoturvalaajennusta. Tutkimuksen kohteina olivat Resource Public Key Infrastructure, Secure BGP, Secure Origin BGP sekä Interdomain Routing Validation.

RPKI-infrastruktuurista toteutettiin virtuaaliympäristössä käytännön implementaatio Cisco-, Juniper- ja QuaggaSRx-reitittimillä. Reittitiedon ja alkuperäisen mainostajan oikeellisuus tarkistettiin käyttäen RIPE:n RPKI Validator API -ohjelmaa. Tutkimuksessa saatiin turvattua autonomisen järjestelmän reititys virheellisiltä mainostuksilta. Muita laajennuksia ei voitu toteuttaa ohjelmistotuen puutteen takia.

Laajennusten vertailu pohjautui suurimmaksi osaksi teoriapohjaan, mutta RPKI:n kohdalla toteutettua implementaatiota ja julkaistua statistiikkaa käytettiin hyväksi. Kaikkien laajennuksien kohdalla niistä saadun hyödyn huomattiin olevan suoraan verrannollinen toimijoiden määrään.

Työssä toteutettua RPKI-ympäristöä voidaan käyttää pohjatietona tuleville toteutuksille.

Avainsanat
BGP, S-BGP, soBGP, IRV, RPKI, Prefix Origin Validation

Muut tiedot

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

Abstract

Author(s)	Type of publication	Date
Ruisaho, Erne	Bachelor's thesis	11.12.2015
	Number of pages 117	Language of publication Finnish
		Permission for web publication: (X)

Title of publication
A comparison between BGP security extensions

Degree programme
Information Technology

Tutor(s)
Karo Saharinen
Antti Häkkinen

Assigned by
JYVSECTEC
Marko Vatanen

Abstract

The research compares four security extensions for BGP as assigned by JYVSECTEC. The extensions in question were Resource Public Key Infrastructure, Secure BGP, Secure Origin BGP and Interdomain Routing Validation.

Due to software limitations, RPKI was the only one implemented using Cisco, Juniper and QuaggaSRx routers. RPKI Validator API developed by RIPE was used to examine whether a BGP speaker is authorized to advertise a route as its origin. An autonomous systems routing table was protected from unauthorized advertisements in the final implementation.

The benefits obtained by the extensions were found to be extremely dependant on the magnitude of their user base. The security extensions were compared mainly using the document that defines them as the basis. With RPKI there was also the possibility of using the gathered research data as a source.

The implementation described may be used as a guideline when constructing an RPKI-environment.

Keywords/tags
BGP, S-BGP, soBGP, IRV, RPKI, Prefix Origin Validation

Miscellaneous

SISÄLTÖ

KUVIOT

TAULUKKOLUETTELO

LYHENTEET

AA	Address Attestation
AS	Autonomous System
BGP	Border Gateway Protocol
CA	Certification Authority
CIDR	Classless Inter-Domain Routing
CRL	Certificate Revocation List
CSV	Comma Separated Value
DNS	Domain Name System
EBGP	External Border Gateway Protocol
EE	End-Entity
ESP	Encapsulating Security Payload
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
IBGP	Internal Border Gateway Protocol
IGP	Interior Gateway Procotol
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IRV (arkkitehtuuri)	Interdomain Routing Validation
IRV (palvelin)	Interdomain Routing Validator
ISP	Internet Service Provider
IT	Informaatioteknologia
IXP	Internet Exchange Point
JSON	JavaScript Object Notation
JYVSECTEC	JYVSECTEC - Jyväskylä Security Technology
MD5	Message Digest 5
MITM	Man-in-the-Middle
NIR	National Internet Registry
NIST	National Institute of Standards and Technology
NLRI	Network Layer Reachability Information
PKI	Public Key Infrastructure
RA	Route Attestation
REST	Representational State Transfer
RFC	Request For Comments
RGCE	Real Global Cyber Environment
RIB	Routing Information Base
RIPE	Réseaux IP Européens
RIR	Regional Internet Registry
ROA	Route Origin Authorization
RPKI	Resource Public Key Infrastructure
RPSL	Routing Policy Specification Language
RTR	Resource Public Key Infrastructure To Router

S-BGP	Secure BGP
soBGP	Secure Origin BGP
SSL	Secure Sockets Layer
TA	Trust Anchor
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TLV	Type-Length-Value
TTL	Time-To-Live
VRP	Validated ROA Payload
XML	Extensible Markup Language

1 TYÖN LÄHTÖKOHDAT

1.1 Toimeksiantaja

Opinnäytetyön toimeksiantaja oli Jyväskylän ammattikorkeakoulun IT-instituutin tiloissa toimiva puolueeton kyberturvallisuuden tutkimus-, kehitys- ja koulutuskeskus JYVSECTEC - Jyväskylä Security Technology. Se tarjoaa asiantuntijuutta sekä järjestelmiä eri organisaatioiden kyberturvallisuustietämyksen kehittämiseen hyödyntäen varta vasten rakennettua Real Global Cyber Environment-kehitysympäristöä. Ympäristö mahdollistaa kyberturvallisuusharjoitusten järjestämisen asiantuntevassa ohjauksessa. JYVSECTEC-hanke sai keväällä 2015 kahden miljoonan euron jatkorahoituksen vuoden 2017 loppuun saakka. (JYVSECTEC - Tietoa meistä 2015.)

Real Global Cyber Environment (RGCE) on kyberturvallisuuden tutkimiseen ja mallintamiseen kehitetty tutkimus-, kehitys- ja koulutusympäristö. Se on ulkoverkosta eristetty kokonaisuus, jonka toiminta mallintaa Internetin rakennetta ja toiminnallisuuksia. Oikean elämän Internet-palveluntarjoajat on toteutettu maantieteellisellä tarkkuudella ja autenttisilla julkisilla IP-osoitteilla. Lisäksi ne tuottavat ympäristöön erilaisia palveluita, kuten nimipalvelun tosimaailman hierarkiaa vastaten. Ympäristön sisällä voidaan generoida liikennettä JYVSECTECin verkkoliikenteen generointisovelluksella ja täten mallintaa asiakasliikennettä, palvelunestohyökkäyksiä tai erilaisia muita hyökkäyksiä. (JYVSECTEC - RGCE - Kybertoimintaympäristö 2015.)

1.2 Tavoitteet

Opinnäytetyön tavoitteena oli verrata BGP-reititysprotokollan tietoturvalaajennusten toimintaperiaatteita ja vertailla niitä toisiinsa. Työn kohteeksi valittiin neljä toisiaan vastaan kilpailevaa laajennusta: S-BGP, soBGP, IRV ja RPKI. Valintaperusteena käytettiin kehitettyihin laajennuksiin alustavaa tutustumista ja työn tekijän harkintaa. Yksi

osuus työstä oli tutustua kyseisten laajennusten käytännön implementaatioihin ja toteuttaa ympäristö, jolla voidaan testata niiden toimintaa ja käyttöönottoa kontrolloidussa ympäristössä. Toteutusosuus rakennettiin JYVSECTECin tarjoamilla laitteilla ja resursseilla. Alun perin epäiltiin, että kaikilla laajennuksista ei ole laitevalmistajien ohjelmistotukea, joten työhön nostettiin myös laajennusten tietoturvan, asteittaisen implementoinnin ja mahdollisten ongelmien vertailu. Kilpailuaseman vuoksi laajennusten yleistymisen todennäköisyyttäkin pohdittiin.

2 TIETOLIIKENNEPROTOKOLLAT

2.1 Internet Protocol version 4

2.1.1 Yleistä

Internet Protocol version 4 (IPv4 tai IP) on tietoliikenteessä käytetty protokolla, joka on suunniteltu mahdollistamaan bittien lähettäminen toisiinsa yhdistettyjen isäntäyksiköiden välillä. Lähde- ja kohdeisäntien välillä lähetettyjä bittikokonaisuuksia kutsutaan datagrammeiksi. IP hyödyntää alemman tason verkkoprotokollia toimiessaan kuljetusvälineenä ylemmän tason protokollille. Protokollana se toimii siirtoyhteys- ja kuljetuskerroksen välissä. (Postel 1981a, 1-2.)

Jotta kaksi isäntäkonetta pystyy kommunikoimaan keskenään, tulee niillä molemmilla olla neljän oktetin eli 32 bitin mittaiset yksilöivät osoitteet. Näitä kutsutaan lähde- ja kohdeosoitteiksi. Postelin (1981b, 2) mukaan osoitteet voidaan kirjoittaa erottamalla oktettien sisältämät desimaaliarvot pisteillä. IP-osoitteet voidaan jakaa kolmeen eri luokkaan niiden sisältämien verkko- ja paikallisosien mukaan seuraavasti: A-luokan osoitteen merkittävin bitti on aina nolla, ja ensimmäisen oktetin jäljelle jäävät seitsemän bittiä muodostavat varsinaisen verkko-osan. Loput 24 bittiä eli kolme oktettia muodostavat osoitteen paikallisosan. B-luokan osoitteessa kaksi merkittävintä bittiä ovat yksi-nolla (10) ja verkko-osa muodostuu kahden ensimmäisen oktetin jäljelle jäävistä biteistä. Käyttämättömistä biteistä tulee osoitteen paikallisosa. Luokan C osoitteet alkavat aina bittisarjalla 110, ja seuraavat 21 bittiä kuuluvat verkko-osaan jättäen paikallisosalle yhteensä kahdeksan bittiä. Lisäksi osoiteavaruuden lopusta on varattu tulevaisuuden käyttöä varten bittikuviolla 111 alkavat osoitteet. Näillä säännöillä voidaan taulukon 1 mukaisesti laskea, kuinka monta osoitevariaatiota kullakin osoiteluokalla on. (Postel 1981a, 1, 2, 5, 7, 24.)

Taulukko 1. IP-osoiteavaruuden jakautuminen

Luokka	Bittikuvio	Verkko-osia	Paikallisosia
A	0	126	16 777 214
B	10	16 382	65 534
C	110	2 097 150	254

Keskenään kommunikoivat isäntälaitteet voivat sijaita itsenäisesti toimivissa verkoissa, jotka eivät ole suoranaisesti yhteydessä toisiinsa. Kommunikointi perustuu siihen, että yksittäinen datagrammi liikkuu verkossa toimivalta komponentilta toiselle, kunnes se saavuttaa päämääränsä eli kohdeosoitteensa. Tässä se luottaa täysin verkon komponenttien tekemiin päätöksiin, mitä kutsutaan reitittämiseksi. Reitittäminen perustuu sääntöihin ja niiden suhteeseen IP-datagrammissa olevien lähde- ja kohdeosoitteen kanssa. Jokaista datagrammia kohdellaan komponenttien toimesta yksilöllisesti, eivätkä reitityspäätökset ole sidottuja ennalta määrättyihin polkuihin. IP-datagrammissa on määritettynä myös kenttä nimeltä Time to Live (TTL), joka estää niitä kiertämässä verkossa loputtomasti. Jokainen komponentti, joka prosessoi verkossa liikkuvan datagrammin, vähentää olemassa olevaa TTL-arvoa yhdellä ennen datagrammin edelleen lähettämistä. Mikäli TTL-luku, jonka maksimiarvo on 255, saavuttaa arvon nolla ennen määränpäähänsä saapumista, datagrammi tuhotaan. (Postel 1981a, 2, 7, 14.)

2.1.2 Classless Inter-domain Routing

Kuten aiemman osion IP-osoiteavaruuden jakautumista käsittelevästä taulukosta 1 voidaan havaita, IP-osoitteet ovat jakautuneet verkko- ja isäntäosoitteiksi hyvin epätasaisesti. Esimerkiksi B-luokan verkko mahdollistaa yli 65 000 isäntäkonetta, kun taas ainoa pienempi vaihtoehto eli C-luokan verkko sisältää 254. Kyseinen osoitteistus jättää pahimmassa tapauksessa paljon hukkaan meneviä osoitteita ja kiihdyttää

32-bittisen osoiteavaruuden loppumista. Ratkaisuksi kehitettiin Classless Inter-domain Routing (CIDR), joka hylkää osoitteiden luokkajaon ja mahdollistaa täsmällisen verkko- ja paikallisosan määrittelyn. (Fuller & Li 2006, 3-7.)

CIDR:in tuomassa merkintätavassa IP-osoitteen jälkeen merkitään kauttaviiva, jonka jälkeinen luku kertoo yksiselitteisesti, montako bittiä osoitteen verkko-osassa on. Loput bitit kuuluvat isäntäkoneille tarkoitettuun paikallisosaan. CIDR:illä määritettyä verkko-osuutta kutsutaan nimellä prefiksi. Esimerkiksi merkintä 172.16.100.0/25 kertoo osoitteen 25 merkitsevimmän bitin muodostavan verkko-osan jättäen paikallisosalle seitsemän bittiä. Aiemmin käytetyn luokallisen järjestelmän osoitteet voidaan ilmaista myös CIDR:iä käyttäen. Osoite 10.0.0.0/8 mukailee luokan A osoitetta. Prefiksihierarkia toi mukanaan kaksi uutta osoitetta: x.x.x.x/32 ja 0.0.0.0/0. Näistä ensimmäinen viittaa yksittäiseen isäntäkoneeseen merkaten sitä omana verkkonaan (32 verkko-osan bittiä). Jälkimmäistä notaatiota käytetään oletusreittinä, koska se pitää sisällään kaikki mahdolliset prefiksit. (Fuller & Li 2006, 3-7, 12.)

Luokaton reititysjärjestelmä mahdollistaa myös reittitietojen yhteen niputtamisen. Mikäli kaksi prefiksin /26 verkkoa sijaitsevat loogisesti prefiksin /24 verkon takana, (tarkoittaen, että pienempiin verkkoihin menevät paketit kiertävät isomman verkon läpi), riittää isomman verkon komponentille, kun se mainostaa hallinnoimiaan verkkojaan prefiksillä /24 naapureilleen. Mikäli pienempään verkkoon matkalla oleva paketti ohjautuu kyseiselle komponentille, se voi tehdä reitityspäätöksen oman tarkemman reittinsä perusteella. Näin yksi niputettu reitti mahdollistaa tavoitettavuuden monille pienemmille verkoille. (Mts. 4, 9.)

Aiemmasta luokallisesta järjestelmästä poiketen CIDR:in käyttöönotto vaati, että 32-bittinen prefiksitieto kuljetetaan täsmällisesti reititystiedon mukana, jotta verkkojen sijainnit voidaan yksilöidä ja tehdä tarvittavat reitityspäätökset. Prefiksi kertoo, kuinka monta merkittävintä bittiä verkkomaskissa merkitään bitillä yksi. Esimerkiksi prefiksi /24 tarkoittaa maskissa olevan 24 ykkösbittiä ja kahdeksan nollabittiä. Se voidaan merkitä desimaaleina muotoon 255.255.255.0. Luokallisessa järjestelmässä

lähde- ja kohdeosoite kuuluivat johonkin luokkaan kolmen merkittävimmän bittinsä johdosta. Tämä ero merkitsee, että luokallinen reititysprotokolla kuten RIP ei toimi CIDR-merkinnän kanssa, koska sen reittimainostukset eivät sisällä prefiksitietoa. (Mts. 5-6, 11.)

2.2 Border Gateway Protocol 4

2.2.1 Yleistä

Border Gateway Protocol 4 (BGP) on reititysprotokolla, joka toimii autonomisten järjestelmien (Autonomous System, AS) välisenä reititysprotokollana. Autonominen järjestelmä tarkoittaa yksittäisen teknisen hallinnon alla olevia reitittimiä, jotka ovat yhteydessä toisiinsa sisäisen reititysprotokollan avulla. Sisäinen reititysprotokolla (internal gateway protocol, IGP) huolehtii pakettien reitittämisestä AS:n sisällä. BGP:n tärkein toiminallisuus on välittää verkkojen saatavuustietoja BGP:tä käyttävien autonomisten järjestelmien välillä. BGP:n toiminta reititysprotokollana perustuu pelkästään IP-datagrammin kohdeosoitteen mukaan tehtävään reititykseen. Muodostuvat reittitiedot pitävät sisällään kuljetut AS-alueet, joista muodostuvaa sarjaa kutsutaan AS-poluksi. AS-poluista voidaan muodostaa AS-alueita kuvaava kaavio, jota tutkimalla voidaan estää reitityssilmukat. BGP:n suunnittelussa on otettu huomioon CIDR sekä reittien niputtaminen. (Rekhter, Li & Hares 2006, 4.)

BGP toimii TCP-protokollan päällä käyttäen porttia 179 ja hyödyntäen sen ominaisuuksia. TCP:n määrittelevässä RFC:ssä 793 mainitaan, kuinka kyseinen protokolla käyttää järjestysnumeroita, kuittausviestejä sekä uudelleenlähettämistä varmistaakseen vastaanottajan saavan kaikki halutut paketit onnistuneesti (Postel 1981c, 10). Näin ollen kyseiset toiminteet on tietoisesti jätetty pois BGP:stä. (Rekhter, Li & Hares 2006, 8.)

Kahden BGP:tä käyttävän reitittimen välistä yhteyttä kutsutaan naapuruudeksi. Mikäli reitittimet kuuluvat samaan AS-alueeseen, on kyseessä sisäinen BGP (internal

BGP, IBGP). Mikäli reitittimet ovat eri autonomisista järjestelmistä, kutsutaan naapuruutta ulkoiseksi BGP:ksi (external BGP, EBGP). Mikäli yhdellä AS-alueella on useampi kuin yksi EBGP-liityntäpiste ulkoverkkoon, on tärkeää pitää jokainen reunareititin tietoisena kyseisen AS-alueen sisäisistä reiteistä käyttäen sisäistä reititysprotokollaa. Näin reitityspäätökset pysyvät johdonmukaisina riippumatta siitä, minkä EBGP-reitittimen kautta liikenne tulee. (Mts. 9.)

Naapureina toimivat BGP-reitittimet mainostavat osaamiansa reittejä UPDATE-viestillä, joka sisältää kohdeverkkojen prefiksit Network Layer Reachability Information-kentässä (NLRI). Tätä kautta opittuja reittejä hyödynnetään erinäisissä reititystietokantoissa (RIB). Saadessaan uuden reitin UPDATE-viestillä BGP-reititin tallentaa sen Adj-RIBs-In-tietokantaan, joka sisältää kaikki naapureilta opitut reitit. Tämän jälkeen reititin vertaa opittuja reittejä ennalta määritettyihin paikallisiin sääntöihin, joiden perusteella se valitsee sopivimmat reitit omiin reitityspäätöksiinsä. Valitut reitit pidetään tallessa Loc-RIB-kannassa ja ne voidaan viedä reitittimen reititystauluun. Säännöistä huolimatta tutkittavan reitin seuraavan hypyn tulee löytyä reititystaulusta, jotta reitti voidaan huolia Loc-RIB:iin. (Mts. 9-10.)

Viimeinen tietokanta, Adj-RIBs-Out, sisältää reititystiedot, joita BGP-reititin tulee mainostamaan naapureilleen käyttäen UPDATE-viestiä. BGP-naapurit voivat poistaa reittejä toistensa tiedoista lähettämällä UPDATE-viestin WITHDRAWN ROUTES-kentän ja halutun prefiksin kera, mainostamalla eri reittiä halutulle prefiksille tai katkaisemalla BGP-yhteyden. BGP-yhteyden katkeaminen poistaa automaattisesti kaikki poistuneelta naapurilta opitut reitit. (Mts. 9-10.)

Mainostuksista opittujen reittien paremmuus ratkaistaan kolmivaiheisessa vertailuprosessissa. Ensimmäisessä vaiheessa uusille vastaanotetuille reittitiedoille asetetaan LOCAL PREFERENCE -arvot. Mikäli mainostus on saatu IBGP-naapurilta, käytetään naapurin UPDATE-viestissä määrittämää arvoa, mikäli jokin reitittimen ylläpitäjän määrittämä sääntö ei ylikirjoita kyseistä arvoa. EBGP-naapureilta opittujen reittien

LOCAL PREFERENCE -arvot tulee määrittää reitittimeen konfiguroiduilla säännöillä. (Mts. 76-77.)

Tämän jälkeen siirrytään toiseen vaiheeseen, jossa reititin vertailee kaikkia oppimiaan reittejä ja valitsee niistä käyttöön sen, jonka LOCAL PREFERENCE -arvo kyseiseen verkkoon on korkein tai joka on ainoa reittitieto sinne. Mikäli BGP-reititin on oppinut monia reittitietoja kohdeverkkoon identtisillä painotusarvoilla, ratkaistaan tasapeli seuraavasti:

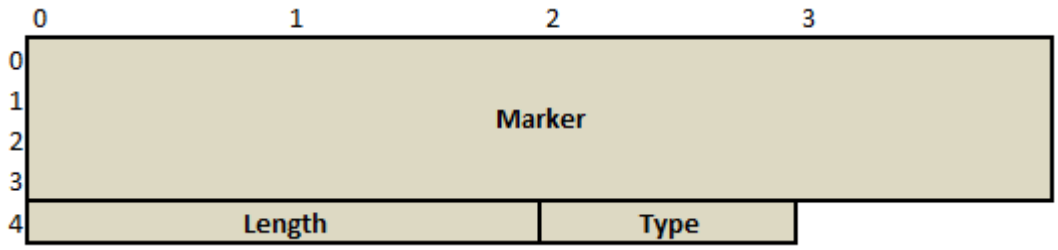
- Eliminoidaan reitit, joiden AS_PATH-polun pituus ei ole lyhin mainostetuista. Mikäli reitille on määritetty AS_SET-arvo, lasketaan polun pituudeksi yksi.
- Eliminoidaan reitit, joiden ORIGIN-arvo ei ole pienin.
- Eliminoidaan reitit, joiden MULTI_EXIT_DISC-arvo ei ole kaikista alin. Reiteille, joille ei ole määritetty kyseistä arvoa, lasketaan arvoksi alin mahdollinen arvo eli nolla.
- Jos ainakin yksi reiteistä on opittu EBGP-naapurilta, eliminoidaan kaikki IBGP-naapureilta opitut reitit.
- Eliminoidaan reitit, joiden reititystaulusta saatu laskennallinen metriikka NEXT_HOP-osoitteeseen ei ole paras. Mikäli metriikoita ei voida laskea, tulee tämä sääntö jättää väliin.
- Eliminoidaan reitit, joiden BGP-naapurin 4-oktettinen BGP-tunniste ei ole alimmainen.
- Valitaan reitti, jonka BGP-naapurin IP-osoite on alimmainen. (Mts. 77-82.)

Toisen vaiheen valintaprosessia jatketaan, kunnes vain yksi reitti on jäljellä, minkä jälkeen siitä tulee paras reitti ja se otetaan käyttöön Loc-RIB-reititystaulussa korvaten siellä aiemmin käytetyn reittitiedon kyseiseen verkkoon. Kolmannessa vaiheessa valittu reitti otetaan käyttöön naapureille mainostettavassa Adj-RIBs-Out-taulussa, minkä jälkeen se mainostetaan niille uudella UPDATE-viestillä. (Mts. 78, 82-83.)

2.2.2 BGP:n kehysrakenne

Jokainen BGP:n käyttämästä viestistä alkaa **aina** 19 oktetin mittaisella otsikkokehyksellä. Se koostuu 16 oktetin mittaisesta yhteensopivuutta varten käytettävästä Marker-kentästä, jonka bittiarvojen tulee olla pelkkiä ykkösiä. 2 oktetin mittainen Length-kenttä puolestaan kertoo yksiselitteisesti, kuinka pitkästä viestistä on kyse. Kyseinen kenttä voi saada arvoja väliltä 19-4096, mikä määrittää BGP-viestien pituusskaalan. Viesti, joka koostuu pelkästä otsikkokehyksestä, sisältää arvon 19 Length-kentässä. Viimeinen kenttä, Type, on kooltaan yhden oktetin ja voi saada seuraavat arvot: 1 (OPEN), 2 (UPDATE), 3 (NOTIFICATION) tai 4 (KEEPALIVE). Type määrittää, minkä tyyppisestä viestistä on kyse. (Rekhter, Li & Hares 2006, 11-12).

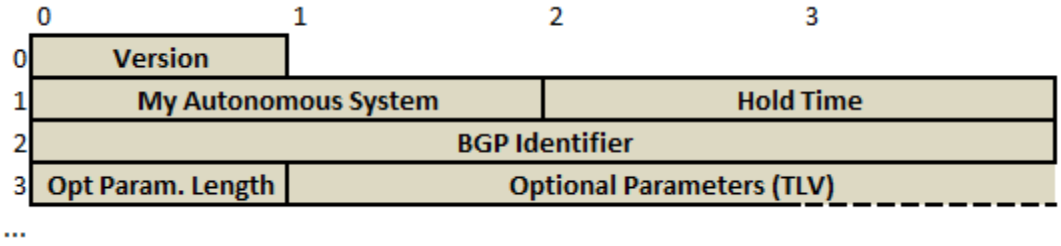
KEEPALIVE-tyypin viesti koostuu pelkästään otsikkokehyksestä, ja sitä käytetään pitämään BGP-naapuruutta yllä. Kummankin osapuolen tulee lähettää KEEPALIVE-viestejä, ennen kuin määritetty Hold Time -aikaraja täyttyy. Oletuksena viestien lähetysväli on kolmasosa edellä mainitusta aikarajasta. Mikäli Hold Time -raja täyttyy, oletetaan BGP-naapurin olevan tavoittamattomissa ja BGP-yhteys suljetaan. Hold Time -aikaraja resetoidaan myös UPDATE- tai NOTIFICATION-viestin yhteydessä. Kuviossa 1 on kuvattu BGP:n otsikkokehys. (Mts. 12, 21, 34.)



Kuvio 1. BGP:n otsikkokehys

OPEN-viesti on ensimmäinen BGP-protokollaa käyttävä viesti naapuruutta muodostavien reitittimien välillä, ja se lähetetään heti TCP-yhteyden muodostamisen jälkeen. Viestin minimipituus on 29 oktettia. OPEN-viestin ensimmäinen kenttä on yhden ok-

tetin mittainen Version, joka kertoo käytetyn BGP-version. Tämän dokumentin puitteissa versiota neljä oletetaan käytettäväksi. Kahden oktetin mittainen My Autonomous System -kenttä ilmaisee lähettäjän AS-alueen tunnisteen. Desimaaleina arvot vaihtelevat välillä 0-65535. Toinen kahden oktetin mittainen kenttä, Hold Time, kertoo sekunteina kauanko naapuruutta pidetään hengissä, mikäli KEEPALIVE- tai UPDATE-viestiä ei olla vastaanotettu. Oletuksena BGP-reitittimen tulee käyttää sille konfiguroidusta ja OPEN-viestissä vastaanotetusta Hold Time -arvosta pienempää. Naapuruus on mahdollista hylätä epäsopivan arvon johdosta. BGP Identifier on neljän oktetin mittainen kenttä, jota käytetään yksilöimään BGP-kumppani. Arvoa kuvataan IP-osoitteella. Seuraava kenttä on yhden oktetin mittainen Optional Parameters Length, joka kuvaa sitä seuraavan kentän pituutta oktetteina. Viimeinen kenttä on Type-Length-Value-tyyppinen (TLV) kokonaisuus, joka kertoo käytetyn valinnaisen asetuksen. Se koostuu yhden oktetin mittaisista Parameter Type ja Parameter Length -kentistä, joiden arvot määrittävät vaihtelevan mittaisen Parameter Type -kentän arvon. Näiden kenttien käyttö on tämän dokumentin rajojen ulkopuolella. OPEN-viestin rakenne löytyy kuviosta 2. (Mts. 13-14.)



Kuvio 2. OPEN-viestin kehysrakenne

UPDATE-viestiä käytetään BGP-naapureiden väliseen reittitietojen vaihtamiseen. Tietojen vaihtaminen mahdollistaa sekä reittien lisäämisen että niiden poistamisen. Lopputuloksena saaduista tiedoista voidaan muodostaa AS-alueiden välisiä suhteita kuvaava kaavio, jota voidaan hyödyntää reitityssilmukoiden eliminoimiseen. Viestin ensimmäinen kahden oktetin mittainen kenttä eli Withdrawn Routes Length kertoo sitä

seuraavan kentän pituuden okteteissa. Mikäli kentän arvo on nolla, ei seuraavaksi käsiteltävää kenttää ole viestissä ollenkaan. Vaihtelevan mittainen Withdrawn Routes-kenttä määrittää reititystiedoista poistettavat prefiksit. Prefiksit ilmaistaan yhden oktetin mittaisella Length-kentällä sekä vaihtelevalla Prefix-kentällä. Yhdessä nämä kaksi arvoa kertovat yksiselitteisesti poistettavan reittitiedon. Poistettavia tietoja voi olla yhdessä UPDATE-viestissä useita riippuen Withdrawn Routes Length -kentän arvosta. Total Path Attribute Length on kahden oktetin mittainen kenttä, joka kuvaa oktetteina Path Attributes -kentän pituuden. Mikäli kentän arvo on nolla, ei viestissä ole kumpaakaan jäljellä olevista kentistä. (Mts. 14-16.)

Path Attributes on sarja vaihtelevan mittaisia TLV-rakennetta mukailevia kenttiä. Näistä Attribute Type ja Attribute Length ovat suurimmaksi osaksi tämän dokumentin rajojen ulkopuolella. Ne määrittävät, kuinka Attribute Value -kentän arvoa tulkitaan ja kuinka pitkä kyseinen kenttä on. Attribute Type pitää kuitenkin sisällään pakollisia Attribute Type Code -oktetetteja, jotka kertovat polun ominaisuuksista ja ovat sidottu suoraan Attribute Value -arvoon seuraavasti (Mts. 16-17):

ORIGIN (Type Code 1) on tietue, joka kertoo reittitiedon alkuperän. Alkuperä on opittu sisäiseltä reititysprotokollalta Value-arvon ollessa nolla. Arvon ollessa yksi on alkuperänä ulkoinen reititysprotokolla. Viimeinen vaihtoehto on arvo kaksi, joka tarkoittaa, että tieto on opittu jotain muuta kautta. Tietueen arvon muodostaa reittiä alun perin mainostava BGP-reititin. (Mts. 18, 25.)

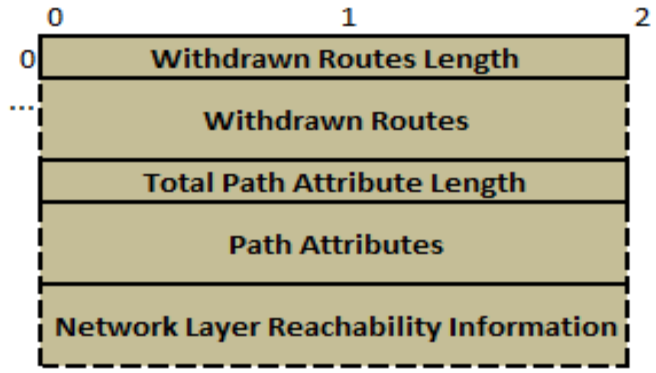
AS_PATH (2) kertoo reittitiedon kulkemien AS-alueiden sarjan ja se on pakollinen tietue, joka muodostuu TLV-kentistä. Tyyppikenttä kertoo onko mainittu AS-sarja oikeassa järjestyksessä (arvo yksi) vai ei (arvo kaksi). Yhden oktetin mittainen pituuskenttä kertoo, kuinka monta AS-aluetta sarja pitää sisällään. Se myös rajoittaa niiden maksimimääräksi 255. Arvokenttä kertoo varsinaisen AS-sarjan pitäen sisällään vähintään yhden AS-alueen. Yhden AS-alueen kuvaamiseen on varattu kaksi oktettia. Mikäli mainostuksessa on kyseessä BGP-reitittimen omistama alkuperäinen reitti, se

laittaa vain oman AS-numeronsa arvokenttään ja määrittää sarjan olevan järjestyksessä tyyppikentän arvolla yksi. Mikäli BGP-reititin mainostaa aiemmin opittua reittitietoa UPDATE-viestillä ulkoiselle naapurille, se lisää oman AS-numeronsa arvokentän ensimmäiseksi tiedoksi sekä kasvattaa pituuskentän arvoa yhdellä. Sisäiselle naapurille mainostettaessa AS_PATH-tietuetta ei muokata. (Mts. 18, 25-26.)

NEXT_HOP (3) on tietue, joka kertoo, mitä reititintä mainostetuille reiteille tulisi käyttää seuraavana hyppynä. Arvoa ilmaistaan halutun reitittimen IP-osoitteella. Ulkoiselta naapurilta opittua reittiä sisäiselle naapurille mainostettaessa ei reititin lähtökohtaisesti muokkaa tietueen arvoa. Oletuksena ulkoisille naapureille reittejä mainostetaan käyttäen naapuruussuhteen muodostamiseen käytetyn rajanpinnan IP-osoitetta NEXT_HOP-arvona. UPDATE-viestin vastaanottanut reititin tarkastaa tiedon paikkansapitävyyden rekursiivisella reitinhaulla, joka paljastaa myös ulospäin lähtevän rajapinnan kyseisellä reitittimellä. BGP-reititin ei saa hyväksyä reittejä, joissa se on itse seuraavana hyppynä. (Mts. 19, 26-28.)

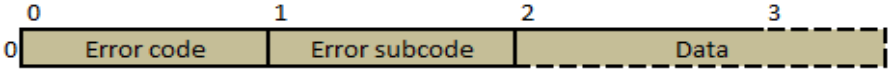
Edellä mainittujen pakollisten tietueiden lisäksi on olemassa vapaavalintaisia tietueita, joiden käyttötarkoitukset käydään lyhyesti läpi. MULTI_EXIT_DISC:iä (4) voidaan käyttää metriikkana reitityspäätöksissä, joissa kyseessä on useita yhdistymispisteitä samaan naapuri-AS-alueeseen. BGP-naapurina toimiva reititin voi kertoa oman näkemyksensä parhaasta reitistä sen autonomiseen järjestelmään. UPDATE-viestin vastaanottava reititin voi halutessaan hylätä arvon. LOCAL_PREF-tietueella (5) sisäiset BGP-naapurit voivat määrittää halutut reitit etusijalle reitityspäätöksiä tehtäessä. Kyseistä tietuetta ei käytetä ulkoisille naapureille mainostaessa. ATOMIC_AGGREGATE (6) mahdollistaa BGP-reitittimen poistaa AS_PATH-kentästä yhden tai useamman AS-alueen, mikäli se tietää, että tämä ei aiheuta reitityssilmukoita verkkoon. Tämä muokkaus puolestaan mahdollistaa useamman reititystiedon mainostamisen samalla UPDATE-viestillä, mikäli alun perin AS-sarjat eivät olleet identtiset. AGGREGATOR (7) antaa BGP-reitittimelle mahdollisuuden yhdistää monia pienempiä verkkoja yhdeksi suuremmaksi verkoksi, minkä jälkeen sille riittää yksi UPDATE-viesti koko verkkoalueen mainostamiseksi. (Mts. 19, 28-30, 87)

UPDATE-viestin viimeinen kenttä on vaihtelevan mittainen Network Layer Reachability Information (NLRI). Se sisältää mainostetut verkot, joihin kaikki Path Attributes -kentän määrittämät ominaisuudet täsmäävät. Verkko ilmaistaan yhden oktetin mittaisella Length-kentällä, joka kertoo verkon prefiksin pituuden bitteinä, sekä maksimissaan neljän oktetin mittaisella Prefix-kentällä, joka kertoo IP-osoitteena ilmaistuna varsinaisen verkko-osan. Kuviossa 3 on UPDATE-viesti kokonaisuudessaan. (Mts. 20.)



Kuvio 3. BGP:n UPDATE-viestin kehysrakenne

Viimeinen viestityyppi on NOTIFICATION, jota käytetään BGP-yhteyden sulkemiseen virhetilanteessa. Error code on 1 oktetin pituinen kenttä, joka ilmaisee, mistä virhetilanteesta on kyse. Sitä seuraava samanpituinen Error subcode täsmentää sattunutta virhetilannetta. Viimeinen kenttä, Data, on vaihtelevan mittainen, ja sitä käytetään virhetilanteen tarkempaan diagnosoitiin. Esimerkiksi yhteensopimatonta BGP-versiota käyttävälle naapurille lähetettäisiin NOTIFICATION-viesti Error code -arvolla yksi, Error subcode-arvolla yksi ja Data-kentän kahden oktetin mittaisella lukuarvolla, mikä kertoo paikallisen reitittimen tukeman versionumeron, joka on lähimpänä naapurin ehdottamaa arvoa. Viestin kehysrakenne käy ilmi kuviosta 4. (Mts. 21-23, 31-32.)



Kuvio 4. BGP:n NOTIFICATION-viestin kehysrakenne

2.2.3 BGP:n tilakone

BGP:n toiminta ja naapuruuksien muodostaminen perustuu tilakoneeseen. Jokainen BGP-naapuruus tarvitsee oman tilakoneensa, joka ei ole yhteydessä muihin reitittimen ylläpitämiin tilakoneisiin. Tässä dokumentissa käydään läpi tilakoneen toiminta tiivistetysti. (Rekhter, Li & Hares 2006, 53.)

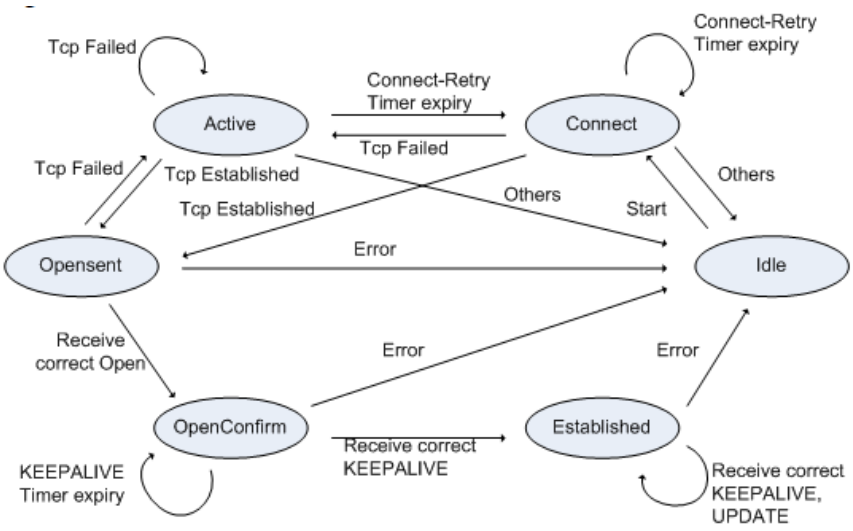
Alkutilanteessa BGP on Idle-tilassa, jolloin se hylkää kaikki vastaanottamansa BGP-viestit ja -yhteydet. Järjestelmä ei varaa BGP:tä varten mitään resursseja tässä tilassa. Mikä tahansa BGP-reitittimen havaitsema virhetilanne tai NOTIFICATION-viestin vastaanottaminen tiputtaa BGP:n tilan takaisin Idleen huolimatta siitä, missä tilassa BGP aiemmin oli. Samaan lopputulokseen päästään myös pysäyttämällä BGP-prosessi. Idle-tilasta päästään pois vain Start-tapahtumalla, joka voidaan panna alulle joko automaattisesti tai manuaalisesti. Sen jälkeen BGP alustaa tarvittavat resurssit, aloittaa TCP-yhteyden haluttuun naapuriin sekä kuuntelee naapurilta tulevia yhteydenottoyrityksiä. BGP:n tilaksi muuttuu Connect. (Rekhter, Li & Hares 2006, 53-54.)

Connect-tilassa BGP-reititin odottaa, että TCP-yhteys toiseen osapuoleen saadaan muodostettua. Tämän tapahduttua BGP:n alustus viimeistellään, naapurille lähetetään OPEN-viesti ja BGP:n tilaksi muutetaan OpenSent. Mikäli Connect-tilassa vastaanotetaan OPEN-viesti naapurilta, reititin viimeistelee BGP-alustuksensa, lähettää naapurille OPEN- ja KEEPALIVE-viestit sekä muuttaa tilakseen OpenConfirm. (Mts. 54-58.)

Reititin voi joutua OpenSent- tai Connect-tilasta TCP-yhteyden katketessa Active-tilaan, jossa se koittaa muodostaa TCP-yhteyttä uudestaan. Mikäli yhteys saadaan muodostettua, toimii reititin kuten Connect-tilassakin ja muuttaa tilansa OpenSent. Mikäli TCP-yhteys saadaan muodostettua, mutta naapuri ehtii lähettää OPEN-viestin ensin, siirtyy BGP tarvittavien toimenpiteiden jälkeen tilaan OpenConfirm. Jos TCP-yhteys epäonnistuu uudestaan, siirtyy BGP takaisin Idle-tilaan. (Mts. 56, 59-64.)

OpenSent-tilassa oleva BGP odottaa OPEN-viestiä naapuriltaan. Viestin vastaanottamisen jälkeen sen sisältö tarkastetaan ja mikäli viestissä ei ole virheitä ja sen arvot ovat hyväksyttäviä, reititin lähettää naapurilleen KEEPALIVE-viestin ja vaihtaa tilakseen OpenConfirm. Jos viestissä havaitaan virheitä, reititin lähettää naapurilleen NOTIFICATION-viestin, vapauttaa BGP-resurssit, sulkee TCP-yhteyden ja muuttaa tilakseen Idle. OpenConfirm-tilaan päässyt BGP-prosessi odottaa joko yhteyden sulkevaa NOTIFICATION-viestiä tai naapuruuden varmistavaa KEEPALIVE-viestiä. KEEPALIVE-viestin vastaanottaminen muuttaa BGP:n tilaksi Established. (Mts. 63-65, 67-70.)

Established-tila mahdollistaa BGP-naapureiden välisen kommunikoinnin käyttäen UPDATE-, NOTIFICATION- ja KEEPALIVE-viestejä. UPDATE-viestillä voidaan lisätä, poistaa ja muokata mainostettuja reititystietoja. Se myös nollaa HoldTimer-laskurin, jonka täyttyessä naapuruus purkautuisi ja tilakoneprosessi tulisi aloittaa alusta. KEEPALIVE-viestiä käytetään ainoastaan kyseisen laskurin nollaamiseen. NOTIFICATION-viesti mahdollistaa BGP-naapuruuden purkamisen virhetilanteen sattuessa. Lisäyksenä aiempiin naapuruuksien katkaisemisiin Established-tilasta Idle-tilaan siirtyessä tulee naapurilta opitut reitit poistaa reititystaulusta. Tilakoneen toiminnan pääpiirteet löytyvät kuvattuna kuviosta 5. (Mts. 71-74.)



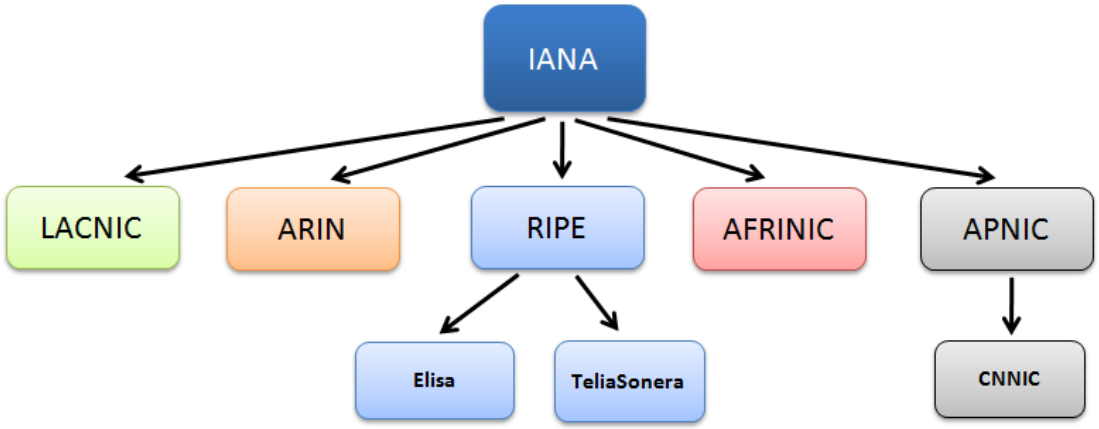
Kuvio 5. BGP:n tilakone yksinkertaistettuna (Huawei - Feature Description 2014.)

3 BGP:N TIETOTURVALAAJENNUKSET

3.1 Resource Public Key Infrastructure

Tässä osiossa käsitellään BGP-reitityksen turvallisuuden parantamiseksi suunniteltua infrastruktuuria, jonka rakennuspalikkoina toimivat Resource Public Key Infrastructure (RPKI), digitaalisesti allekirjoitetut reititysobjektit sekä jaettu repositoriojärjestelmä, jossa kyseisiä objekteja säilytetään. Se mahdollistaa toimijan mainostaa omistamaansa IP-osoitealuetta siten, että tiedon paikkansapitävyys voidaan yksiselitteisesti varmistaa. Samoin osoitealueen haltija voi valtuuttaa muita AS-alueita mainostamaan kyseistä verkkoaluetta. Infrastruktuuri on suunniteltu siten, että sen toiminnalisuutta voidaan hyödyntää muissa BGP:n turvallisuusprotokollissa kuten esimerkiksi tämän dokumentin tutkimissa Secure BGP:ssä ja Secure Origin BGP:ssä. (Lepinski & Kent 2012, 3.)

Käyttöönoton helpottamiseksi infrastruktuuri peilaa PKI-hierarkiansa käytössä olevasta IP- ja AS-alueita allokoivasta hierarkiasta, joka löytyy kuviosta 6. Tämän osion sisällä sanalla resurssi viitataan IP-osoitealueisiin ja AS-alueisiin. Hierarkian juuritasona toimii IANA, joka on IP- ja AS-alueiden alkuperäinen myöntäjä. Sen alla vaikuttaa viisi Regional Internet Registery-toimijaa (RIR), joiden tehtävä on allokoida IANA:n jakamia resursseja omalla määrätyllä geopoliittisella alueellaan. Hierarkian alimmalla tasolla toimivat kansalliset National Internet Registryt (NIR) sekä internetpalveluntarjoajat (ISP), jotka voivat jakaa RIR:ltä saamiaan resursseja asiakkailleen tai muille toimijoille. Kuvailtua hierarkiaa voidaan hyödyntää suoraan PKI-rakenteessa, jonka tarkoituksena on X.509-sertifikaattien avulla osoittaa resurssien omistajuussuhteita. Tätä erikoistunutta PKI-hierarkiaa kutsutaan Resource Public Key Infrastructureksi. (Mts. 3, 5.)

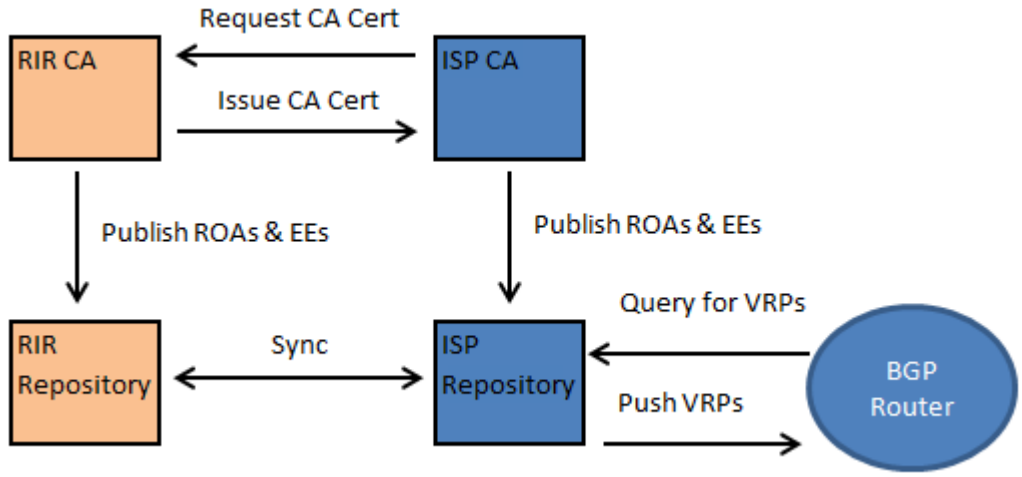


Kuvio 6 Resursseja allokoiva hierarkia

RPKI:n sertifikaatteja kutsutaan resurssisertifikaateiksi ja ne osoittavat, että sertifikaatin varmentaja on allokoinut tietyn verkkoalueen käyttöoikeuden sertifikaatin kohteelle sitoen sen joko kohteen AS-alueeseen tai IP-osoitteeseen. Certificate Authority-sertifikaattien (CA) tehtävä on pelkästään osoittaa resurssin valtuutuksesta eivätkä ne ota kantaa sertifikaatin kohteen identiteettiin. RPKI-hierarkiassa CA-sertifikaatti tulee löytyä jokaiselta taholta, jonka tarvitsee allokoida resursseja toiselle taholle, sillä sitä käytetään allokoinnin valtuuttavien End-Entity-sertifikaattien (EE) myöntämiseen. Käytännössä tämä tarkoittaa pääsääntöisesti IANA:aa, RIR:ejä ja NIR/ISP:itä. Poikkeuksena ovat montaa palveluntarjoajaa käyttävät multi-homing-asiakkaat, jotka tarvitsevat CA-sertifikaatin valtuuttaakseen kaikki käyttämänsä palveluntarjoajat mainostamaan osoiteavaruuttaan. (Mts. 6-7.)

Kuviossa 7 on kuvattuna esimerkki Resource Public Key Infrastructure -ympäristössä tapahtuvasta resurssisertifikaattien julkaisusta. Palveluntarjoaja (ISP) haluaa allokoida omistamiansa resursseja, minkä takia sen tulee pyytää alueelliselta RIR-toimijalta CA-sertifikaattia. Myöntämisen jälkeen ISP:n on mahdollista julkaista omia EE-sertifkaattejaan allekirjoittaen ne CA-sertifikaatilla. EE-sertifikaatit puolestaan toimivat perustana verkkoalueen ja AS-alueen välisen sidoksen muodostaville ROA-objekteille. Tarvittavien EE- ja ROA-tietueiden luomisen jälkeen ISP lataa ne repositorioonsa muiden käytettäviksi. Tämän jälkeen kyseinen repositorio ja muut RPKI-

repositoriot synkronoivat tietonsa, minkä jälkeen kaikilla toimijoilla on tieto uusista sertifikaateista. Lopuksi palveluntarjoajan BGP-reititin voi kysellä voimassa olevien ROA-objektien tiedot repositoriolta Validated ROA Payload -muodossa (VRP). Palveluntarjoajan ei tarvitse ylläpitää omaa CA-sertifikaattiaan tai julkaista itse ROA-objektejaan vaan se voidaan ulkoistaa RIR-toimijalle (ARIN - Resource Public Key Infrastructure n.d..) (Mts. 9-11.)



Kuvio 7. RPKI-ympäristön toiminta

CA-sertifikaattien käytössä on muutama normaalista PKI-ympäristöstä eroava käytäntö. Kohteen Distinguished name-tieto on sertifikaatin varmentajan päätettävissä ja sen arvoksi suositellaan tällä ennestään käytössä olevaa viittausta kohteeseen kuten asiakasnumeroa. Tarkoituksena on helpottaa RPKI:n integroimista nykyiseen infrastruktuuriin. Tieto ei saa yrittää kuvata kohdetta helposti tunnistettavaksi, mikä vähentää sertifikaattien mahdollista väärinkäyttöä, sillä kohde ei voi vakuuttaa olevansa jokin toinen taho sertifikaatin nimen perusteella. Distinguished name-tiedon tulee sisältää CommonName-attribuutti ja se voi sisältää serial-attribuutin. Varmentajan myöntämien CA-sertifikaattien sisältämän CommonNamen tulee olla jokaisen kohteen osalta ainutlaatuinen, mutta ei globaalisti ainutlaatuinen. Yksittäiselle toimijalle voidaan allokoida resursseja monien toisten toimijoiden toimesta, jolloin sille

myönnetyissä CA-sertifikaateissa kohteen nimi voi olla eri. Tahot voivat päättää keskenään monellako sertifikaatilla jokin osoitealue valtuutetaan kohteelle. (Mts. 6-7.)

End-Entity-sertifikaatteja (EE) käytetään pääsääntöisesti vahvistamaan siihen liittyvien allekirjoitettujen objektien ja siten, niissä mainittujen resurssien oikeellisuus. Tämän dokumentin puitteissa noita objekteja ovat Route Origin Authorizationit (ROA) ja manifestit. EE-sertifikaatilla luodaan yksi-yhteen sidos kyseisen sertifikaatin ja yhden allekirjoitetun objektin välille. Yhdellä EE-sertifikaatin salaisella avaimella allekirjoitetaan vain yksi objekti ja vain yhtä avainta käytetään objektin allekirjoittamiseen. Koska salaista avainta ei sen ainoan käyttökerran jälkeen tarvita, voidaan se tämän jälkeen tuhota, mikä puolestaan yksinkertaistaa salaisten avainten hallintaa. Samalla mahdollistetaan manifestien ja ROA-objektien kumoaminen EE-sertifikaatin kumoamisella CRL-listaa käyttäen. (Mts. 4, 7-8.)

RPKI-hierarkian Trust Anchor-kohteina (TA) toimivat IANA ja/tai viisi alueellista RIR-organisaatiota, mutta jokaisella toimijalla on oikeus päättää TA:nsa itse. Tällä mahdollistetaan toimijoiden, kuten privaattiosoitteita verkossaan käyttävien palveluntarjoajien, käyttää verkossaan omaa palvelinta, joka pystyy valtuuttamaan resurssijakoja. Näin RPKI-hierarkiaa voidaan täysin hyödyntää toimijan AS-alueen sisällä eikä vain virallisesti allokoitujen resurssien kanssa. Sisäisessä allokoinnissa tehdyt virheet vaikuttavat vain kyseisen AS-alueen toimivuuteen, koska palvelin ei ole muille toimijoille Trust Anchor-kohde. (Mts. 8.)

Route Origin Authorization -objekteja (ROA) käytetään AS-numeroiden ja IP-prefiksien väliseen sidokseen. Ne liittyvät aina johonkin EE:hen ja niiden allekirjoittamiseen käytetään EE:n julkista avainta vastaavaa salaista avainta. Lepinski ja Kent (2012, 19) kuvaavat, kuinka EE-luodaan vartavasten yksittäistä ROA-objektia varten. Yhden ROA:n avulla voidaan yksiselitteisesti valtuuttaa yksi AS-alue (asID) mainostamaan yhtä tai useata prefiksiä (address). Lisäksi voidaan määrittää, kuinka tarkkaa prefiksiä (maxLength) kyseinen AS-alue saa mainostaa. Esimerkiksi prefiksimerkintä 192.168.0.0/24 lisämääreellä maxLength 25 sallii verkkojen 192.168.0.0/24 ja

192.168.0.0/25 mainostamisen, mutta ei 192.168.0.0/27. Mikäli lisämäärettä ei ole mainittu, saa AS-alue mainostaa vain tarkalleen kyseistä prefiksiä. Alla näkyy ROA-tietueen rakenne (Lepinski, Kent & Kong 2012, 3-5.):

```
RouteOriginAttestation ::= SEQUENCE {
  version [0] INTEGER DEFAULT 0,
  asID  ASID,
  ipAddrBlocks SEQUENCE (SIZE(1..MAX)) OF ROAIPAddressFamily }

  ASID ::= INTEGER

  ROAIPAddressFamily ::= SEQUENCE {
    addressFamily OCTET STRING (SIZE (2..3)),
    addresses SEQUENCE (SIZE (1..MAX)) OF ROAIPAddress }

  ROAIPAddress ::= SEQUENCE {
    address IPAddress,
     maxLength INTEGER OPTIONAL }

  IPAddress ::= BIT STRING
```

Manifestit ovat ROA:n tavoin allekirjoitettuja objekteja samoilla riippuvuussuhteilla EE-sertifikaattiin. Yksi manifesti pitää sisällään listauksen yhden repositorion kaikista repositoriossa olevista allekirjoitetuista objekteista (pl. manifesti) kyseisellä ajanhetkellä. Listaukseen kuuluvat CRL-listat, ROA:t sekä EE- ja CA-sertifikaatit. Repositorion haltijan on luotava uusi manifesti aina, kun repositoriossa oleva sisältö muuttuu tai kun manifestossa määritetty seuraava päivitysajankohta täyttyy. Uuden manifestin luonti kasvattaa sen juoksevaa sarjanumeroa. Aina manifestin päivittyessä tulee siihen liittyvän CA-sertifikaatin varmentajan julkaista uusi CRL-lista, joka kumoaa vanhaan manifestiin liittyvän EE-sertifikaatin. (Lepinski & Kent 2012, 15-16.)

Infrastruktuurin repositorioita ylläpitävät IANA, RIR:t sekä NIR:t ja ISP:t ja niistä löytyvät sertifikaatit, allekirjoitetut resurssiobjektit sekä CRL-listat. Niitä hyödyntääkseen RPKI-hierarkian alimmalla tasolla olevat ISP hankkii itselleen ensin kaikki ROA-objektit, jotka valtuuttavat jonkin AS-alueen mainostamaan verkkoalueita. ROA:n vahvistamiseksi sen tulee seuraavaksi ladata kaikki sertifikaatit ja CRL-listat. Tämän

jälkeen sillä on paikallinen kopio repositorioiden tiedoista. Repositorioiden idea on taata, että kyseiset tiedot ovat saatavilla niistä riippuville tahoille jokaisena ajankohtana. Vähimmäisvaatimus on, että repositorio pitää sisällään kaikki kyseisen toimijan allekirjoittamat CA- ja EE-sertifikaatit, CRL-listat ja manifestit. Palveluntarjoajilla lisävaatimuksena ovat myös ROA-objektit. Suotavaa olisi, että toimijoiden repositoriot sisältäisivät edellä mainitut tiedot myös heidän asiakkailtaan ja asiakkaiden asiakkailta. Samoin alueellisten RIR-toimijoiden repositorioista löytyisi parhaassa tapauksessa kaikki sen oman geopoliittisen alueen toimijoiden RPKI-objektit. Digitaalisten allekirjoitusten myötä repositorion objekteja hakevat toimijat pystyvät havaitsemaan, mikäli tietoja on peukaloitu. Tarkoitus olisi lopulta estää oikeuttamattomat tiedonmuutokset erillisellä pääsynvalvonnalla, jolloin vain objektien alkuperäiset tuottajat voisivat lisätä, muokata ja poistaa niitä. (Mts. 11-13, 15.)

Prefix Origin Validation

Aiemman osion infrastruktuurin käyttämien objektien hyödyntämiseksi on kehitetty säännöstö, jonka mukaan reitityspäätöksiä voidaan tehdä. Aluksi jokainen RPKI-hierarkiaa hyödyntävä BGP-reititin lataa muistiinsa oman AS-alueensa käyttämän paikallisen repositorion ROA-objektit, jotka pitävät sisällään IP-osoitteen, prefiksin pituuden, prefiksin maksimipituuden (maxLength) sekä alkuperäisen AS-numeron. Näitä tietoja kutsutaan reitittimellä Validated ROA Payloadiksi (VRP). (Mohapatra, Scudder, Ward, Bush, Austein 2013, 3-4.)

BGP:n UPDATE-viestissä saatu reittitieto voi saada tarkistusten jälkeen kaksi määritelmää (Mts. 5.):

- Covered. Mainostuksessa oleva prefiksi on tarkempi tai yhtä tarkka kuin jossain VRP:ssä mainittu prefiksi ja niiden bitit ovat identtisiä VRP:n prefiksin tarkkuuteen asti.
- Matched. Mikäli mainostuksessa oleva reitti on arvoltaan Covered ja sen prefiksi on epätarkempi tai yhtä tarkka kuin VRP:ssä oleva maksimipituus **ja** reitin

alkuperäisen mainostajan AS-alue (AS_PATH-kentän oikeanpuoleisin arvo) on identtinen VRP:ssä olevaan AS-alueeseen.

Näitä määritelmiä hyödyntämällä voidaan jokainen saatu reittitieto jakaa yhteen kolmesta tilasta (Mts. 5.):

- NotFound. Yksikään VRP ei anna reitille määritelmää Covered.
- Valid. Ainakin yksi VRP antaa reitille määritelmän Matched.
- Invalid. Ainakin yksi VRP antaa reitille määritelmän Covered, mutta yksikään ei anna määritelmää Matched.

Tämä tarkastus tulee tehdä jokaiselle UPDATE-viestin reitille erikseen. Tila-arvoja voidaan käyttää suosimaan reittitietoja, joiden alkuperä on vahvistettu tai hylkäämään reitit, joiden kohdalla tarkistus palautti arvon Invalid. Loppujen lopuksi päätösvalta toiminnasta jää BGP-reitittimen paikalliset säännöt luoneelle taholle. (Mts. 7.)

3.2 Secure BGP

Vuonna 1997 BBN Technologies aloitti Secure BGP:n (S-BGP) kehittämisen, jonka tarkoitus on ottaa kantaa BGP:n haavoittuvuuksiin. Haavoittuvuudet mahdollistavat maailmanlaajuisiin reitityspäätöksiin vaikuttavat hyökkäykset. Näitä haavoittuvuuksia käsitellään tarkemmin dokumentin osiossa 4. Jotta BGP toimisi halutulla tavalla, tulee sen käyttämien UPDATE-viestien saapua naapurireitittimelle muuttumattomina ja ajallaan. Naapurilla tulee myös olla oikeus mainostaa kyseisiä reittejä. Vastaanottavan BGP-reitittimen tulee käsitellä saadut reitityspäivitykset BGP-määritysten ja paikallisten sääntöjen mukaisesti. (Kent n.d.)

Naapurireitittimien vaihtamien BGP-viestien eheys ja autenttisuus voidaan varmistaa muodostamalla niiden välille IPsec-yhteys ja kierrättämällä BGP-protokollan viestit salattuna tunnelin läpi ESP-protokollaa hyödyntäen. UPDATE-viestien sisällön eli reititystiedon oikeellisuuden tarkistaminen jää oletuksena vastaanottavan reitittimen

harteille. Se voi verrata saatua tietoa ennalta määrättyihin sääntöihin ja tehdä reititystaulua koskevat päätökset niiden pohjalta. Siitäkään huolimatta reititin ei voi pitää varmana, että BGP-naapurilta saatu tieto pitää paikkaansa, sillä tämä voisi muodostaa helposti kaadettavan luottamuksen ketjun. UPDATE-viestin reititystietojen oikeellisuus perustuu seuraaviin kysymyksiin (Mt.):

- saako mainostava BGP-reititin todellisuudessa mainostaa kyseistä reittiä sen alkuperänä olevan AS-alueen puolesta?
- onko mainostavaa reititintä edeltävällä AS-alueella kyseinen oikeus?
- onko reittien alkuperänä olevalla reitittimellä oikeus mainostaa kyseisiä reittitietoja?
- reittien poistamisen ollessa kyseessä, onko niiden poistosta ilmoittava reititin alun perin mainostanut kyseisiä reittitietoja?

S-BGP:n toiminta perustuu PKI-rakenteeseen, jolla varmistetaan prefiksien ja AS-alueiden omistajuussuhteet, IPsec-yhteyksiin ja AS-polun todentamiseen. Ajateltu PKI-rakenne käyttäisi X.509v3-sertifikaatteja, joiden avulla reitittimet voisivat varmistaa toisten AS-alueiden reitittimien valtuudet. Sertifikaattien avulla reitittimet voivat myös tarkistaa omistaako kyseinen AS-alue mainostetut reitit. Käyttöönoton helpottamiseksi PKI:n infrastruktuuri tulisi vastaamaan nykyään käytössä olevaa verkko- ja AS-alueiden myöntämiseen liittyvää organisaatiorakennetta. Täten ylimpänä tahona tulisi olemaan Internet Assigned Numbers Authority (IANA), jonka alapuolella toimivat alueelliset järjestöt eli Regional Internet Registry (RIR) -elimet. RIR tulisi myöntämään sertifikaatit AS- ja verkkoalueita käyttäville palveluntarjoajille. Jokainen sertifikaatti pitää sisällään sekä palveluntarjoajan omistamat AS-alueet että verkkoalueet luoden näiden välille sidoksen. Palveluntarjoajat puolestaan pystyvät myöntämään sertifikaatit käyttämilleen BGP-reitittimille valtuuttaen edustamaan kyseistä palveluntarjoajaa ja sen hallinnoimia alueita. (Mt.)

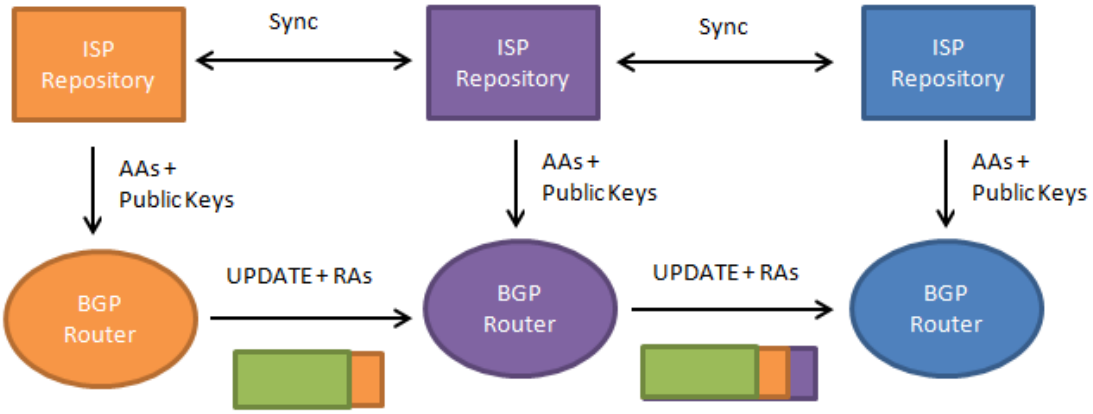
S-BGP käyttää digitaalisesti allekirjoitettuja todistuksia, joilla varmennetaan, että mainostavalla AS-alueella on oikeus mainostaa verkkoalueelle vievää polkua (AS-

PATH) sekä annetaan naapureille lupa mainostaa saatuja polkuja eteenpäin. Address Attestation (AA) on palveluntarjoajan allekirjoittama todistus, jonka se jakaa haluamilleen AS-alueille antaen niiden toimia reittien alkuperäisenä mainostajana. Route Attestation (RA) on puolestaan palveluntarjoajan sertifioiman BGP-reitittimen allekirjoittama todistus, jonka se lähettää UPDATE-viestissä naapurireitittimille. Sitä kuljetetaan uudessa vapaavalintaisessa transitiivisessa Path Attribute-kentässä. Se antaa kyseisille laitteille luvan mainostaa edelleen viestissä opittuja reittejä. Koska RA-todistus toimitetaan aina reittitietojen mukana, ei voi syntyä tilannetta, jossa todistus ja reittitiedot olisivat eri ajankohdilta. (Mt.)

Toimiakseen järjestelmä tarvitsee AA-tietueen jokaista eri alkuperää olevaa prefiksiä kohden sekä alkuperäisten palveluntarjoajien julkisen avaimen. Lisäksi AS-polun jokaisen autonomisen järjestelmän RA-tietue tulee löytyä UPDATE-viestistä sitä vastaavan palveluntarjoajan julkisen avaimen kera. Näiden todistusten avulla voidaan missä välissä ketjua tahansa tarkastaa, että reittiä alun perin mainostava taho on oikea ja AS-PATH-polussa mainitut AS-alueet saavat levittää kyseistä tietoa. AA-tietuetta ei tarvita reittejä poistettaessa tarkistamaan mainostiko kyseinen reititin alun perin reittiä, sillä tarkistus voidaan suorittaa BGP-naapureiden välisen IPsec-yhteyden ja naapurikohtaisten Adj-RIB-In-reititystietokantojen avulla. IPsec-tunnelia käyttäen voidaan varmistua siitä, että tieto ei ole muuttunut matkan varrella ja reititystietokannasta löytyy alkuperäinen mainostaja. Jotta reititin voi tehdä tarvittavat tarkistukset, tulee sillä olla pääsy julkisiin avaimiin sekä AA-tietueisiin. (Mt.)

BGP:n UPDATE-viestiä käytetään S-BGP:n tuomiin lisätarpeisiin vain RA-todistuksen lähettämisessä. AS-alueiden omistamat verkkoalueet harvoin muuttuvat, joten AA-todistuksen kuljettaminen jokaisessa viestissä veisi turhaan osuuden UPDATE-viestin 4096 tavun maksimikoosta. Tieto tulisi myös useaan kertaan monelta eri naapurilta, vaikka tiedon arvo ei olisi välttämättä muuttumassa lähiaikoina. AA-todistuksien jakamiseen ja säilömiseen käytetään repositorioita. Lopullisessa toteutuksessa näitä ylläpitävät joko RIR:t tai isoimmat operaattorit sekä Internet Exchange Point -toimijat (IXP), jotka voivat jakaa tietoja eteenpäin pienemmille operaattoreille. (Mt.)

Repositoryt tulevat määräaikaisesti päivittämään keskenään tietokantojaan muuttuneiden ja uusien sertifikaattien, AA-tietueiden ja CRL-listojen (Certificate Revocation List) osalta. Jokainen operaattori lataa määräajoin kaikkien S-BGP:tä hyödyntävien operaattoreiden edellä mainitut tiedot repositoriolta. Operaattori käsittelee saamansa raakadatan ja muodostaa siitä reitittimille jaettavan tiiviimmän version, joka sisältää AA-tiedot, julkiset avaimet sekä AS ja reittitiedot. Repositorioiden lähettämät ja vastaanottamat viestit suojataan Secure Socket Layer -protokollalla (SSL). Järeämpää suojausta ei tarvita, sillä tiedot ovat digitaalisesti allekirjoitettuja ja niitä käytetään vain ajoittaiseen tietojen voimassaolon tarkistamiseen. Kuviossa 8 on havainnollistettu, kuinka repositoriot jakavat tiivistetyt resurssitiedot reitittimille. Samalla voidaan huomata, kuinka RA-tietoa kuljetetaan itse UPDATE-viestissä. (Mt.)



Kuvio 8. Secure BGP:n toiminta

RA-tietoa tultaisiin kuljettamaan uudessa valinnaisessa transitiivisessa UPDATE-viestin Attestation Path Attribute -kentässä (ATTEST). BGP:n UPDATE-viestin rakenne löytyy tämän dokumentin kuviosta 3. Mikäli AS-PATH-tietoa muokataan reittitietoja yhdistäessä, tulee reitittimen kuljettaa RA-tieto viestin mukana, vaikka varsinainen AS-tieto olisi hävinnyt reittejä yhdistettäessä. Vastaanottava reititin toimii saatujen reittitietojen kanssa normaalisti, mutta sen tulee tämän lisäksi tallentaa saadut RA-tiedot omaan BGP-reititystauluunsa (Adj-RIBs-In, Loc-RIB tai Adj-RIBs-Out). (Lynn, Mikkelson & Seo 2003 , 12, 15, 22.)

3.3 Secure Origin BGP

Secure Origin BGP (soBGP) on suunniteltu takaamaan, että BGP-reitin alkuperäisellä mainostajalla on itse asiassa oikeus mainostaa kyseistä reittiä. Lisäksi se takaa, että reittiä eteenpäin mainostavalla reitittimellä on reitti mainostettuun verkkoon ja että tällä reitittimellä on oikeus mainostaa mainittua verkkoa. Mainostuksen vastaanottavan reitittimen tulee myös hyväksyä paikallisten sääntöjensä rajoissa reitin ja AS-polun sisältämät tiedot. (White n.d.)

Secure Origin BGP:n toiminta perustuu EntityCert-sertifikaattiin, jolla luodaan AS-numeron ja julkisen avaimen (ja sitä myötä salaisen avaimen) välinen sidos. Näin saadaan taattua, että julkiseen avaimeen ja sen alkuperään voidaan luottaa. Jokainen AS-alue luo sertifikaattinsa käyttäen X.509v3-standardia. Sertifikaattia pyytävä AS-alue ilmoittaa sertifikaatin myöntävälle järjestelmälle AS-numeronsa ja julkisen avaimensa. Myöntävä taho tarkistaa tiedot ja tämän jälkeen kyseiset tiedostot tullaan allekirjoittamaan kyseisen osapuolen salaisella avaimella niiden oikeellisuuden varmistamiseksi. (Mt.)

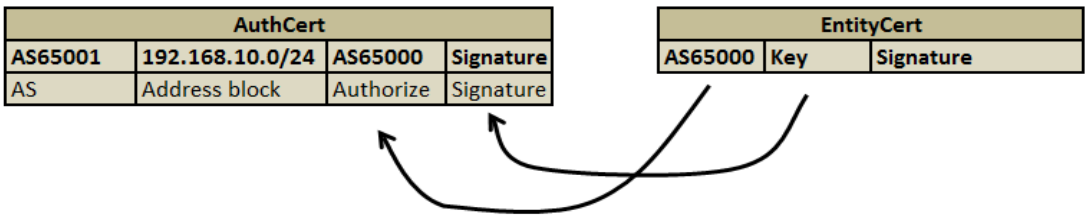
Allekirjoituksen jälkeen syntynyttä tiedostoa kutsutaan EntityCert-sertifikaatiksi, joka lähetetään sitä pyytäneelle autonomiselle järjestelmälle. Tarkoituksena on aluksi allekirjoittaa ulkoisia menetelmiä käyttäen isoimpien toimijoiden sertifikaatit, jotka voivat puolestaan allekirjoittaa luottamiensa tahojen sertifikaatit muodostaen yhä uusia AS-alue-sertifikaatti-siteitä. Lopulta kaikilla AS-alueilla tulisi olla toisen toimijan allekirjoittama sertifikaatti, jonka aitous voidaan tarkistaa. Tarkistamiseen käytetään allekirjoituksen ja muiden tietojen varmistamista myöntäneeltä taholta. AS-alueiden tulee pitää salainen avaimensa piilossa, sillä sitä tarvitaan uusien allekirjoitusten muodostamiseen. Kuviossa 9 on EntityCertin pelkistetty rakenne. (Mt.)

EntityCert		
AS	Key	Signature

Kuvio 9. EntityCertin rakenne

Kun jokaisen AS-alueen allekirjoitus on saatu varmistettua, voidaan sitä hyödyntää reittitietojen oikeellisuutta todistavassa Authorization Certificate -sertifikaatissa (AuthCert). Mikäli AS-alue haluaa valtuuttaa toisen AS:n mainostamaan sen hallinnoimaa verkkoa alkuperäismainostajana, se luo AuthCert-serfitikaatin. Se pitää sisällään mainostavan AS-alueen, mainostetun verkkoalueen, valtuuttavan AS-alueen sekä valtuuttavan AS-alueen allekirjoituksen. Huomionarvoista on, että toisin kuin BGP:ssä, AuthCert määrittää verkkoalueen eikä yksittäistä prefiksitietoa. Näin ollen yksi verkkoalue voi pitää sisällään monia prefiksejä vähentäen tarvittavien sertifikaattien määrää. (Mts.)

Kuviossa 10 näkyy yksinkertaistettuna, kuinka AS65000 valtuuttaa AS65001:n mainostamaan osoitealuetta 192.168.10.0/24. Tässä tapauksessa AS65000:n tulee joko omistaa kyseinen verkkoalue tai olla puolestaan valtuutettu mainostamaan sitä. Valtuuttava AS suojaa AuthCert-tiedoston kentät omalla salaisella avaimellaan tiivistesummafunktiota hyödyntäen. Sen jälkeen se liittää saadun tuloksen sertifikaatin allekirjoitukseksi. Kuviosta käy ilmi myös kenttien nimitykset ei-lihavoituna. (Mts.)



Kuvio 10. AuthCert-sertifikaatin rakenne

Mikäli AuthCertin tietojen oikeellisuus halutaan tarkastaa, tulee tarkastajan tunnistaa käytetty EntityCert ja verrata sen yksilöllistä sarjanumeroa AuthCertissä käytettyyn

sarjanumeroon sekä tarkistaa EntityCertin oikeellisuus. Mikäli EntityCertin oikeellisuutta ei voida vahvistaa, hylätään myös AuthCert. Lopuksi sen tulee käyttää EntityCertissä olevaa julkista avainta allekirjoituksen salauksen purkamiseen, minkä jälkeen laskettua tiivistesummaa verrataan salattuna olleeseen hashiin. AuthCert hyväksytään vain mikäli arvot täsmäävät. (Weis 2006, 15-16.)

Jokaiselle prefiksille ja osoitealueelle voidaan sitä mainostavan tahon toimesta luoda sääntölista PrefixPolicyCert-sertifikaatilla, jolla voidaan määrittää esimerkiksi AS-alueet, joita ei saa ottaa mukaan AS-polkuun. Mikäli näin tehdään, niin AuthCert tulee sisällyttää tähän sertifikaattiin. Myös PrefixPolicyCert suojataan myöntävän tahon salaisen avaimen avulla tehdyllä allekirjoituksella ja salauksella. PrefixPolicyCertin vastaanottavalla taholla ei ole pakotetta ottaa sen arvoja huomioon. (White n.d.)

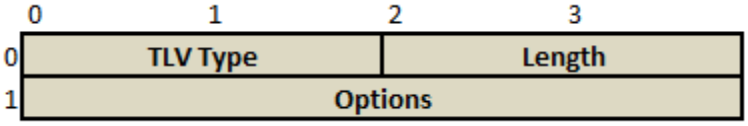
Jotta soBGP voi taata sen, että reittejä mainostavilla reitittimillä on reitti mainostamaansa verkkoon, luodaan siinä topologiakartta käyttäen hyväksi ASPolicyCert-sertifikaattia. Kyseisen sertifikaatin luo AS-alueen hallinoima taho ja niitä voi olla voimassa vain yksi kerrallaan per AS-alue. Se pitää sisällään tiedon kaikista AS-alueista, joihin kyseinen autonominen järjestelmä on suoraan yhteydessä. Lisäksi siinä voidaan määrittää TRANSIT- ja NON-TRANSIT-liikennöintisäännöt, joilla voidaan kertoa, minkä naapurialueiden läpi liikennöidään. NON-TRANSIT-alueet toimivat soBGP-naapureina, mutta niitä ei käytetä hyötyliikenteelle. Mikäli kaikki verkon soBGP:tä hyödyntävät AS-alueet ovat yhteydessä toisiinsa, syntyy lopputuloksena yksi topologiakartta, muussa tapauksessa syntyy useita erillään olevia karttoja. (Weis 2006, 22; White 2006, 3-4.)

Kartan rakentaminen voidaan aloittaa mistä tahansa ASPolicyCertin omaavasta AS-alueesta ja merkitä mihin alueisiin se on sertifikaattinsa mukaan yhteydessä. Tämän jälkeen jokaisen väitetyn naapurijärjestelmän sertifikaatista tarkistetaan löytyykö aloituspisteen AS-alue mainittuna. Mikäli tieto naapuruudesta löytyy molemmilta AS-alueilta, tulee ne lisätä topologiakarttaan ja merkata yhteys luotetuksi. Mikäli yhteys on vain toisinpäin voimassa, voidaan tieto halutessaan lisätä silti karttaan merkaten

se alemmalla luotettavuustasolla. Tämän jälkeen kummassa tahansa sertifikaatissa olevat liikennöintisäännöt otetaan käyttöön. Prosessia toistetaan, kunnes kaikki soBGP-toimijat on käyty läpi ja sen hetkinen topologiakartta on saatu hahmotettua. (White 2006, 3-4.)

Sertifikaattien lähettämiseen soBGP:n osalta on ehdotettu uutta BGP:n viestityyppiä nimeltään SECURITY. BGP:n otsikossa Type-kentän arvona toimii kuusi ja sen käytöstä tulee sopia OPEN-viestin Optional Parameters-kentässä tällä hetkellä määrittelemättömällä arvolla. Tähän mennessä on määritetty vasta SECURITY Option TLV sekä Request TLV. SECURITY Option TLV:tä käytetään reitittimien paikallisten asetusten kertomiseen ja sen TLV Type-koodi SECURITY-viestissä on yksi ja Length aina arvoltaan kaksi. Molemmat kentät ovat kahden oktetin mittaisia. Viimeinen kenttä, Options, on 32-bitin eli neljän oktetin mittainen ja se mahdollistaa 32 eri asetuksen kertomisen. (Ng, J. 2004, 4-6.)

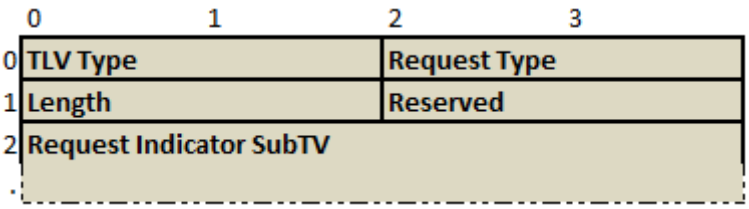
Mikäli bitti nolla on arvoltaan yksi, tulee BGP-naapurin lähettää ensin sertifikaatit tulevissa SECURITY-viesteissä ennen varsinaisia reittitietoja UPDATE-viesteissä. Mikäli sen arvo on nolla, tulee reittitiedot lähettää ensin. Tämä asetus mahdollistaa ylläpitäjän päättää halutaanko turvallisuus vai nopeus maksimoida naapuruuksia muodostaessa. Mikäli bitillä yksi on arvo, lähettää reititin vain jo vahvistettuja sertifikaatteja naapurilleen. Mikäli bitillä kaksi on arvo, sallii reititin vain vahvistettuja sertifikaatteja naapuriltaan. Huomionarvoista on, että bitit yksi ja kaksi ovat käytössä vain IBGP-naapuruuksissa. Näin alueen hallinnoija voi päättää, missä kohdin verkkoa vahvistukset tehdään. Kuviossa 11 näkyy kyseinen SECURITY-viesti. (Ng, J. 2004, 4-6.)



Kuvio 11. SECURITY Option TLV

Toinen käytettävistä viesteistä on Request TLV, jolla nimensä mukaisesti määritetään pyydettävät sertifikaatit. Kahden oktetin mittaisen TLV-kentän arvon tulee olla kaksi. Yhtä pitkän Request Type-kentän arvo kertoo pyydettävän sertifikaatin tyypin. Kaikkien näiden sertifikaattien tulee täsmätä Request Indicator-kentän arvoon. Request Type voi saada arvot: (1) Mikä tahansa sertifikaatti, (2) EntityCert, (3) ASPolicyCert tai (4) PrefixPolicyCert. Length-kenttä, kaksi oktettia, kertoo viestin kokonaispituuden. Seuraava kenttä on kahden oktetin mittainen ja se on varattu tulevaisuuden käyttötarkoituksiin. (Mts. 6-7.)

Viimeinen kenttä eli Request Indicator SubTV rajaa palautettavien sertifikaattien joukkoa ja se koostuu kahden oktetin Type-tiedosta ja sovitun mittaisesta varsinaisesta rajaustiedosta. Rajauksia voidaan tehdä (Type 1-6) valtuutetun AS-alueen numeron, valtuuttajan AS-numeron, IPv4-osoitteen, IPv6-osoitteen, alkavan sarjanumeron tai lopettavan sarjanumeron perusteella. Esimerkiksi tyypin kolme rajauksella IPv4-osoitetta pyydettäessä varsinaisen rajaustiedon pituus olisi neljä oktettia. Request Indicator SubTV -kenttiä voi olla monta yhdessä Request TLV -viestissä, jonka kehysrakenne löytyy kuviosta 12. Saman viestin SubTV-kenttien arvoja kohdellaan toisiinsa nähden vaihtoehtoisina, joten yhden täsmääminen riittää halutun sertifikaatin tunnistamiseksi. (Mts. 6-7 & 10.)



Kuvio 12. Request TLV

Sertifikaattien käsittely ja purkaminen voidaan halutessaan jättää joko AS-alueen reunareitittimille, erilliselle palvelimelle tai kolmannelle osapuolelle. Mitään pakotetta sertifikaattitietojen tarkistamiselle välittömästi niitä vastaanottaessa ei ole vaan se voidaan tehdä ajastetusti tai viiveellä. Reunareitittimien kohdalla jokainen reititin

rakentaa oman sertifikaattitietokantansa sen mukaan, kun se oppii niitä SECURITY-viestissä. Saman AS-alueen sisällä olevat reunareitittimet voivat vaihtaa IBGP-naapuruuksien avulla sertifikaattitietoja vahvistetuiksi merkattuina, mikäli ensimmäisessä SECURITY-viestissä niin sovitaan. Tällöin sertifikaattien tarkistukset tarvitsee tehdä vain kerran AS-alueelle tultaessa. Halutessaan kaikki reunareitittimet voidaan konfiguroida tarkastamaan vastaanottamansa sertifikaatit lähteestä huolimatta. (White 2006, 8-11; Ng 2004, 10.)

Vähemmän reitittimiä kuormittavaa on käyttää sertifikaattien tarkastamiseen erillistä palvelinta ja sitä varten on kehitetty jo erillisiä RADIUS-attribuutteja ja lisää SECURITY-viestejä. Niiden käyttöönotto lopullisessa soBGP-toteutuksessa ei ole varmaa, joten niitä ei käydä tämän teoksen puitteissa läpi. Tarkoituksena on, että jokaisella AS-alueella on valitsemansa määrä palvelimia sertifikaattien käsittelyyn, joita kyseinen alue ylläpitää. Tässä skenaariossa palvelin pitää tallessa kaikki EntityCertit ja reitittimen tehtäväksi jää lähettää sille olennaiset kentät muista sertifikaateista, jotka se haluaa varmistaa. Palvelin hoitaa laskennallisen työn, jonka jälkeen se palauttaisi myöntävän Access-Accept- tai kielteisen Access-Reject-viestin. Koska käytössä on BGP:n lisäosa, voidaan soBGP-palvelimien välille muodostaa naapuruussuhteita, mikä nopeuttaa tietokantojen kasvamista. (Lonvick 2004, 4; White 2004, 8-9.)

Kolmas vaihtoehto on ulkoistaa sertifikaattien tarkastaminen jollekin luotettavalle ulkopuoliselle taholle, kuten toiselle operaattorille. Toiminnaltaan tämä ei eroa aiemmasta vaihtoehdosta muuten kuin, että palveluntarjoaja ei tarvitse omaa kalustoaan, mutta on riippuvainen kolmannesta osapuolesta BGP-reittitietojensa oikeellisuuden tarkistamisessa. Mitä tahansa vaihtoehtoa käytetäänkin, tulisi sekä normaalit soBGP-naapuruudet että sertifikaattien varten olevat soBGP-naapuruudet suojata IPsecillä tai BGP MD5-autentikoinnilla. (White 2006, 7,11.)

3.4 Interdomain Routing Validation

Interdomain Routing Validation (IRV) on erillinen arkkitehtuuri, jonka tarkoituksena on suojata BGP:tä väärin konfiguroiduilta ja haitallisilta AS-alueilta. Sitä voidaan lisäksi hyödyntää reititykseen liittyvien konfiguraatio-ongelmien tunnistamisessa ja analysoinnissa. IRV on tarkoituksella kehitetty erilliseksi arkkitehtuuriksi, jotta sen toiminnallisuudet eivät vaikuta negatiivisesti olemassa olevan BGP-verkoston käyttöön. Sen tarkoitus on mahdollistaa AS-alueiden dynaamisten (mainostetut reitit) ja staattisten (säännöt) tietojen vahvistaminen haluttuna ajankohtana. Se antaa pohjan reittitietoihin liittyvien tietojen passiivisen hankkimisen ulkopuolisilta toimijoilta. Yksi lähtökohdista on myös mahdollistaa protokollan vaiheittainen käyttöönotto häiritsemättä IRV:hen kuulumattomia toimijoita. (Goodell, Aiello, Griffin, Ioannidis, McDaniel & Rubin 2003, 1,3.)

IRV-arkkitehtuurin toiminta perustuu erillisiin AS-aluekohtaisiin Interdomain Routing Validator -palvelimiin (IRV), joilta toisten AS-alueiden IRV-palvelimet voivat vahvistaa, onko kyseisellä AS-alueella oikeus mainostaa reittitietoja sekä pyytää muuta informaatiota. Jotta järjestelmä toimisi, tulee AS-alueiden tuntea jokaisen toisen AS-alueen IRV-palvelimen osoite. Eri AS-alueiden IRV-palvelinten IP-osoitteita jaetaan ja hallinnoidaan keskitetyissä rekistereissä. Tietoja voidaan jakaa joko DNS-kyselyllä tai HTTP-uudelleenohjauksella. Palvelimen tulee olla valtuutettu AS-alueen puolesta ja sen tulee pystyä jakamaan reititykseen liittyviä tietoja kyselyrajapintojen kautta. (Mts. 3-5.)

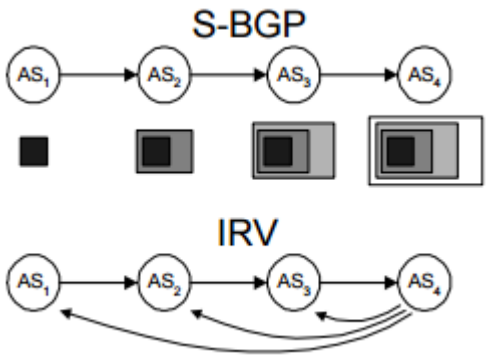
Lopulta eri AS-alueiden palvelimet muodostavat keskenään kyselyihin pohjautuvan verkoston. Kyselyillä voitaisiin hankkia AS-alueen tuonti- ja vientisäännöt sisältävät RPSL-tietueet, käytetyt BGP:n communityt, järjestelmän ylläpitäjien yhteystiedot sekä UPDATE-viesteillä vastaanotetut reittien lisäykset ja poistamiset. Viimeisimmällä tiedolla saadaan muodostettua kyseisen AS-alueen näkökulmasta tämänhetkinen reititystietoihin perustuva topologiakartta. Lisäksi AS-alueelta voidaan kysyä parhaillaan käytetyt UPDATE-viestien reittitiedot, joita se mainostaa naapureilleen ja verrata

niitä vastaanotettuihin UPDATE-viesteihin. Saatu tieto vastaa toiminnallisuudeltaan S-BGP:n Route Attestation (RA) -tietoa. Erona on tosin, että mainostetut reitit tulee kysyä jokaisen polulla olevan AS-alueen IRV-palvelimelta eikä niitä kuljeteta varsinaisessa UPDATE-viestissä. IRV-palvelin voidaan konfiguroida vastaamaan kyselyihin eri tavalla riippuen lähteenä olevan IRV-palvelimen AS-alueesta. (Mts. 3-4.)

Jotta IRV-palvelin saa tietoonsa autonomisen järjestelmän vastaanotetut ja lähetetyt UPDATE-viestit, tulee sen muodostaa IBGP-naapuruudet kaikkien oman AS-alueensa reunareitittimien kanssa. Vastaanotettujen UPDATE-viestien lisäksi IRV-palvelimen tulee tallentaa tietokantaansa viestin viimeksi lähettänyt BGP-reititin ja kyseisen reitittimen AS-alue. Mainostetut UPDATE-viestit käsittävät sekä IRV:n omalta AS-alueelta alun perin olevat mainostukset sekä sen edelleenmainostamat reitit. Mainostusten lisäksi palvelin pitää kirjaa, mille naapurialueille mainostukset on lähetetty. (Mts. 3-6, 9.)

IRV-protokollan tapauksessa AS-alueen ja sen omistamien prefiksien välinen sidos, joka voidaan tarvittaessa vahvistaa, toteutettaisiin digitaalisesti allekirjoitetuilla lausunnoilla. Näitä lausuntoja jakaisivat esimerkiksi IANA ja alueelliset RIR:t, sillä ne ovat alun perin jakaneet osoiteavaruudenkin ja antaneet verkkoalueita eri toimijoiden käyttöön. Lausuntojen tarkempaa koostumusta ei ole määritetty, vaan siinä mahdollisesti seurataan Secure BGP:n kehittäjien valitsemaa ratkaisua. (Mts. 7.)

Mikäli UPDATE-viestissä oleva AS-polku ja sen oikeellisuus halutaan tarkistaa, voidaan jokaiselta polulla olevalta AS-alueen IRV-palvelimelta kysyä kyseisen AS-alueen käyttämät tämänhetkiset UPDATE-viestit. Lisäksi reittitiedon alkuperäisen mainostajan eli AS_PATH-tietueen ensimmäisen AS-alueen oikeus mainostaa kyseistä reittiä voidaan tarkistaa edellisessä kappaleessa mainittua lausuntoa kysymällä. Alla olevassa kuviossa 13 käy ilmi, kuinka S-BGP:n tapauksessa UPDATE-viestiin lisätään RA-todistuksia jokaista AS-aluetta varten lisäten BGP-viestin kokoa. IRV:n tapauksessa BGP:n viestin koko ei kasva vaan tarkistuksen tekevän AS-alueen kyselyjen määrä kasvaa. (Mts. 4,8.)



Kuvio 13. Vertailu S-BGP:n ja IRV:n käyttämästä AS_PATH:in tarkastuksesta (Goodell ym. 2003, 8.)

Interdomain Routing Validator-palvelimet ovat käytännössä HTTP:ta käyttäviä web-palvelimia. Palvelimia tulee tietoturvasyistä käyttää vain IRV:n käyttöön. Ne käyttävät kyselyihin ja niiden vastauksiin XML-formaattia sen levinneisyyden ja modulaarisuuden vuoksi. XQuerya käytetään IRV-tietokantoihin tehtyjen kyselyiden salaamiseen. Palvelinten väliset yhteydet ja kyselyt suojataan, jotta käsitellyn datan autenttisuus, eheys ja ajallisuus voidaan varmistaa. Suojaukseen on ehdotettu joko TLS-, SSL- tai IPsec-ratkaisuja. (Mts. 6.)

4 HAAVOITTUVUUDET BGP-REITITYKSESSÄ

4.1 BGP- ja TCP-protokollien aiheuttamat ulkoiset haavoittuvuudet

BGP:tä ei ole suunniteltu sisältämään suojausmekanismeja tahallisia tai tahattomia virheitä vastaan, vaikka ne voivat aiheuttaa häiriöitä BGP-reitityksessä. Man-in-the-middle-hyökkäykset (MITM) ovat mahdollisia, koska BGP-naapureita ei todenneta mitenkään. Lisäksi BGP-protokolla ei itsessään tarjoa keinoja viestiensä muokkaamisen, poistamisen tai lisäämisen havaitsemiseksi, mutta on otettava huomioon, että BGP toimii TCP-protokollan päällä, joka puolestaan huolehtii kyseisistä uhkakuvista hyödyntäen esimerkiksi viestien sarjanumerointia. TCP:n suojausmekanismeista huolimatta viestien muokkausta ei voida huomata, mikäli muokatun viestin hyötykuorman pituus ja syntaksi täsmäävät alkuperäiseen. (Murphy 2006, 3, 6.)

Esimerkiksi TCP RST-viesteillä voidaan huijata BGP-reititin luulemaan, että BGP-naapuriin ei ole yhteyttä, joka puolestaan johtaa naapuruuden tiputtamiseen. Hyväksytty TCP RST johtaa TCP-session katkaisemiseen riippumatta session tilasta. Jotta kyseiset hyökkääjän lähettämät TCP-segmentit ovat hyväksyttyjä, tulee niiden sarjanumeroinnin täsmätä eli haitallisen segmentin sarjanumeron tulee olla yhden isompi kuin sitä edeltävän lähetetyn segmentin. Nykyisillä tietoliikennenopeuksilla oikean sarjanumeron arvaamiseen ja halutun BGP:n TCP-yhteyden tappamiseen kuluva aika on laskenut huomattavasti. Esimerkiksi 100 Mbps yhteydellä hyökkääjä tarvitsee 3436 RST-viestiä, joiden lähettäminen vie 10 millisekuntia. (Touch 2007, 7-9.)

Aiemmin mainittuja haavoittuvuuksia varten on kehitetty TCP-protokollan laajennus, joka lisää BGP-viestejä kuljettavaan TCP-segmenttiin 16-bittisen MD5-tiivistelmän. Lähettävä BGP-reititin laskee tiivistelmän BGP-viestin TCP:n kehysrakenteen, hyötykuorman ja ennalta sovitun avaimen avulla. Laskettu tiivistelmä kuljetetaan viestin mukana vastaanottajalle, jonka tehtäväksi jää laskea vastaanotetusta TCP-

segmentistä uusi tiivistelmä ja verrata sitä kehyksen sisältämään tiivistelmään. Mikäli tiivistelmät eivät täsmää vastaanottaja hylkää segmentin eikä tuota mitään vastausviestiä lähettäjälle. MD5-tiivistelmää käyttävien viestien väärentäminen vaatisi sekä sarjanumeroinnin että käytetyn avaimen murtamista. Murphy (2006, 3) huomauttaa, että vaikka BGP-toteutuksien tulee nykyään tukea MD5-tiivistelmän käyttöä, ei se tarkoita, että tuettua ominaisuutta käytettäisiin. (Heffernan 1998, 1-2.)

MD5-tiivistelmän käyttö estää BGP-viestien lisäämiseen, poistamiseen ja muokkaamiseen perustuvat hyökkäykset. Lisäksi se estää ulkopuolisten tekemät MITM-hyökkäykset todentamalla naapurin oikeellisuuden. BGP:n käyttämät viestit kuljettavat reititystiedot selkokielisinä naapurilta toiselle, joka mahdollistaa niiden salakuuntelun. Mikäli viestien luottamuksellisuus haluttaisiin turvata, voitaisiin siihen käyttää IPsecin ESP-protokollaa. (Murphy 2006, 6,19.)

4.2 Oikeutetun BGP-naapurin aiheuttamat haavoittuvuudet

4.2.1 Uhkakuvat

Väärennettyjä viestejä vastaan suojautuminen ei auta kuitenkaan, jos oikeutettu BGP-naapuri haluaa mainostaa vääriä reittitietoja tai lopettaa naapuruuden. Hyökkääjä voi käyttää merkittävimmin hyväkseen BGP:n UPDATE-viestissä olevia AS_PATH-, NEXT_HOP- ja ATOMIC_AGGREGATE-kenttiä. Se voi myös väärentää Network Layer Reachability Information-arvon. Lopputuloksena voi syntyä tavoittamattomissa olevia verkkoalueita tai epäoptimaalisesti reititettyä liikennettä. Väärin reititetty liikenne voi altistua myös salakuuntelulle. (Murphy 2006, 4, 13-16.)

AS_PATH-tiedon väärentämisellä naapuri voi vaikuttaa merkittävästi, kuinka UPDATE-viestissä mainittuihin prefikseihin liikennettä reitite tään. Mikäli vastaanottava reititin ei tarkista, että AS_PATH-tiedon ensimmäinen AS-alue kuuluu mainosta-

valle naapurireitittimelle, voi naapuri korvata koko polun haluamallaan tiedolla. Tarkistuksesta huolimatta naapurin on mahdollista lyhentää AS-polkua poistamalla siitä alueita, jolloin sen todennäköisyys tulla valituksi käytetyksi reitiksi nousee. Tämän jälkeen naapuri voisi saada haltuunsa sille kuulumatonta liikennettä. AS-polun lyhentäminen voi johtaa myös reitityssilmukoihin, koska silmukanestoa koitetaan tehdä vaillinaisella tiedolla. (Mts. 14.)

AS_PATH-tiedon muokkaamisella naapuri voi mainostaa olevansa jonkin prefiksin alkuperäinen mainostaja muokkaamalla oman AS-alueensa polun oikeanpuolimmaiseksi AS-alueeksi. Kyseisen UPDATE-viestin NLRI-tiedossa oleville verkkoalueille pyrkivä liikenne kiertää virheellisesti mainostavan AS-alueen kautta kaikkien mainostuksen hyväksyvien tahojen osalta. Tämän jälkeen mainostajalle jää valinta, mitä tehdä vastaanotetulla liikenteellä. Tekemättä mitään, se aiheuttaisi mainostettujen verkkojen olevan tavoittamattomissa ja liikenteen hukkumisen. Vaihtoehtoisesti se voi itse reitittää verkot alkuperäiselle ylikirjoittamalleen AS-alueelle, jolloin kaikki verkkoalueille menevä liikenne kiertää virheellisen mainostajan kautta ja mahdollisesti syntyy epäoptimaalinen reitityspolku. (Mts. 14.)

NEXT_HOP-tietoa käytetään määrittämään, mitä reititintä tulisi käyttää seuraavana hyppynä mainostetuille NLRI-arvoille. EBGP-naapuruuksissa NEXT_HOP:in arvon ja mainostuksen vastaanottavan reitittimen tulee olla samasta aliverkosta. Mikäli ehdot täyttyvät voidaan virheellisellä mainostuksella ohjata liikenne BGP-reitittimelle, joka ei oman reititystaulunsa takia pysty reitittämään liikennettä eteenpäin. Toinen skenaario on, että liikenne ohjataan AS-alueelle, joka ei normaalisti kuulu liikenteen AS-polkuun, mutta jolla on reitti kohdeverkkoon. Tällöin syntyy epäoptimaalinen reitityspolku. (Mts. 15.)

ATOMIC_AGGREGATE-arvo kertoo, että jokin AS-alue reitityspolulla on yhdistänyt reittitietojen verkkoalueita toisiinsa muodostaen niistä yhden NLRI-reittitiedon. Mikäli arvo on määritetty, ei BGP-reitittimillä ole oikeutta mainostaa NLRI:n sisältämiä

verkkoja samalla AS-polulla, mutta tarkemmilla prefikseillä. Arvon tahallinen poistaminen voi johtaa tarkempien prefiksien mainostamiseen yhdistettyjen verkkojen polulla, mikä puolestaan voi aiheuttaa vääriä reitityspäätöksiä. (Mts. 16.)

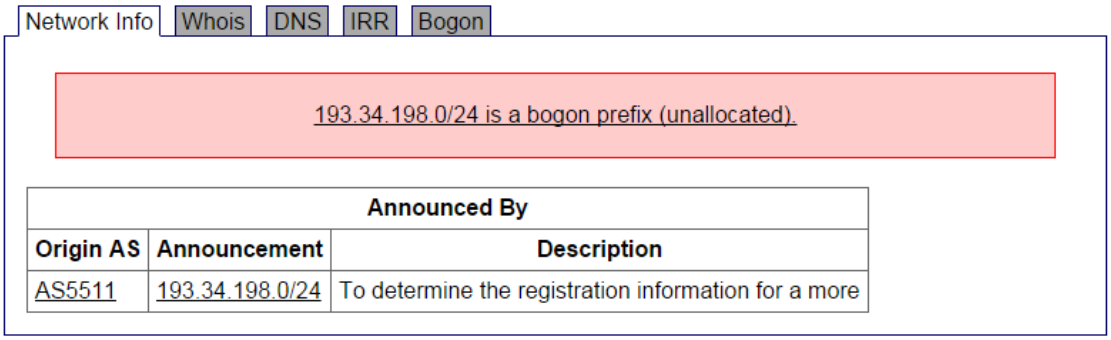
Lisäksi oikeutettu BGP-naapuri voi häiritä yksittäistä naapuruutta lähettämällä NOTIFICATION-viestin, joka johtaa yhteyden katkaisemiseen ja kaikkien liittyvien reittien poistamiseen reititystaulusta. Vaihtoehtoisesti naapurille voidaan lähettää BGP:n tilakoneen normaalin toiminnan ulkopuolisia viestejä, joista lähes kaikki voivat johtaa naapuruuden alasajoon. Naapuruuden lopettamiseen voi johtaa liian monen prefiksin mainostaminen UPDATE-viesteissä, jolloin naapuruuskohtainen konfiguroitu maksimiprefiksimäärä ylittyy. Naapurin asetuksista riippuen yhteys voidaan katkaista myös OPEN-viestin tullessa Established-tilassa. Konfiguroinnista huolimatta naapuruus katkeaa epämuodostuneen UPDATE-viestin vastaanoton yhteydessä. (Murphy 2006, 9-11, 18.)

4.2.2 Nykyinen toimintamalli

Internet palveluntarjoajat estävät oikeutettujen BGP-naapureiden ja -toimijoiden aiheuttamia uhkakuvia käyttämällä erinäisiä suodatustekniikoita. IBGP-sessioita suojataan suodattamalla sisäisiin rajapintoihin kohdistettu TCP-porttia 179 käyttävä liikenne palveluntarjoajan reunareitittimillä. Mikäli porttiin 179 menevää liikennettä tulee palveluntarjoajan verkon sisällä, sitä ei ohjata IBGP-rajapinnasta reitittimen EBGP-rajapintaan eikä reitittimen BGP-naapurin EBGP-rajapintaan. (Murphy 2006, 20.)

Gaurab (n.d.) on kerännyt palveluntarjoajille tarkoitettuja suosituksia BGP-reitityksen suojaamiseen. On esimerkiksi olemassa erikoiskäyttöön tarkoitettuja IP-osoitteita, joita ei tulisi koskaan mainostua julkiseen internettiin, vaan ne tulisi suodattaa pois AS-alueiden rajoilla. Suodatus tulisi tehdä sekä sisään- että ulospäin tehtävissä mainostuksissa. Näihin osoitteisiin kuuluvat esimerkiksi privaatit osoitealueet kuten 10.0.0.0/8 ja loopback-osoitteet 127.0.0.0/8. Täydellinen lista on koottu RFC:hen

3330 - Special-Use IPv4 Addresses. Tämän lisäksi internettiin ei saisi mainostua osoitealueita, joita IANA ei ole allokoinut. Kuviossa 14 näkyy Hurricane Electric-palveluntarjoajan ylläpitämä BGP-statistiikkasivu allokoimattomasta osoitealueesta 193.34.198.0/24 ja kuinka AS-alue 5511 (Orange) mainostaa kyseistä verkkoa. (Gaurab n.d., 23-27.)



Kuvio 14. Orangen allokoimattoman verkon mainostus (Hurricane Electric - 193.34.198.0/24 2015.)

Yksittäisen AS-alueen tulisi mainostaa itseään reittien alkuperäisenä lähteenä naapureilleen vain omistamiensa prefiksien ja asiakkaidensa omistamien prefiksien osalta. Muut reitit tulisi suodattaa pois mainostuksista. Näihin kuuluvat allokoimattomat ja RFC:n 3330 mainitsemat osoitealueet. Lisäksi tarkempia kuin /24 prefiksejä ei tulisi mainostaa EBGP-naapureille. Naapureilta tulevista BGP-mainostuksista tulisi suodattaa pois edellä mainittujen lisäksi vastaanottajan itse omistamat prefiksit ja oletusreitit. Jokaiselle naapurille olisi hyvä määrittää myös mainostettujen prefiksien yläraja. AS-alueen omistajan, kuten ISP:n, tulisi sallia asiakkailtaan vain tarkalleen heille allokoitujen verkkojen mainostamisen. Muut mainostukset tulisi suodattaa. (Mts. 37, 40-41, 44-46, 53-54.)

Vuoden 2008 helmikuussa pakistanilainen Pakistan Telecom mainosti vahingossa olevansa alkuperäinen lähde YouTube.com-osoitteelle. Koska kyseisen operaattorin internet-liikenteen tarjoava Hong Kongilainen PCCW ei tarkistanut, onko verkkoalue al-

lokoitu Pakistan Telecomille ja sen jälkeen suodattanut sitä virheellisenä mainostuksena, se mainosti reittiä eteenpäin naapureilleen. PCCW:n naapurit mainostivat verkkoa edelleen. Lopputuloksena Pakistan Telecomin mainostama 208.65.153.0/24 otettiin käyttöön globaaleissa reititystauluissa YouTuben itsensä mainostaman epätarkemman 208.65.152.0/22 ohitse. Konfiguraatiovirhe aiheutti yli kahden tunnin kokonaiskatkoksen YouTube.com-sivustolle. (McCullagh 2008.)

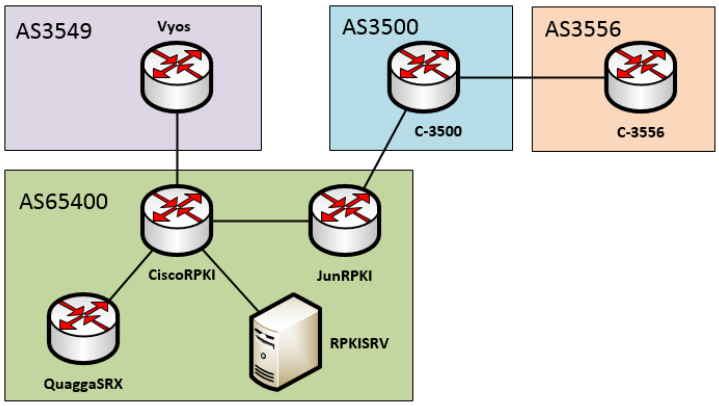
Vuoden 2010 huhtikuussa Kiinan suurimman palveluntarjoajan China Telecomin omistama datacenter, jonka AS-alue on 23724, mainosti 15 minuutin ajan olevansa alkuperä noin 37 000 prefiksille. Normaalisti AS23724 on alkuperänä noin 40 prefiksille. Mainostetuista reiteistä vain 10 prosenttia jatkomainostui Kiinan verkon ulkopuolelle, mutta niiden joukkoon kuuluivat esimerkiksi dell.com- ja amazon.de-sivustojen prefiksit. Kaikkien virheellisten mainostusten AS-polut sisälsivät sarjan 4134-23724-23724, jossa AS-alue 4134 kuuluu Chinanet Backbonelle. AS4134:n vertaisnaapurit eivät tarkistaneet tai suodattaneet mainostuksia vaan levittivät niitä edelleen. (Toonk 2010.)

5 LAAJENNUSTEN IMPLEMENTOINTI

5.1 Ympäristö ja esivalmistelut

Ainoastaan Resource Public Key Infrastructure -laajennuksesta löytyi työn tekovaiheessa laitevalmistajien ohjelmistotuki, joten se valikoitui ainoaksi käytännön implementaatioksi. Eri näkökulmien saamiseksi implementaatio päätettiin toteuttaa hyödyntäen Ciscon ja Juniperin reitittimiä sekä National Institute of Standards and Technologyn (NIST) kehittämää Quagga-reititinohjelmistoon perustuvaa QuaggaSRX-versiota. Valitettavasti käytössä ei ollut Juniperin tai Ciscon fyysisiä laitteita vaan implementointi toteutettiin kokonaan virtualisoituna.

Resurssit virtualisointiin tarjosi JYVSECTEC ja kaikkia laitteita ajettiin heidän VMware-ympäristössä. Kokonaisuus pidettiin eristettynä muista virtuaaliympäristöistä. Laitteiden välisille linkeille luotiin erilliset virtuaaliverkot, joilla niiden välinen tosimaailmasta eroava keskustelu estettiin. Käytännössä laitteet toimivat keskenään point-to-point-yhteyksillä. Jaotteluun käytettiin myös liitteen 1 mukaista aliverkoittamista. Kuviossa 15 näkyy implementaatiossa käytetty topologia AS-alueineen.



Kuvio 15. Implementaatiossa käytetty topologia

Juniper on julkaissut reitittimilleen RPKI-ohjelmistotuen Junos OS:n versiossa 12.2, joka on käytössä vain M-, MX- ja T-sarjan laitteissa. Cisco puolestaan tukee RPKI:ta

useammassa tuoteperheessä, mutta vaatimuksena on vähintään joko IOS-XR versio 4.2.1 tai IOS-XE versio 3.5. Näiden rajoitteiden valossa ainoat mahdolliset toteutuskohteet JYVSECTECin puolesta olivat Juniper vMX- ja Cisco CSR1000v-virtuaalireitittimet. QuaggaSRX:n kohdalla käytettiin uusinta saatavilla olevaa ohjelmistokokonaisuutta. Koko reitityslaitteiston käytetyt ohjelmistoversiot löytyvät taulukosta 2. (Example: Configuring Origin Validation for BGP 2013; Router Configuration 2014.)

Taulukko 2. Käytetyt reitittimien ohjelmistoversiot

Reititin	Malli	Ohjelmisto
CiscoRPKI	Cisco CSR1000V	IOS-XE 3.13.01.S
JunRPKI	Juniper vMX	14.1R2.12
QuaggaSRX	QuaggaSRx	0.4.1.3
	SRx Server	0.3.0.8
	SRx Crypto API	0.1.1.1
vyos	vyOS	1.1.6
C-3500	Cisco CSR1000V	IOS-XE 3.13.01.S
C-3556	Cisco CSR1000V	IOS-XE 3.13.01.S

Ideana implementaatiossa oli mallintaa yhtä AS-aluetta 65400, joka käyttäisi reitittimissään RPKI:ta ja Prefix Origin Validationia varmistamaan, että ulkoiset naapurit mainostavat vain verkkoalueita, joihin heillä on oikeus. Virheelliset mainostukset haluttiin hylätä suoraan lisäämättä niitä reititystauluun. Topologiassa AS-alueita 3549 ja 3556 käytettiin mainostamaan erinäisiä verkkoja kuin ne omistaisivat kyseiset verkot. Lisäksi AS3500 mainosti yksittäistä osoitetta. Alla olevassa taulukossa 3 näkyy AS-alueiden mainostamat reitit ja onko niillä oikeus mainostaa niitä. Syy mainostusoikeuden kielteisyydelle on lihavoituna. Keltaisella merkityillä verkkoalueilla ei ole

ROA-objektia, joten niitä saa periaatteessa mainostaa vapaasti RPKI:n puitteissa. Käytetyt AS-alueet valittiin mielivaltaisesti ROA-objekteja julkaisseiden AS-alueiden joukosta.

Taulukko 3. AS-alueiden mainostamat verkkoalueet

AS-alue	Mainostama verkko	ROA AS	ROA Verkkoalue	MaxLength	Mainostusoikeus?
3549	12.34.56.0/24	N/A	N/A	-	Periaatteessa
	181.224.172/24	3549	181.224.172/24	/24	Kyllä
	181.225.80.0/21	3549	181.225.80.0/21	/21	Kyllä
	191.97.17.0/24	3549	191.97.17.0/22	/30	Kyllä
	200.229.217.0/24	**3556**	200.229.217.0/24	/24	Ei
3556	181.225.81.0/24	**3549**	181.225.80.0/21	**/21**	Ei
	200.229.217.0/24	3556	200.229.217.0/24	/24	Kyllä
	200.229.217.1/32	3556	200.229.217.0/24	**/24**	Ei
3500	35.0.0.1/32	N/A	N/A	-	Periaatteessa
Kaikki	Privaattiosoitteet	N/A	N/A	-	Periaatteessa

Tieto mainostuksen oikeellisuudesta perustui alueellisten RIR-toimijoiden julkaisemiin ja keräämiin ROA-objekteihin. Alla olevassa kuviossa 16 näkyy 7.11.2015 oleva ROA-objektien lukumäärä jaettuna RIR-toimijoiden kesken. Kuvio on otettu RPKI Validator API:n web-käyttöliittymästä.

Enabled	Trust anchor	Processed Items	Expires in	Last updated	Next update in
☑	APNIC from AFRINIC RPKI Root	12 0 0	4 years and 1 month	3 minutes ago	6 minutes
☑	APNIC from ARIN RPKI Root	105 0 0	4 years and 9 months	3 minutes ago	6 minutes
☑	APNIC from IANA RPKI Root	1955 0 0	4 years and 11 months	3 minutes ago	6 minutes
☑	APNIC from LACNIC RPKI Root	6 0 0	4 years and 1 month	3 minutes ago	6 minutes
☑	APNIC from RIPE RPKI Root	27 0 0	4 years and 1 month	3 minutes ago	6 minutes
☑	AfriNIC RPKI Root	308 0 0	4 years and 6 months	4 minutes ago	5 minutes
☑	LACNIC RPKI Root	2388 0 0	6 years and 4 months	3 minutes ago	6 minutes
☑	RIPE NCC RPKI Root	12870 0 0	4 years and 11 months	38 seconds ago	9 minutes

Kuvio 16. RIR-toimijoiden julkaisemien ROA-objektien lukumäärä

AS-alueella 65400 toimivat RPKI-reitittimet muodostivat jokainen oman yhteytensä RPKI-palvelimeen, josta ne latasivat muistiinsa sen hetkiset ROA-objektien tiedot käyttäen RPKI To Router -protokollaa (RTR). Liitteessä 2 on havainnollistettu verkon aktiivilaitteiden välinen topologia sekä merkattu taulukon 3 värikoodeilla AS-alueiden mainostamat verkot.

5.2 Implementaation toteuttaminen

Ensimmäisenä asennettiin CentOS 6:n päällä pyörivä Linux-palvelin nimeltä RPKISRV, jonka tehtävänä oli toimia ympäristön RPKI-palvelimena, jolta reitittimet pystyivät kysymään virallisesti julkaistujen Router Origin Authorization-objektien tietoja. Palvelin ylläpiti muistissaan omaa tietokantaansa objekteista. Palvelimelle ladattiin RIPE:n sivuilta RPKI Validator API 2.20-ohjelmiston pakkaus, jota pystyttiin sen purkamisen jälkeen ajamaan suoraan ilman asentamista. Vaatimuksena ohjelmiston toiminnalle Linux-käyttöjärjestelmän lisäksi oli Javan versio 7 sekä rsync-ohjelmisto. Lisäksi palvelimelle asennettiin curl- ja lynx-ohjelmat. Toteutuksessa käytetty Javan versio ja asennuskansio näkyvät kuviossa 17.

```
[root@RPKISRV ~]# which java
/usr/bin/java
[root@RPKISRV ~]# java -version
java version "1.7.0_91"
OpenJDK Runtime Environment (rhel-2.6.2.2.el6_7-x86_64 u91-b00)
OpenJDK 64-Bit Server VM (build 24.91-b01, mixed mode)
```

Kuvio 17. Asennetun Javan tiedot

Ohjelman konfiguroimiseen käytettiin puretun kansion conf-alikansiossa olevaa rpki-validator.conf-tiedostoa. Lopullisessa ympäristössä ohjelmaa ajettiin oletuskonfiguraatioilla, mutta sen web-käyttöliittymän TCP-porttia sekä reitittimien kanssa kommunikointiin käytettävää TCP-porttia olisi voitu vaihtaa seuraavilla arvoilla:

ui.http.port=8080
rtr.port=8282

JYVSECTEC-ympäristön asettamien rajoitusten takia palvelinta ei saanut pitää sekä testausverkossa että Internetissä, joten palvelin liitettiin ensin vain Internettiin ja annettiin sen ladata ROA-objektit muistiinsa rsync-ohjelmistoa käyttäen. Tämän jälkeen laite vaihdettiin vain testausverkkoon. Palvelimen osoitteeksi asetettiin 192.168.0.1 ja oletusreitin seuraavaksi hypyksi määritettiin 192.168.0.254. Tästä ei ollut palvelimen toiminnalle haittaa, sillä vanhojen objektien poistoväli olisi voitu määritellä halutuksi. Arvo jätettiin kuitenkin oletukseksi eli:

validation.remove_old_objects.interval = 7d

Asetusten tarkistamisen jälkeen ohjelmisto käynnistettiin, jonka jälkeen se automaattisesti aloitti ROA-objektien lataamisen. Täydellinen rpki-validator.conf-tiedosto löytyy liitteestä 3. Käynnistys tapahtui seuraavalla komennolla:

./rpki-validator.sh start

Palvelimen pystyttämisen jälkeen siirryttiin konfiguroimaan RPKI:ta hyödyntänyttä Ciscon CSR1000V-reititintä nimeltä CiscoRPKI. Yhteydet naapurireitittimiin sekä RPKI-

palvelimeen mahdollistettiin asettamalla rajapintojen IP-osoitteet vastaamaan reitittimien osoitteistustaulukkoa, joka löytyy liitteestä 1.

Tämän jälkeen luotiin Prefix Origin Validationia varten route-map-sääntö rpkistate, jonka tarkoitus oli hyväksyä vain Valid- ja Not Found -tilaiset reittimainostukset BGP-naapureilta. Invalid-reitit tiputettiin tarkoituksella pois hyödyntämällä route-map-säännön implicit deny-toiminnallisuutta, joka hylkää kaikki mainostukset, jotka eivät jää kiinni mihinkään alisääntöön. Valid-tilaisille reiteille nostettiin Local Preference -arvoa 200:aan, jotta ne otettaisiin käyttöön reititystaulussa Not Found -tilaisten reittien ylitse. Jälkimmäisten reittien Local Preference -arvo päätettiin asettaa oletusarvoon 100 seuraavasti:

```
route-map rpkistate permit 10
 match rpki valid
 set local-preference 200

route-map rpkistate permit 20
 match rpki not-found
 set local-preference 100
```

CiscoRPKI asetettiin toimimaan AS-alueella 65400 ja sille määritettiin 2 IBGP-naapuria 172.16.0.1 (JunRPKI) ja 172.16.0.5 (QuaggaSRX) sekä EBGP-naapuri 10.0.1.1 (AS3549). Kaikille naapureille otettiin käyttöön vastaanotettujen mainostusten suodattamiseen edellä mainittu sääntö rpkistate. Sisäisille naapureille meneviin mainostuksiin muutettiin CiscoRPKI seuraavaksi hypyksi, jotta AS-alueelta ulospäin menevä liikenne toimisi. Cisco toimi topologian takia IBGP-naapurille 172.16.0.5 Route Reflector-isäntänä:

```
router bgp 65400
 neighbor 10.0.1.1 remote-as 3549
 neighbor 10.0.1.1 route-map rpkistate in
neighbor 172.16.0.1 remote-as 65400
 neighbor 172.16.0.1 next-hop-self
 neighbor 172.16.0.1 route-map rpkistate in
neighbor 172.16.0.5 remote-as 65400
 neighbor 172.16.0.5 route-reflector-client
 neighbor 172.16.0.5 next-hop-self
```

neighbor 172.16.0.5 route-map rpkistate in

BGP Prefix Origin Validation-tilojen tarkistamista varten reitittimelle kerrottiin RPKISRV-palvelimen osoite, käytetty TCP-portti sekä ROA-tietojen tarkistusväli sekunteina:

bgp rpki server tcp 192.168.0.1 port 8282 refresh 600

Näillä konfiguraatioilla EBGP-naapureilta saatujen mainostusten tarkistaminen olisi onnistunut, mutta Cisco on ohjelmistossaan päättänyt, että IBGP-naapureiden mainostukset merkataan ilman tarkistuksia Valid-tilaisiksi. Tilatietoja ei myöskään oletuksena mainosteta IBGP-naapureille. Tämän kiertämiseksi IBGP-naapureille määritettiin erikseen Prefix Origin Validation -tilojen mainostaminen ja kuunteleminen. Koska tilaa mainostetaan tietyllä Extended Community -arvolla, tuli myös niiden lähettäminen ottaa käyttöön seuraavasti (BGP-Origin AS Validation n.d., 3):

neighbor 172.16.0.1 send-community extended
neighbor 172.16.0.1 announce rpki state
neighbor 172.16.0.5 send-community extended
neighbor 172.16.0.5 announce rpki state

Ympäristön testaamisen, toimivuuden ja konfiguraatioiden yksinkertaisena pitämisen takia lähes kaikilta reitittimiltä mainostettiin BGP-naapureille reitittimeen kytkettyjä verkkoalueita. Ciscon tapauksessa tämä tehtiin seuraavasti:

redistribute connected

Juniper vMX -reitittimelle nimeltä JunRPKI asetettiin rajapintakonfiguraatiot Ciscon tavoin suunnitelman mukaisesti. Tämän jälkeen reitittimelle luotiin sääntö direct, jolla mahdollistettiin siihen kytkettyjen verkkojen mainostaminen EBGP-naapureille. Sisäisille IBGP-naapureille tehtiin sääntö ibgp, joka kytkettyjen reittien lisäksi määritti JunRPKI:n reittien seuraavaksi hypyksi.

set policy-options policy-statement direct term networks from protocol direct
set policy-options policy-statement direct term networks then accept

set policy-options policy-statement ibgp term connected from protocol direct
set policy-options policy-statement ibgp term connected then next-hop self
set policy-options policy-statement ibgp term connected then accept
set policy-options policy-statement ibgp term nexthop then next-hop self
set policy-options policy-statement ibgp term nexthop then accept

Jotta JunRPKI ja CiscoRPKI saisivat vaihdettua keskenään Prefix Origin Validation -tiloja, luotiin reitittimelle Extended Communityt origin-validation-state-valid ja origin-validation-state-unknown Juniperin dokumentaation mukaisesti, missä syntaksi on 0x43:AS-alue:tila-arvo. Komentoina tämä tehtiin seuraavasti (Example: Configuring Origin Validation for BGP 2013):

set policy-options community origin-validation-state-unknown members 0x43:65400:1
set policy-options community origin-validation-state-valid members 0x43:65400:0

Kyseisellä notaatiolla tila-arvojen kommunikointia reitittimien välillä ei saatu toteutettua. Vianrajauksen ja juurisyyn etsinnän jälkeen löydettiin, että ongelma johtui heksadesimaaliarvon väärästä käsittelystä JunRPKI-laitteella. Juurisyystä on lisää asiaa myöhemmin tässä dokumentissa. Lopulta päädyttiin käyttämään seuraavia arvoja:

set policy-options community origin-validation-state-unknown members 0x4300:65400:1
set policy-options community origin-validation-state-valid members 0x4300:65400:0

Varsinainen Prefix Origin Validation -tilan tarkistaminen tehtiin säännöllä RPKI. Mikäli vastaanotetun BGP-viestin reittitiedon tila reitittimen muistissa oli Valid, asetettiin Local Preference -arvoksi 200, merkattiin tilaksi Valid ja lisättiin sille community origin-validation-state-valid. Not Found -tilojen kohdalla arvot olivat 100, Unknown ja origin-validation-state-unknown. Lopuksi mainostukset hyväksyttiin. Säännön termeissä mainittu validation-state-toiminne saa aikaan Prefix Origin Validation -tilan

tarkistamisen muistista. Kaikki muun tilaiset reittitiedot hylättiin tarkoituksella seuraavasti:

set policy-options policy-statement RPKI term valid from protocol bgp
set policy-options policy-statement RPKI term valid from validation-database valid
set policy-options policy-statement RPKI term valid then local-preference 200
set policy-options policy-statement RPKI term valid then validation-state valid
set policy-options policy-statement RPKI term valid then community add origin-validation-state-valid
set policy-options policy-statement RPKI term valid then accept
set policy-options policy-statement RPKI term notfound from protocol bgp
set policy-options policy-statement RPKI term notfound from validation-database unknown
set policy-options policy-statement RPKI term notfound then local-preference 100
set policy-options policy-statement RPKI term notfound then validation-state unknown
set policy-options policy-statement RPKI term notfound then community add origin-validation-state-unknown
set policy-options policy-statement RPKI term notfound then accept
set policy-options policy-statement RPKI term invalid from protocol bgp
set policy-options policy-statement RPKI term invalid then reject

Sääntöjen luomisen jälkeen reititin asetettiin samaan AS-alueeseen 65400 kuin Cisco ja sille määritettiin käytetty RPKI-palvelin. Huomionarvoista on, että palvelin ei sijainnut suoraan Juniperin kanssa samassa aliverkossa vaan laitteen CiscoRPKI takana. Ciscoon nähden erona hold-time-arvon pystyi määrittämään itse. Lisäksi piti määrittää Juniperin kommunikointiin käyttämä osoite:

set routing-options autonomous-system 65400
set routing-options validation group rpkisrv session 192.168.0.1 refresh-time 600
set routing-options validation group rpkisrv session 192.168.0.1 hold-time 800
set routing-options validation group rpkisrv session 192.168.0.1 port 8282
set routing-options validation group rpkisrv session 192.168.0.1 local-address 172.16.0.1

Lopuksi reitittimelle konfiguroitiin käyttöön BGP, joka hyödynsi aiemmin määritettyjä sääntöjä. IBGP-naapuriksi asetettiin 172.16.0.2 (CiscoRPKI) ja EBGP-naapuriksi 10.0.2.2 (AS3500). Prefix Origin Validation-tilojen tarkastaminen UPDATE-viesteistä otettiin käyttöön sekä sisäisille että ulkoisille naapureille. Juniper ei oleta IBGP-naapureilta saatuja mainostuksia Valid-tilaisiksi.

set protocols bgp group int type internal
set protocols bgp group int import RPKI
set protocols bgp group int export ibgp
set protocols bgp group int neighbor 172.16.0.2
set protocols bgp group ext type external
set protocols bgp group ext import RPKI
set protocols bgp group ext export direct
set protocols bgp group ext peer-as 3500
set protocols bgp group ext neighbor 10.0.2.2

Kolmas RPKI:ta hyödyntävä reititin oli CentOS 6:n päällä toiminut QuaggaSRx-toteutus, joka on NIST:in kehittämä arkkitehtuuri, joka on suunniteltu tukemaan erinäisiä BGP-tietoturvalaajennuksia pohjautuen Quagga-versioon 0.99.22. Se koostuu kolmesta eri osasta, joiden keskinäinen toiminta mahdollistaa Prefix Origin Validation -tilojen tarkistelun. Ohjelmistot pitää hakea NIST:in ylläpitämästä ohjelmistorepositoriosta, joten CentOS-käyttöjärjestelmän yum-asennusohjelmaa varten tuli luoda uusi tiedosto /etc/yum.repos.d/srx.repo. Tiedoston sisälle asetettiin seuraavat arvot:

[srx]
name = Secure Routing Extension (SRx) and QuaggaSRx
baseurl = http://bgpsrx.antd.nist.gov/srx-repo/
enabled=1
gpgcheck=0

Tämän jälkeen tarvitut ohjelmistot ladattiin käyttäen yum-ohjelmaa. Asennusten jälkeen palvelin siirrettiin Internet-yhteyden perästä testausverkkoon. Pakettien asentaminen tapahtui näin:

yum install srxcryptoapi srx quaggasrx

SRx Server -ohjelman funktio on olla yhteydessä RPKI-palvelimeen ja pitää muistissaan ROA-objektien arvoja. Ohjelman konfiguraatiotiedosto sijaitsee asennuksen jälkeen kohteessa /etc/srx_server.conf. Täydellinen konfiguraatiotiedosto löytyy liitteestä 4. Asetukset pidettiin muuten oletuksina, mutta RPKI-palvelimen RPKISRV osoitteeksi vaihdettiin 192.168.0.1 ja portiksi määritettiin 8282. Ohjelman kuunteluporttina pidettiin 17900 seuraavilla muokkauksilla:

```
sync    = true;
port    = 17900;
rpki: {
  host = "192.168.0.1";
  port = 8282;
};
```

SRx Crypto API palvelee sekä QuaggaSRx- että SRx Server -ohjelmistoja tarjoten niille salausprotokolliin liittyviä toimintoja BGPSEC-polkuihin liittyen. Toteutettu RPKI-ympäristö ei hyödyntänyt API:a, mutta kaksi muuta ohjelmistoa vaativat sen toimiakseen. Sen konfiguraatiotiedostona toimii /etc/srxcryptoapi.conf, joka löytyy kokonaisuudessaan liitteestä 4.

Varsinainen BGP-konfiguraatio tehdään tiedostoon /etc/bgpd.conf, joka toimii QuaggaSRx-ohjelman asetustiedostona. Reititin asetettiin Ciscon ja Juniperin kanssa samaan AS-alueeseen ja määritettiin CiscoRPKI:n 172.16.0.6 IBGP-naapuriksi seuraavasti:

```
router bgp 65400
bgp router-id 172.16.0.5
neighbor 172.16.0.6 remote-as 65400
```

Koska Prefix Origin Validation -tilat löytyvät SRx Server -ohjelmalta, piti reitittimelle kertoa, mistä kyseinen palvelin löytyy. Tähän liittyi myös aiemmin määritetty kuunteluportti 17900. Quaggan käyttämä oma IP-osoite 172.16.0.5 tuli myös määrittää. Koska ympäristössä testattiin vain RPKI-laajennusta, mainostusten arviointiperus-

teeksi laitettiin origin_only. Toinen vaihtoehto olisi ollut BGPSEC. Prefix Origin Validationiin liittyvät show-lisäkomennot otettiin käyttöön srx display-määreellä seuraavasti:

srx set-proxy-id 172.16.0.5
srx set-server 127.0.0.1 17900
srx keep-window 900
srx evaluation origin_only
srx display

Vastaanotettujen reittitietojen Prefix Origin Validation-tilojen oletusarvoksi määritettiin undefined, kunnes SRx Serveriltä saataisiin varmistettua oikea tila. Lisäksi asetettiin reititin valitsemaan identtisistä reittitiedoista se, jonka tila on Valid. Invalid-tilaiset mainostukset päätettiin hylätä. Lopuksi ohjeistettiin reititin yhdistämään SRx Serveriin seuraavasti:

srx set-origin-value undefined
srx policy prefer-valid
srx policy ignore-invalid
srx connect

Valid-tilaisille mainostuksille asetettiin Local Preference -arvo 200 ja Not Found -tilaisille arvo 100. QuaggaSRX:llä otettiin käyttöön myös Extended Community -arvot Prefix Origin Validation -tilojen kommunikoitiin IBGP-naapureiden kanssa seuraavilla komennoilla:

srx extcommunity 1 only_ibgp
srx policy local-preference valid 200
srx policy local-preference notfound 100

Käytetty konfigurointitiedosto löytyy liitteestä 4. Tehdyn konfiguraation jälkeen sekä QuaggaSRx että SRx Server tuli käynnistää CentOS-käyttöjärjestelmästä. Jotta ohjelmat saatiin ajettua taustalla häiritsemättä muuta toimintaa, niille annettiin lisämääre & seuraavasti:

srx_server &
bgpd &

Muilla ympäristön reitittimillä oli rajapintakonfiguraatioiden lisäksi minimaaliset BGP-konfiguraatiot. AS-alueen 3500 reitittimellä nimeltään C-3500 mainostettiin siihen kytkettyjä reittejä sekä sen loopback-osoitetta seuraavasti:

network 35.0.0.1 mask 255.255.255.255
redistribute connected

AS-alueella 3556 toiminut reititin C-3556 konfiguroitiin mainostamaan verkkoja 181.225.81.0/24 ja 200.229.217.0/24 tekemällä niihin staattiset reitit, jotka viittasivat Null0-rajapintaan. Näin reititin lisää ne reititystauluunsa vaikka sillä ei oikeasti olisi niitä kytketyissä verkoissaan ja niitä voidaan mainostaa edelleen. Myös reitittimeen kytkettyjä verkko mainostettiin:

interface Loopback0
ip address 200.229.217.1 255.255.255.255
router bgp 3556
redistribute connected
redistribute static
ip route 181.225.81.0 255.255.255.0 Null0
ip route 200.229.217.0 255.255.255.0 Null0

VyOS-käyttöjärjestelmän päällä toiminut vyos-reititin toimi AS-alueella 3549 ja sen tehtävänä oli mainostaa seuraavia verkkoja: 12.34.56.0/24, 181.224.172.0/24, 181.225.80.0/21, 191.97.17.0/24, 200.229.217.0/24. Kyseisiä verkkoja se mainosti kytkettyinä seuraavalla komennolla:

set protocols bgp 3549 redistribute 'connected'

Kaikki reitittimissä käytetyt konfiguraatiot löytyvät tämän dokumentin liitteistä 4-9.

5.3 RPKI:n todentaminen

RIPE RPKI Validator API:n toiminta varmistettiin tekemällä palvelimelle RESTful-rajapintaa hyödyntäen kyselyitä curl-ohjelmistolla. Kuviossa 18 näkyy tarkistus, jossa

API:lta kysyttiin kuuluuko verkkoalue 200.229.217.0/24 AS-alueelle 3549. Vastauksesta käy ilmi, että verkkoalueen oikea omistaja on AS-alue 3556, joten Prefix Origin Validation-tilan arvoksi tulee Invalid. Tuloste on JSON-muodossa.


```
[root@localhost ~]# curl localhost:8080/api/v1/validity/3549/200.229.217.0/24
{
  "validated_route":{
    "route":{
      "origin_asn":"AS3549",
      "prefix":"200.229.217.0/24"
    },
    "validity":{
      "state":"Invalid",
      "reason":"as",
      "description":"At least one VRP Covers the Route Prefix, but no VRP ASN match
es the route origin ASN",
      "VRPs":{
        "matched":[],
        "unmatched_as":[{
          "asn":"AS3556",
          "prefix":"200.229.217.0/24",
          "max_length":24
        }],
        "unmatched_length":[]
      }
    }
  }
```


Kuvio 18. RPKI-palvelimen curl-tuloste

Palvelimelta voidaan myös kysyä web-liittymän kautta kaikkien ROA-objektien tulosteet kerrallaan joko RPSL-, JSON- tai CSV-muodossa ulosvientiä varten. Alla olevassa kuviossa 19 näkyy rinnakkain pätkä CSV-tulostetta (vasen) ja RPSL-tulostetta (oikea).


```
ASN,IP Prefix,Max Length
AS31207,86.106.22.0/24,24
AS6830,195.234.172.0/24,24
AS2200,2001:660::/32,32
AS18014,113.29.248.0/24,24
AS39388,93.152.253.0/24,24
AS6453,194.126.0.0/19,24
```

```
route: 212.124.199.0/24
origin: AS8468
descr: exported from ripe ncc validator
mnt-by: NA
created: 2015-11-05T15:25:34Z
last-modified: 2015-11-05T15:25:34Z
source: ROA-RIPE-NCC-RPKI-ROOT
```


Kuvio 19. RPKI-palvelimen vientituloste

Testauksen kannalta tärkein ominaisuus oli nähdä web-käyttöliittymässä reitittimien ja palvelimen väliset istunnot. Kuviosta 20 käy ilmi kaikki ympäristön 3 RPKI-reititintä ja kuinka kauan istunto on ollut voimassa. Voidaan myös huomata sekä palvelimen että jokaisen reitittimen viimeksi lähettämä RTR-protokollan viesti.

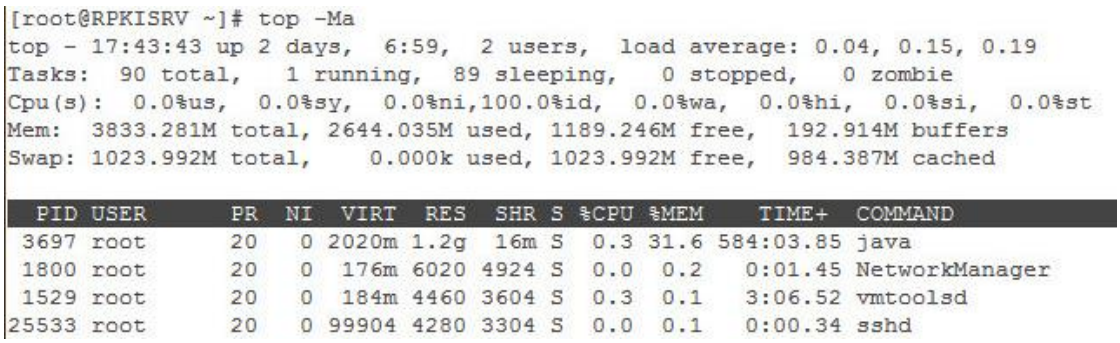
Router Sessions

This table shows all routers connected to this RPKI Validator. Requests and responses are described in RFC 6810. For debugging, please refer to rtr.log.

Remote Address	Connection Time	Last Request Time	Last Request	Last Reply
172.16.0.5:41518	2015-11-05T15:26:34+02:00	2015-11-05T17:20:58+02:00	ResetQuery	SerialNotifyPdu
192.168.0.254:40182	2015-11-05T17:05:38+02:00	2015-11-05T17:15:39+02:00	ResetQuery	SerialNotifyPdu
172.16.0.1:52002	2015-11-05T09:57:32+02:00	2015-11-05T17:20:58+02:00	ResetQuery	SerialNotifyPdu

Kuvio 20. RPKI-palvelimen ja reitittimien väliset istunnot

Tutkimalla palvelimen käyttämiä resursseja CentOS:in sisäänrakennetulla top-ohjelmalla, voidaan kuvion 21 mukaisesti huomata, että RPKI Validator API:n käyttämä Java-rajapinta vei 31 prosenttia palvelimen neljästä gigatavusta keskusmuistia eli noin 1,2 gigatavua. Palvelimella ei pyörinyt käyttöjärjestelmän ja API:n lisäksi mitään muuta ohjelmia.



```
[root@RPKISRV ~]# top -Ma
top - 17:43:43 up 2 days,  6:59,  2 users,  load average: 0.04, 0.15, 0.19
Tasks:  90 total,   1 running,  89 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.0%us,  0.0%sy,  0.0%ni,100.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:  3833.281M total, 2644.035M used, 1189.246M free,  192.914M buffers
Swap: 1023.992M total,    0.000k used, 1023.992M free,  984.387M cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 3697 root      20   0 2020m 1.2g  16m S  0.3 31.6 584:03.85 java
 1800 root      20   0  176m 6020 4924 S  0.0  0.2   0:01.45 NetworkManager
 1529 root      20   0  184m 4460 3604 S  0.3  0.1   3:06.52 vmtoolsd
25533 root      20   0 99904 4280 3304 S  0.0  0.1   0:00.34 sshd
```

Kuvio 21. RPKI-palvelimen muistinkäyttö

AS-alueen 65400 reunoilla sijainneet CiscoRPKI- ja JunRPKI-reitittimet onnistuivat suodattamaan virheelliset Invalid-tilaiset verkot pois EBGP-naapureiden mainostuksista. Kuviossa 22 näkyy CiscoRPKI:n BGP:llä oppimat reitit ulkoisille AS-alueille. Ulkoiselta naapurilta 10.0.1.1 opittu reitti 12.34.56.0/24 on tunnistunut oikein Not Found-tilaiseksi, kuten myös JunRPKI:lta IBGP:ltä opittu Not Found -tilainen 35.0.0.1/32 reitti. Tästä voidaan päätellä, että Ciscon oletusarvoinen sisäisiltä naapureilta opittujen reittien Valid-tilaiseksi merkkaaminen saatiin kierrettyä. Kuviosta on karsittu selkeyttämiseksi pois privaattiosoitealueet. Kaikkien testiympäristössä käytettyjen reitittimien reititystaulut lopullisessa implementaatiossa löytyvät liitteestä 10.

```
CiscoRPKI#sh ip bgp
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
N*>  12.34.56.0/24    10.0.1.1                 1    100      0 3549 ?
N*>i 35.0.0.1/32      172.16.0.1               0    100      0 3500 i
V*>  181.224.172.0/24 10.0.1.1                 1    200      0 3549 ?
V*>  181.225.80.0/21  10.0.1.1                 1    200      0 3549 ?
V*>  191.97.17.0/24   10.0.1.1                 1    200      0 3549 ?
V*>i 200.229.217.0    172.16.0.1                    200      0 3500 3556 ?
```

Kuvio 22. CiscoRPKI:n BGP-reitit

Juniperin BGP:llä opituissa reiteissä suurin ero on se, että IBGP-naapureilta tulleita mainostuksia ei automaattisesti merkata Valid-tilaiseksi. Kuviossa 23 CiscoRPKI-reitittimeltä opittu reitti 12.34.56.0/24 tunnistuu oikein Not Found -tilaiseksi. Jostain syystä Juniper on päättänyt käyttää tilasta nimitystä Unknown, vaikka heidän dokumentaationsa viittaa tilojen välisen logiikan tulevan RFC:n 6811 luonnosversiosta, joka käsittää tilat Valid, Invalid sekä Not Found. Tämä epäkohta toistuu niin konfiguraatioissa kuin tulosteissa. Juniperillakin Local Preference -arvojen asettaminen onnistui ongelmitta. Kuten Cisconkin tapauksessa, kuviosta on karsittu pois privaattiosoitteet.

```
root@JunRPKI> show route protocol bgp brief
inet.0: 16 destinations, 18 routes (14 active, 0 holddown, 2 hidden
+ = Active Route, - = Last Active, * = Both

12.34.56.0/24          *[BGP/170] 01:02:16, MED 1, localpref 100
                          AS path: 3549 ?, validation-state: unknown
                        > to 172.16.0.2 via em3.0
35.0.0.1/32            *[BGP/170] 01:37:53, MED 0, localpref 100
                          AS path: 3500 I, validation-state: unknown
                        > to 10.0.2.2 via em2.0
181.224.172.0/24       *[BGP/170] 01:03:11, MED 1, localpref 200
                          AS path: 3549 ?, validation-state: valid
                        > to 172.16.0.2 via em3.0
181.225.80.0/21        *[BGP/170] 01:03:11, MED 1, localpref 200
                          AS path: 3549 ?, validation-state: valid
                        > to 172.16.0.2 via em3.0
191.97.17.0/24         *[BGP/170] 01:03:11, MED 1, localpref 200
                          AS path: 3549 ?, validation-state: valid
                        > to 172.16.0.2 via em3.0
200.229.217.0/24       *[BGP/170] 01:38:48, localpref 200
                          AS path: 3500 3556 ?, validation-state: valid
                        > to 10.0.2.2 via em2.0
```

Kuvio 23. JunRPKI:n BGP-reitit

QuaggaSRX-palvelimella saatiin halutusti määritettyä Valid-reiteille Local Preference 200 ja Not Found -reiteille 100. Lisäksi tila saatiin tarkistettua oikein. Prefix Origin Validation -tila näkyy kuvion 24 tulosteen toisessa sarakkeessa sulkujen sisällä. Koska BGPSEC-toimintoa ei käytetty, periytyy Valid-/Not Found -tila sellaisenaan koko reitin oikeellisuuden tilaksi. QuaggaSRX-reititin sijaitsi Route Reflector -roolissa toimineen CiscoRPKI:n takana, joten kaikki mainostukset ovat kiertäneet sen kautta. Tulosteesta on karsittu pois privaattiosoitteiden mainostukset.

```
bgpd# show ip bgp
Validation:    v - valid, n - notfound, i - invalid, ? - undefined
SRx Status:    I - route ignored, D - SRx evaluation deactivated
SRxVal Format: validation result (origin validation, path validation)
Origin codes: i - IGP, e - EGP, ? - incomplete

   Ident    SRxVal SRxLP Status Network          Next Hop            Metric  LocPrf Weight Path
*>i41ADAC47 n(n,-)   100,       12.34.56.0/24    172.16.0.6               1    100a       0 3549 ?
*>i823AE0B4 n(n,-)   100,       35.0.0.1/32      172.16.0.1               0    100a       0 3500 i
*>iF0DB6D7A v(v,-)   200,       181.224.172.0/24 172.16.0.6               1    200a       0 3549 ?
*>iC0741CE3 v(v,-)   200,       181.225.80.0/21  172.16.0.6               1    200a       0 3549 ?
*>iAD3FCEB7 v(v,-)   200,       191.97.17.0/24   172.16.0.6               1    200a       0 3549 ?
*>i5354D53E v(v,-)   200,       200.229.217.0    172.16.0.1                    200a       0 3500 3556 ?
```

Kuvio 24. QuaggaSRX:n BGP-reitit

Reitittimen konfiguraatiosta huolimatta Prefix Origin Validation -tilojen oikein tunnistumisesta kunnia kuuluu SRx Server -komponentille, jolta QuaggaSRX kysyi tila-arvoja jokaisen reitin kohdalla. Tämä voidaan päätellä siitä, että kuvion 25 tulosteessa reittitiedosta 200.229.217.0/24 Ciscon ja Juniperin käyttämä Extended Community-arvo tunnistuu kysymysmerkeillä. Lisäksi kyseinen toiminnallisuus huomattiin jo ennen Extended Communityjen käyttämistä testausverkossa. Siltä osin QuaggaSRX:n toiminta muistutti oletusarvoisesti JunRPKI:ta.

```
bgpd# show ip bgp 200.229.217.0/24
BGP routing table entry for 200.229.217.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  3500 3556
    SRx Information:
      Update ID: 0x5354D53E
      Validation:
        prefix-origin: valid
        path:          processing disabled!
      PathType: AS-PATH
    172.16.0.1 from 172.16.0.6 (10.0.2.1)
      Origin incomplete, localpref 200, valid, internal, best
      Extended Community: ? ?
      Originator: 10.0.2.1, Cluster list: 192.168.0.254
      Last update: Tue Nov 10 08:53:27 2015
```

Kuvio 25. QuaggaSRX:n tunnistama Extended Community

Ciscon tapauksessa väärän Prefix Origin Validation -tilan takia hylättyjä reittejä ei näe route-map-toteutuksen takia BGP-reititystaulussa. AS-alueen 3549 reititin kuitenkin mainosti virheellisesti omistavansa 200.229.217.0/24-verkon. Toiminnallisuuden todistamiseksi reitittimelle pistettiin väliaikaisesti päälle debug tila ja BGP-naapuruus aloitettiin alusta, jotta naapurit vaihtaisivat reittitietonsa UPDATE-viesteillä. Tämä toteutettiin seuraavilla komennoilla:

debug ip bgp updates 10.0.1.1
clear ip bgp 10.0.1.1

Kun BGP-naapuruus oli muodostettu ja reittitiedot vaihdettu, tuli CiscoRPKI:n lokitiedostoon kuviossa 26 oleva viesti, mikä todistaa, että saadun reitin Prefix Origin Validation -tila ei ollut Valid tai Not Found. Kaikki muut tilat nimittäin suodatettiin säännöllä pois.

```
*Nov  6 12:40:05.069: BGP(0): 10.0.1.1 rcvd 200.229.217.0/24 -- DENIED due to: r
oute-map;
```

Kuvio 26. CiscoRPKI:n route-map-lokimerkintä

Juniper puolestaan näyttää mainostuksilla opitut reitit piilotettuina, vaikka ne säännön RPKI mukaan piti hylätä. Prefix Origin Validation -tila kyseisillä reiteillä on kuvion 27 mukaisesti Unverified, joka on Juniperin käyttämä erikoistila reiteille, joille ei ole asetettu oppimisen yhteydessä mitään kolmesta virallisesta tilasta. Koska kumpikaan

reititin ei lisännyt virheellisesti mainostettuja verkkoja BGP-reititystauluunsa, eivät ne myöskään mainostaneet niitä eteenpäin. Näin AS-alueet 3549 ja 3556 eivät kärsineet vääristä mainostuksista, koska niiden liikenne sattui menemään RPKI:ta hyödyntävän AS-alueen 65400 läpi. Se esti myös AS-alueen 65400 sisäisen virheellisten reittien leviämisen.

```
root@JunRPKI> show route hidden

inet.0: 16 destinations, 18 routes (14 active, 0 holddown, 2 hidden)
+ = Active Route, - = Last Active, * = Both

181.225.81.0/24     [BGP ] 00:06:30, localpref 100
                      AS path: 3500 3556 ?, validation-state: unverified
                    > to 10.0.2.2 via em2.0
200.229.217.1/32    [BGP ] 00:06:30, localpref 100
                      AS path: 3500 3556 ?, validation-state: unverified
                    > to 10.0.2.2 via em2.0
```

Kuvio 27. JunRPKI:n hylätyt reitit

CiscoRPKI:lta voitiin myös tarkastella RPKI-palvelimeen muodostetun yhteyden statistiikkaa. Kuviossa 28 näkyy, että palvelimelta 192.168.0.1 (RPKISRV) on saatu 18788 prefiksitietoa. Parhaillaan käynnissä olevan yhteyden tunniste on 19395 ja sarjanumero on 2330. Voidaan myös huomata, kuinka yhteyden muodostaminen epäonnistui lähes 1300 kertaa, mikä johtui konfiguraatiovirheestä palvelimen päässä ympäristön rakennusvaiheessa. Kuvion ottohetkellä tila oli kuitenkin Established. RTR-protokollan paketeille määritettiin automaattisesti Precedence-arvo kuusi, jotta ne saisivat paremman käsittelyn verkossa. Kauneusvirheenä kuvio on otettu ennen reitittimen konfiguraatiossa olevan tietojen päivitysvälin nostamista 60 sekunnista 600 sekuntiin.

```
CiscoRPKI#sh ip bgp rpki server
BGP SOVC neighbor is 192.168.0.1/8282 connected to port 8282
Flags 64, Refresh time is 60, Serial number is 2330, Session ID is 19395
InQ has 0 messages, OutQ has 0 messages, formatted msg 2269
Session IO flags 3, Session flags 4008
 Neighbor Statistics:
  Prefixes 18788
  Connection attempts: 1303
  Connection failures: 1294
  Errors sent: 0
  Errors received: 4

Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Minimum incoming TTL 0, Outgoing TTL 255
Local host: 192.168.0.254, Local port: 42962
Foreign host: 192.168.0.1, Foreign port: 8282
Connection tableid (VRF): 0
Maximum output segment queue size: 50

SRTT: 1000 ms, RTTO: 1003 ms, RTV: 3 ms, KRTT: 0 ms
minRTT: 1 ms, maxRTT: 1000 ms, ACK hold: 200 ms
uptime: 94438978 ms, Sent idletime: 8199 ms, Receive idletime: 8400 ms
Status Flags: active open
Option Flags: keepalive running, nagle, path mtu capable
IP Precedence value : 6
```

Kuvio 28. CiscoRPKI:n yhteys palvelimeen

JunRPKI:lla pystyttiin katselemaan lähestulkoon samaa statistiikkaa. Juniperin ohjelmisto tosin näyttää kuvion 29 mukaisesti, kuinka monta kertaa RPKI-säännön validation-database-määre on aiheuttanut muistissa olevan tietokannan kyselemisen. Lisäksi tulosteesta käy ilmi jakauma, että tarkastetuista tiedoista 55 on palauttanut arvon Valid, 81 arvon Invalid ja 380 arvon Not Found. Testiympäristön pienestä verkostomäärästä ja laitekannasta johtuen voidaan päätellä, että jokainen BGP-naapuruuden purkaminen ja muodostaminen aiheuttaa uudet kyselyt eikä BGP:llä aiemmin opittujen reittien tiloja pidetä muistissa erillisessä tietokannassa. Tämä kulkee käsi kädessä sen kanssa, että BGP-naapuruuden katketessa kaikki kyseisen naapurin reitit poistetaan reititystaulusta välittömästi. Tulosteesta näkyy myös tietokannan vievän noin 3,6 megatavua muistia.

```
root@JunRPKI> show validation statistics
Total RV records: 18788
Total Replication RV records: 18788
  Prefix entries: 18055
  Origin-AS entries: 18788
Memory utilization: 3641365 bytes
Policy origin-validation requests: 516
  Valid: 55
  Invalid: 81
  Unknown: 380
BGP import policy reevaluation notifications: 6
  inet.0, 6
  inet6.0, 0
```

Kuvio 29. JunRPKI:n ROA-statistiikkaa

Varsinaista Juniperin ja RPKI-palvelimen välistä yhteyttä ja sen tietoja voitiin katsella kuvion 30 komennolla. Taas on huomattavissa eroavaisuus, että Ciscon tapauksessa yhteyden tila oli Established, mutta Juniperilla se on Exchange-Full. Tulosteesta näkyy, että yhteys oli ollut pystyssä 3 tuntia 55 minuuttia 54 sekuntia, joka on paljon helpommin tulkattavissa oleva muoto kuin Ciscon käyttämät millisekunnit. Sarjanumero yhteydellä oli 2333.

```
root@JunRPKI> show validation session detail
Session 192.168.0.1, State: exchange-full, Session index: 2
  Group: rpkisrv, Preference: 100
  Local IPv4 address: 172.16.0.1, Port: 8282
  Refresh time: 600s
  Hold time: 800s
  Record Life time: 3600s
  Serial (Full Update): 2333
  Serial (Incremental Update): 2333
    Session flaps: 0
    Session uptime: 03:55:54
    Last PDU received: 00:00:00
    IPv4 prefix count: 16370
    IPv6 prefix count: 2418
```

Kuvio 30. JunRPKI:n palvelinyhteyden statistiikkaa

Reitittimen muistissa olevia ROA-objektien tietoja pystyttiin halutessaan tulostamaan, joten verkon ylläpitäjällä ei tarvitse välttämättä olla pääsyä varsinaiselle RPKI-palvelimelle 192.168.0.1. Kuviossa 31 näkyy CiscoRPKI:n tuloste AS-alueen 3549 omistamista verkkoalueista. Valitettavasti tulosteeseen ei voi määrittää lisämääreitä vaan komento tulostaa kaikki tietueet, joita tässä tapauksessa oli yli 18000. Tämä

kierrettiin käyttämällä include-lisämäärettä, jolla tulosteesta saatiin karsittua pois muut rivit.

```
CiscoRPKI#sh ip bgp rpki table | inc 3549_
138.59.8.0/22         22       3549        0        192.168.0.1/8282
181.224.172.0/24      24       3549        0        192.168.0.1/8282
181.225.80.0/21       21       3549        0        192.168.0.1/8282
191.97.16.0/22        30       3549        0        192.168.0.1/8282
200.0.30.0/23         23       3549        0        192.168.0.1/8282
200.14.34.0/24        24       3549        0        192.168.0.1/8282
201.130.82.0/23       24       3549        0        192.168.0.1/8282
```

Kuvio 31. CiscoRPKI:n ROA-tietueet AS-alueesta 3549

JunRPKI:lla voitiin tietokantakyselyyn määrittää erinäisiä lisämääreitä rajaamaan tulostetta. AS-alueen 3549 omistamat verkkoalueet saatiin tulostettua kuvion 32 komennolla. Valitettavasti 138.59.8.0/22 puuttuu kuviosta inhimillisen erheen takia. Ciscosta poiketen Juniper tulosti myös IPv6-tietueet.

```
root@JunRPKI> show validation database origin-autonomous-system 3549

181.224.172.0/24-24          3549 192.168.0.1                          valid

181.225.80.0/21-21           3549 192.168.0.1                          valid

191.97.16.0/22-30            3549 192.168.0.1                          valid

200.0.30.0/23-23             3549 192.168.0.1                          valid
 *
200.14.34.0/24-24            3549 192.168.0.1                          valid
 *
201.130.82.0/23-24           3549 192.168.0.1                          valid
 *
2001:11:a000::/48-48         3549 192.168.0.1                          valid
 *
2803:1f80::/32-32            3549 192.168.0.1                          valid

2803:8280::/32-32            3549 192.168.0.1                          valid


  IPv4 records: 7
  IPv6 records: 3
```

Kuvio 32. JunRPKI:n ROA-tietueet AS-alueesta 3549

QuaggaSRX-reitittimelle asennetuissa paketeissa tuli mukana srxsvr_client-ohjelma, joka pystyttiin ajamaan SRx Server-komponentin ollessa päällä. Ohjelman käynnistämisen jälkeen sillä otettiin yhteys kyseiseen komponenttiin ja kysyttiin ROA-objektien Prefix Origin Validation-tiloja. Yhteys muodostettiin kuvion 33 mukaisesti.

```
>> connect
Host [default: 'localhost'] ?
Port [default: 17900] ?
Proxy-id [default: 10.0.0.1] ?
PeerAS [default: 50] ? 65400
PeerAS (enter for stop) ?
Connection to localhost is successfully established!
```

Kuvio 33. Yhdistäminen SRx Server-komponenttiin

Prefix Origin Validation -tiloja voitiin kysyä verify-komennolla. Testiksi päätettiin tiedustella testausverkossa olevien mainostuksien tiloja. Kuviosta 34 käy ilmi, kuinka 181.225.80.0/21 AS-alueen 3549 mainostamana saa tilan Valid. Kuviossa on suoritettu kysely ensin ilman lisämääreitä ja sen jälkeen lisämääreiden kanssa. Jälkimmäisessä kyselyssä on varmistettu, että AS-alue 3556 ei saa mainostaa verkkoa 181.225.80.0/21.

```
>> verify
(Verify) Local ID: [0=disable receipt] ?
(Verify) Method: [0=just store, 1=Origin only, 2=Path only, 3=both] ? 1
(Verify) AS number ? 3549
(Verify) IP Prefix [] ? 181.225.80.0/21
(Verify) DefOriginVal: [0=VALID, 1=UNKNOWN, 2=INVALID, 3=Not Defined] ? 0
(Verify) DefPathVal [0=VALID, 2=INVALID, 3=Not Defined] ? 3
(Verify) BGPSEC some string ?
>> => Received validation result for update [uid=0x55B80E90;lid:0x00000000]:  RO
A=VALID
>> verify 0 1 3556 181.225.80.0/21 0 3
(Verify) BGPSEC some string ?
>> => Received validation result for update [uid=0x72439362;lid:0x00000000]:  RO
A=INVALID
```

Kuvio 34. Prefix Origin Validation-tilojen kysely SRx Server-komponentilta

CiscoRPKI-reitittimellä ei havaittu ympäristön toiminnan aikana muutosta prosessorikuormassa ja oletettavasti lisääntynyt muistinkäyttö ei eronnut juuri Juniperin raportoimasta 3,6 megatavusta. Molemmat reitittimet käyttävät kuitenkin samaa RTR-protokollaa ja ovat yhteydessä samaan palvelimeen, jolloin tietueiden arvoissa ei pitäisi olla eroja.

JunRPKI:lla sen sijaan oli huomattavissa toimintavaikeuksia prosessorikuorman ollessa maksimissa koko ajan. Suurimman osan tehoista veivät Kernel-prosessit. Tämä

tuskin johtui RPKI-toiminnallisuudesta, vaan epäilys oli, että ympäristön virtualisoinnissa oli jotain epäkohtia käytetyn vMX-virtuaalireitittimen kanssa, mikä johti kasvaneeseen kuormaan. Rasitus ei vaikuttanut reitittimen ja palvelimen väliseen kommunikointiin ja kaikki Juniperin BGP-naapuruudet pysyivät vakaina koko testausympäristön olemassaolon ajan. Kuorma sen sijaan aiheutti dramaattisia viiveenvaihteluita verkon aktiivilaitteiden kohdilla. Ping-testillä tehdyt mittaukset osoittivat pahimmillaan sekuntien viiveitä. Ympäristön hyötyliikenne olisi varmasti kärsinyt, mikäli sellaista olisi ollut. Kuviossa 35 näkyy JunRPKI:n prosessorikuorma 4 minuuttia käynnistymisen jälkeen.

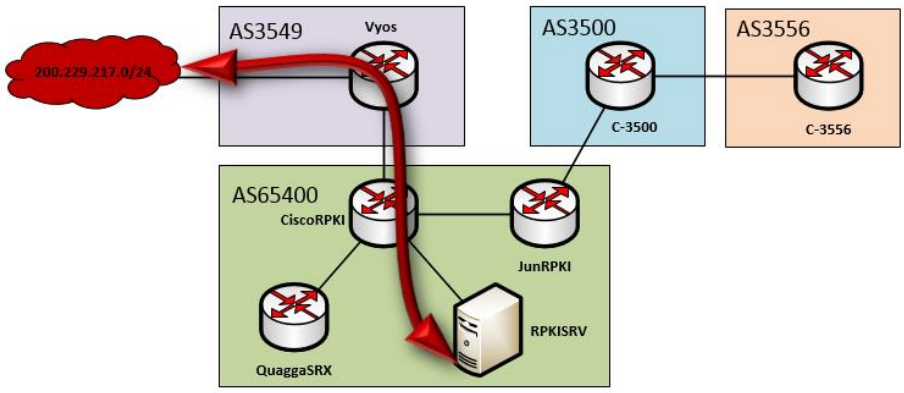
```
root@JunRPKI> show chassis routing-engine
Routing Engine status:
    DRAM                     2048 MB (2048 MB installed)
    Memory utilization         36 percent
    CPU utilization:
      User                     24 percent
      Background                0 percent
      Kernel                   76 percent
      Interrupt                 0 percent
      Idle                      0 percent
    Model                       RE-VMX
    Start time                  2015-11-05 07:56:11 UTC
    Uptime                      4 minutes, 39 seconds
    Last reboot reason          0x10:misc hardware reason
    Load averages:              1 minute   5 minute  15 minute
                                    2.35       1.40       0.65
```

Kuvio 35. JunRPKI:n prosessorikuorma

QuaggaSRX-reitittimen toiminnassa oli toivomisen varaa, sillä BGP-naapuruudet kuolivat useita kertoja testausympäristön ollessa toiminnassa. Toiminta-ajat vaihtelivat tunneista päiviin, mutta keskiarvona bgpd-prosessi piti käynnistää uudestaan kerran päivässä. Asia selittyy sillä, että kyseessä on vasta prototyyppi, joka on tarkoitettu eri BGP-tietoturvalaajennusten testaamiseen eikä suinkaan tuotantoverkkoihin.

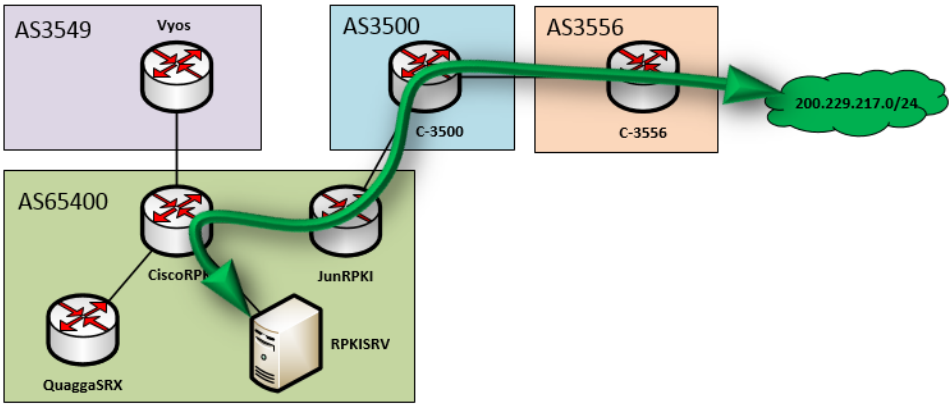
5.4 RPKI:n vaikutus reitityspäätöksiin

Pystytetyn ympäristön avulla testattiin, kuinka virheelliset reitit vaikuttavat reitittimien tekemiin reitityspäätöksiin. Samalla haluttiin todentaa, että RPKI-laajennuksen käyttöönotolla oli todistettavasti merkitystä. Testiverkoksi otettiin 200.229.217.0/24, jonka oikea omistaja AS-alue 3556 sijaitsee toisen AS-alueen 3500 takana, minkä takia sen AS-polun pituus on AS-alueelta 65400 tutkailtuna kaksi. Virheellisenä mainostajana käytettiin AS-aluetta 3549, joka sijaitsee suoraan AS-alueen 65400 vieressä, jolloin AS-polun pituudeksi tulee yksi. Koska AS-polun pituus on identtisten reittien kohdalla BGP:n oletusarvoilla tasatuloksen ratkaiseva arvo, pitäisi virheellisen mainostuksen tulla AS-alueen 65400 käytetyksi BGP-reitiksi ilman RPKI:ta. Testaus suoritettiin traceroute-komennolla RPKI-reitittimeltä 192.168.0.1, joka sijaitsee loogisesti CiscoRPKI-reitittimen takana. Testauksessa ei käytetty QuaggaSRX-reititintä. Kuviossa 36 näkyy hyötyliikenteen teoreettinen reitti kyseisessä skenaariossa.



Kuvio 36. Hyötyliikenteen reitti ilman RPKI-laajennusta

RPKI:n konfiguroimisen jälkeen virheellisen mainostuksen ei tulisi koskaan päästä parhaimmaksi reitiksi. Tämä puolestaan ohjaisi hyötyliikenteen sen oikealle omistajalle eli AS-alueelle 3556 ja palauttaisi verkon optimaalisen toiminnan. Teoreettinen liikennereitti on mallinnettu kuvioon 37.



Kuvio 37. Hyötyliikenteen reitti RPKI-laajennuksen jälkeen

Jotta käytetystä konfiguraatiosta päästiin tilaan, jossa reitittimet eivät hyödynnä Prefix Origin Validation-tiloja, tuli CiscoRPKI:lta poistaa rpkistate-sääntö kaikilta naapureilta sekä kytkeä RPKI-toiminnallisuus pois käytöstä seuraavasti:

no neighbor 10.0.1.1 route-map rpkistate in
no neighbor 172.16.0.1 route-map rpkistate in
no neighbor 172.16.0.5 route-map rpkistate in
bgp bestpath prefix-validate disable

JunRPKI:lla poistettiin pelkästään vastaanotettuja BGP-viestejä koskeva RPKI-sääntö:

delete protocols bgp group ext import RPKI
delete protocols bgp group int import RPKI

Näiden konfiguraatiomuutosten jälkeen CiscoRPKI oppi BGP-reititystauluunsa reitin verkkoon 200.229.217.0/24 pelkästään virheelliseltä mainostajalta eli AS-alueelta 3549 kuvion 38 mukaisesti.

```
CiscoRPKI#show ip bgp 200.229.217.0
BGP routing table entry for 200.229.217.0/24, version 1443
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
     29
  Refresh Epoch 1
  3549
    10.0.1.1 from 10.0.1.1 (10.0.1.1)
      Origin incomplete, metric 1, localpref 100, valid, external, best
      rx pathid: 0, tx pathid: 0x0
```

Kuvio 38. CiscoRPKI:n reittitieto ilman RPKI:ta

Syy, minkä takia verkkoalueen oikean omistajan eli AS-alue 3556:n reittitietoa ei näy Ciscon BGP-reititystaulussa, on JunRPKI:n kuviossa 39 näkyvä reitityspäätös. Se valitsi, että CiscoRPKI:n mainostama reitti kyseiseen verkkoon AS-alueen 3549 kautta on yhden hypyn lyhyempi kuin oikea reitti, jonka AS-polku on 3500-3556. Näin ollen JunRPKI ei mainosta oikean omistajan reittitietoa eteenpäin ja virheellinen reititys pääsee syntymään. Vaikka JunRPKI mainostaisi reittiä CiscoRPKI-reitittimelle, ei se muuttaisi mitään, sillä Cisco tulisi samaan lopputulokseen reittien paremmuudesta.

```
root@JunRPKI> show route 200.229.217.0

inet.0: 17 destinations, 20 routes (17 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

200.229.217.0/24   *[BGP/170] 00:00:33, MED 1, localpref 100
                      AS path: 3549 ?, validation-state: unverified
                    > to 172.16.0.2 via em3.0
                    [BGP/170] 00:30:12, localpref 100
                      AS path: 3500 3556 ?, validation-state: unverified
                    > to 10.0.2.2 via em2.0
```

Kuvio 39. JunRPKI:n reittitiedot ilman RPKI:ta

Reitittimien tekemät reitityspäätökset näkyvät myös traceroute-komennon tulosteessa. RPKI-palvelimelta 192.168.0.1 pyrittiin selvittämään reitti osoitteeseen 200.229.217.2. Kuviossa 40 nähdään, kuinka komentoon vastaava laite on AS-alueen 3549 reititin, jolta virheellinen reitti opittiin. Muistutuksena vielä, että virtualisoidun Juniperin toiminta todennäköisesti häiritsi koko testiverkon toimintaa, joten tulosteen viiveisiin ei tule kiinnittää huomiota.

```
[root@RPKISRV ~]# traceroute 200.229.217.2
traceroute to 200.229.217.2 (200.229.217.2), 30 hops max, 60 byte packets
 1  192.168.0.254 (192.168.0.254)  61.270 ms  73.118 ms  84.097 ms
 2  10.0.1.1 (10.0.1.1)  96.095 ms  110.009 ms  124.002 ms
 3  * * *
```

Kuvio 40. Traceroute RPKISRV:lta verkkoon 200.229.217.0/24

Tämän testin jälkeen aiemmin tehdyt konfiguraatiomuutokset kumottiin, jotta ympäristö toimi jälleen RPKI-toiminnallisuuden kanssa ja hyödynsi Prefix Origin Validation-

tiloja. Tulosteiden konkreettisuuden parantamiseksi CiscoRPKI:lla sallittiin väliaikaisest Invalid-tilaisten reittimainostusten lisääminen BGP-reititystauluun:

bgp bestpath prefix-validation allow-invalid

CiscoRPKI:n BGP-reititystauluun tuli tällä kertaa sekä virheellinen mainostus AS-alueelta 3549 sekä JunRPKI:n oppima reittitieto AS-alueelle 3556. Näistä oikea reittitieto valittiin parhaaksi ja lisättiin reititystauluun kuviossa 41 näkyvän Local Preference -arvon takia. Ilman Invalid-tilaisten reittien sallimista virheellistä mainostusta ei olisi tullut reititystauluun ollenkaan. On muistettava, että oletusarvoisesti Cisco lisää IBGP-naapureilta lisätyt mainostukset oletuksena Valid-tilaisina reititystauluunsa. Tämä tarkoittaa sitä, että mikäli JunRPKI olisi konfiguroitu väärin, niin virheellinen mainostus voisi saada CiscoRPKI:lla aivan väärän kohtelun olettaen, että oletusarvoa ei olisi kumottu.

```
CiscoRPKI#show ip bgp 200.229.217.0
BGP routing table entry for 200.229.217.0/24, version 1446
Paths: (2 available, best #1, table default)
  Advertised to update-groups:
     31
  Refresh Epoch 1
  3500 3556
    172.16.0.1 from 172.16.0.1 (10.0.2.1)
      Origin incomplete, localpref 200, valid, internal, best
      path 7F47698FA470 RPKI State valid
      rx pathid: 0, tx pathid: 0x0
  Refresh Epoch 1
  3549
    10.0.1.1 from 10.0.1.1 (10.0.1.1)
      Origin incomplete, metric 1, localpref 100, valid, external
      path 7F47698FAA70 RPKI State invalid
      rx pathid: 0, tx pathid: 0
```

Kuvio 41. CiscoRPKI:n reittitiedot RPKI:n kanssa

JunRPKI:lla ei näy BGP-reititystaulussa kuin AS-alueelta 3500 opittu reittitieto AS-alueelle 3556, sillä CiscoRPKI ei lisännyt virheellistä reittiä reititystauluunsa eikä näin ollen mainostanut sitä IBGP-naapurilleen. Kuviosta 42 käy ilmi, että reititin oli konfiguroitu oikein ja reittitiedon Prefix Origin Validation -tila Valid olisi pitänyt tällä kertaa paikkaansa myös Ciscon oletusasetuksilla.

```
root@JunRPKI> show route 200.229.217.0

inet.0: 17 destinations, 19 routes (14 active, 0 holddown, 3 hidden)
+ = Active Route, - = Last Active, * = Both

200.229.217.0/24   *[BGP/170] 00:45:08, localpref 200
                      AS path: 3500 3556 ?, validation-state: valid
                    > to 10.0.2.2 via em2.0
```

Kuvio 42. JunRPKI reittitiedot RPKI:n kanssa

Tehty hyötyliikenteen reitin testaus on yhteneväinen reitittimiltä saadun BGP-reititystiedon kanssa. Kuviossa 43 näkyvä RPKI-palvelimelta aloitettu traceroute kiertää ensin CiscoRPKI:lle ja sitä kautta JunRPKI:lle. Tämän jälkeen poistutaan AS-alueelta 65400 ja seuraava vastaava laite on AS-alueen 3500 reititin, minkä jälkeen viimeinen vastaava laite on reitin alkuperäinen mainostaja AS-alueen 3556 reititin.

```
[root@RPKISRV ~]# traceroute 200.229.217.1
traceroute to 200.229.217.1 (200.229.217.1), 30 hops max, 60 byte packets
 1  192.168.0.254 (192.168.0.254)  1.409 ms  1.469 ms  2.014 ms
 2  172.16.0.1 (172.16.0.1)  1.428 ms  1.405 ms  14.044 ms
 3  10.0.2.2 (10.0.2.2)  26.026 ms  36.936 ms  48.910 ms
 4  10.0.3.1 (10.0.3.1)  59.896 ms * *
```

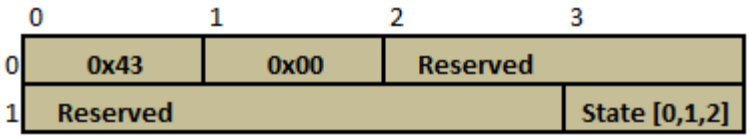
Kuvio 43. Traceroute RPKISRV:ltä verkkoon 200.229.217.0/24 RPKI-ympäristössä

Lopputuloksena voidaan huomata, kuinka Prefix Origin Validation -tilojen hyödyntäminen AS-alueen reitityspäätöksissä turvaa hyötyliikennettä suunnitellulla tavalla. Väittämä tosin pitää paikkaansa vain silloin, kun verkkoalueen omistaja on luonut ROA-objektin kyseiselle verkkoalueelle. Testauksen tulokset olisivat olleet verkolla 12.34.56.0/24 erilaiset, koska kyseistä verkkoaluetta ei ole sidottu mihinkään AS-alueeseen. Olettaen, että AS-alue 3556 omistaisi verkkoalueen 12.34.56.0/24, olisi virheellinen mainostus valittu parhaaksi reitiksi AS-alueella 65400 AS-polun pituuden takia.

5.5 Ciscon ja Juniperin eroavaisuudet Extended Community-arvoissa

Testausverkossa yritettiin saada JunRPKI- ja CiscoRPKI-kertomaan toisilleen niiden mainostamien reittitietojen Prefix Origin Validation -arvo, koska sillä voidaan kiertää Ciscon ohjelmiston oletus merkata IBGP-naapurilta opitut reitit Valid-tilaiseksi. Koska laitevalmistajien ohjeistuksilla toimintoa ei saatu toimimaan, päätettiin tilannetta tutkia Wireshark-pakettianalysaattorilla, sillä jostain syystä tilan arvon kertomiseen käytetyt Extended Community -kentät eivät tunnistuneet oikein.

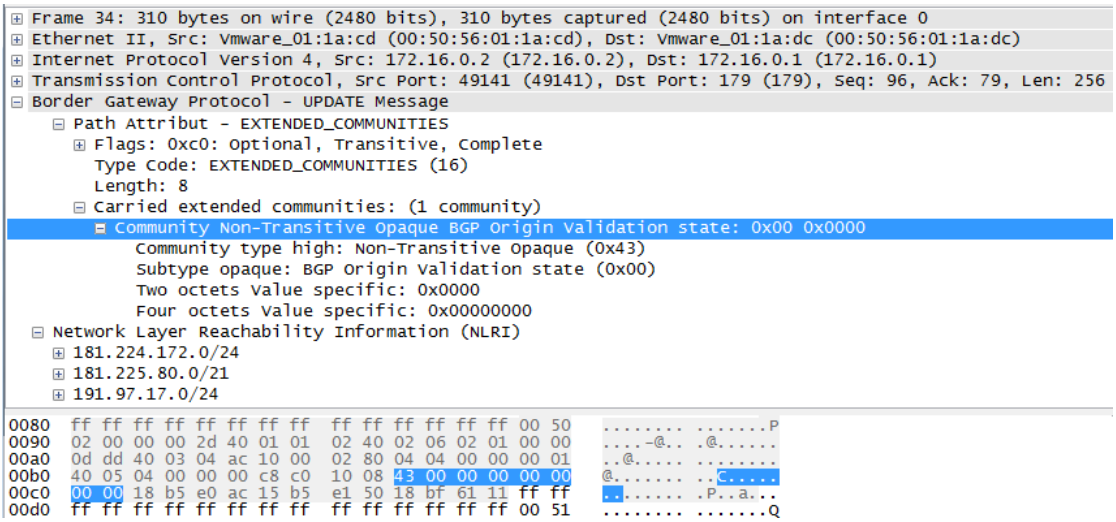
Tilatietojen kuljettamiseen käytetään yhteensä 8 tavun mittaista Opaque Non-transitive Extended Community -kehysrakennetta. Ensimmäisen 2-oktettisen Type-kentän arvo tulee olla heksadesimaaliarvoltaan 0x43, joka määrittää Extended Communityn olevan Non-transitive. Seuraava kahden oktetin mittainen Sub-Type-kenttä saa aina arvoltaan 0x00, mikä määrittää kehyksen sisältävän Prefix Origin Validation -tilan arvon. Seuraavat viisi oktettia ovat varattuja eikä niiden arvoja ole määritetty. Kehyksen viimeinen oktetti voi saada reitin tilasta riippuen seuraavat arvot: Valid (0), Not Found (1) tai Invalid (2). Kuviossa 44 näkyy käytetyn Extended Communityn kehysrakenne. (Mohapatra, Patel, Scudder, Ward & Bush 2015, 2-3.)



Kuvio 44. Prefix Origin Validation-tilan Extended Community

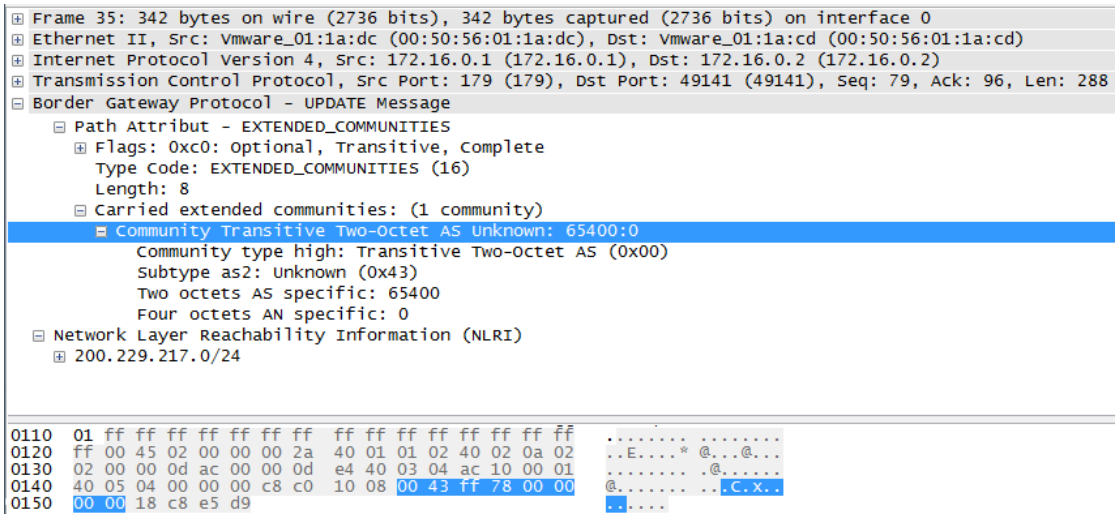
Kun CiscoRPKI:n BGP-konfiguraatioon oli määritetty JunRPKI-naapuruudelle Extended Community -arvojen käyttö (send-community extended) ja Prefix Origin Validation-tilan tarkistaminen (announce rpki state), voitiin pakettianalysaattorilla todeta merkkauksen olevan Mohapatran ym. (2015) määritysten mukainen. Kuviossa 45 näkyy CiscoRPKI:n mainostamien Valid-tilaisten reittitietojen 181.224.172.0/24, 181.225.80.0/21 ja 191.97.17.0/24 UPDATE-viestin pakettikaappaus. Kuvioon on

merkattu sinisellä korostuksella kehyksen heksadesimaaliarvot, jotka täsmäävät yllä-olevaan kuvioon 44.



Kuvio 45. Ciscon käyttämä Extended Community

JunRPKI:lla konfiguraatio tehtiin alun perin Juniperin materiaalissa olevan ohjeistuksen mukaisesti lisäämällä Valid-tilaisille reiteille Community nimeltä origin-validation-state-valid ja määrittämällä sille arvoksi 0x43:65400:0. Lopputuloksena tästä syntyi kuvion 46 mukainen virheellinen heksadesimaalimerkkaus Extended Community -arvoon Valid-tilaiselle verkkoalueelle 200.229.217.0/24. Kuvion alareunassa olevasta sinisellä korostetusta kehyksen heksadesimaalisisällöstä voidaan nähdä, kuinka arvo 0x43 on asetettu yhden oktetin liian myöhään eli Sub-Type-kenttään. Tämän takia Type-kentän arvoksi jää 0x00, joka tunnistuu Transitive Two-Octet AS-Specific Extended Communityksi. Rosenin ja Rekhterin (2014, 10) mukaan Sub-Type arvoa 0x43 ei ole määritetty kyseiselle Extended Communitylle, minkä takia kuvion Subtype as2 ei tunnistu ollenkaan. Loput heksadesimaaliarvoista koostuvat AS-alueesta 65400 ja arvosta nolla.



Kuvio 46. Juniperin käyttämä Extended Community virallisella ohjeella

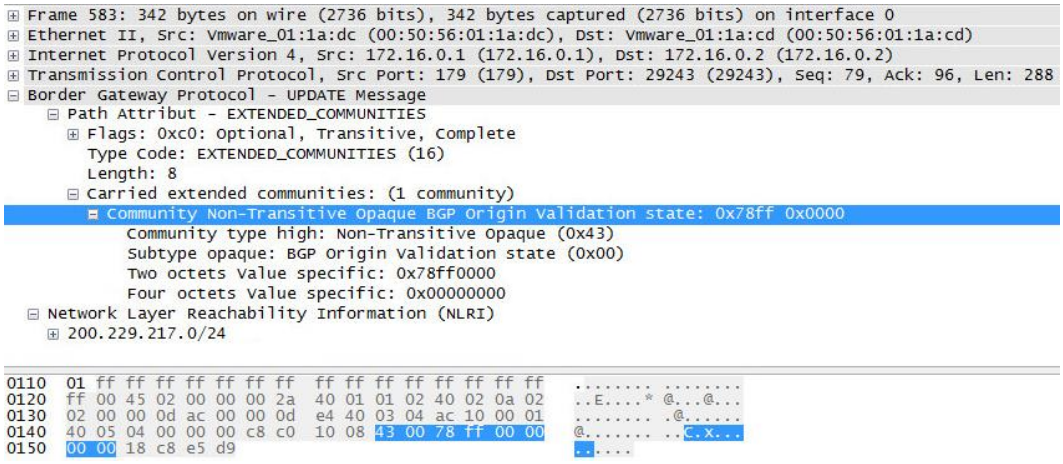
Virheelliset heksadesimaaliarvot aiheuttivat CiscoRPKI:n BGP-reititystaulussa kaikkien JunRPKI:lta opittujen reittien merkkaamisen Valid-tilaan, sillä Prefix Origin Validation-tilaa ei saatu tulkittua. Kuviossa 47 näkyy, kuinka niin opitut privaattiosoitteet kuin Juniperilla Not Found-tilainen 35.0.0.1/32 ovat Valid-tilassa ja niille on asetettu Local Preference 200.

```
CiscoRPKI#show ip bgp neighbors 172.16.0.1 routes
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
V*>i 10.0.2.0/30      172.16.0.1                200      0 i
V*>i 10.0.3.0/30      172.16.0.1             0  200      0 3500 ?
V*>i 35.0.0.1/32      172.16.0.1             0  200      0 3500 i
V* i 172.16.0.0/30    172.16.0.1                200      0 i
V*>i 200.229.217.0    172.16.0.1                200      0 3500 3556 ?

Total number of prefixes 5
```

Kuvio 47. Väärästä heksadesimaaliarvosta johtunut CiscoRPKI:n IBGP-reititystaulu

Ongelmasta ei löytynyt mainintaa missään lähteessä, joten eri arvoja lähdettiin testaamaan päättelyprosessin kera. Viimein kun Juniperin konfiguraatio-ohjeessa ollut 0x43 korvattiin arvolla 0x4300, saatiin heksadesimaaliarvot asetettua oikein. Testauksissa huomattiin, että myös käytetyn heksadesimaaliarvon desimaalinotaatiolla 17152 päästiin samaan tulokseen. Kuviosta 48 nähdään, kuinka JunRPKI asetti desi-

maaliarvon 65400 (0xFF78) Non-Transitive Opaque Extended Communityn ensimmäiseen varattu-tilaiseen kenttään. Lisäksi sen heksadesimaaliarvot olivat osittain käänteisessä järjestyksessä 0x78FF. Aiempaan kuvioon 46 verratessa voidaan nähdä, kuinka community-arvolla 0x43 AS-alue oli merkattu normaalissa heksadesimaalijärjestyksessä.



Kuvio 48. JunRPKI:n käyttämä Extended Community korjatulla konfiguraatiolla

Korjatun community-konfiguraation jälkeen CiscoRPKI osasi tulkata JunRPKI:n lähettämiä Extended Communityja. Kuviosta 49 nähdään CiscoRPKI:n tunnistama Extended Community reittitiedolle 200.229.217.0/24. Varattuihin oktetteihin asetettu AS-alue 65400 tunnistuu laitteella CiscoRPKI desimaaliarvoon 30975 (0x78FF), mutta tämä ei onneksi aiheuttanut ongelmia. Vaikuttaa siltä, että IANA:n määrittämättä jättämien oktettien arvot sivuutetaan.

```
CiscoRPKI#show ip bgp 200.229.217.0
BGP routing table entry for 200.229.217.0/24, version 298
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
     44          45
  Refresh Epoch 1
  3500 3556
    172.16.0.1 from 172.16.0.1 (10.0.2.1)
      Origin incomplete, localpref 200, valid, internal, best
      Extended Community: 0x4300:30975:0
      path 7F47698FA770 RPKI State valid
      rx pathid: 0, tx pathid: 0x0
```

Kuvio 49. CiscoRPKI:n tulkitsema Extended Community

6 LAAJENNUSTEN VERTAILU

6.1 BGP:n tietoturvaongelmien korjaaminen

Resource Public Key Infrastructure turvaa pelkästään sen, että reittitietoa alun perin mainostava AS-alue on verkkoalueen omistaja. Kuten tässä dokumentissa käsitellyssä implementaatiossa huomattiin, infrastruktuuri ei ottanut kantaa muuhun kuin, että Valid-tilaisen mainostuksen 200.229.217.0/24 alkuperäinen AS-alue oli 3556. Testausympäristössä mainostus tehtiin reitittimellä, joka ei ole kyseisen alueen ylläpitäjän hallinnoima eikä kyseinen reititin edusta AS-aluetta 3556. Mainostetun reitin AS-poluksi syntyi mielivaltaisen AS-aluevalinnan johdosta 3500-3556, mutta ei ole mitään takeita ovatko kyseiset alueet toistensa naapureita oikeassa maailmassa. Vaikka ne olisivatkin, niin saako 3500 mainostaa kyseistä verkkoa eteenpäin. On muistettava, että mainitut asiat eivät ole puutteita RPKI:n toiminnassa vaan tehtyjä suunnittelupäätöksiä.

Secure BGP:n toiminta pohjautuu RPKI:n tavoin PKI-infrastruktuuriin ja Address Attestation -objekteihin, millä voidaan osoittaa AS-alueen ja verkkoalueen sidos. Lisäksi se käyttää Route Attestation -objekteja, joiden allekirjoituksen takia niillä voidaan todentaa, että UPDATE-viesti tulee muuttumattomana BGP-naapurilta. Jokaiselle AS-polussa mainitulle alueelle löytyy viestistä RA-tietue, joka todistaa, että niillä alueilla on oikeus mainostaa kyseistä reittiä. AA-objektilla ja ensimmäisellä UPDATE-viestin RA-objektilla voidaan todistaa, että alkuperäisellä mainostajalla on oikeus verkon mainostamiseen verkkoalueen omistajalta. Koko AS-PATH-polku on siis mahdollista tarkistaa. IPsec-salauksen ESP-protokollaa käytetään BGP-naapureiden välisen kommunikoinnin suojaamiseen.

Secure Origin BGP käyttää EntityCert-, AuthCert- ja ASPolicyCert-todistuksia parantaakseen BGP:n tietoturvaa. EntityCert- ja AuthCert-todistuksilla luodaan verkko- ja

AS-alueen välinen sidos, jolla voidaan luotettavasti todistaa verkon alkuperäisen mainostajan oikeus kyseiseen mainostukseen. ASPolicyCert-todistuksella luodaan topologiakartta, joka todistaa, että mainostuksessa mainitulla AS-polulla päästään kohdeverkkoon.

BGP:n turvaamisen puolesta soBGP ei ole juuri parempi kuin RPKI. Molemmat laajennuksista pystyvät todistamaan alkuperäisen mainostajan oikeellisuuden, mutta soBGP:n toimintamekaniikka on monimutkaisempi käyttäen kahta sertifikaattia yksittäisen ROA-objektin sijaan. Lisäksi sen tarjoama kohdeverkon tavoitettavuuden takaaminen ASPolicyCert-todistuksella ei itse asiassa takaa, että käytetty AS-polku olisi sallittu tai optimaalinen. Reittiä alun perin mainostaneen BGP-reitittimen kytköstä AS-alueeseen ei myöskään tarkisteta. Naapuruudet suojataan IPsec-tunneleilla, joten epäsuorasti luotetaan, että alkuperäinen mainostaja oli IPsecin tuoman suojauksen takia luotettava taho.

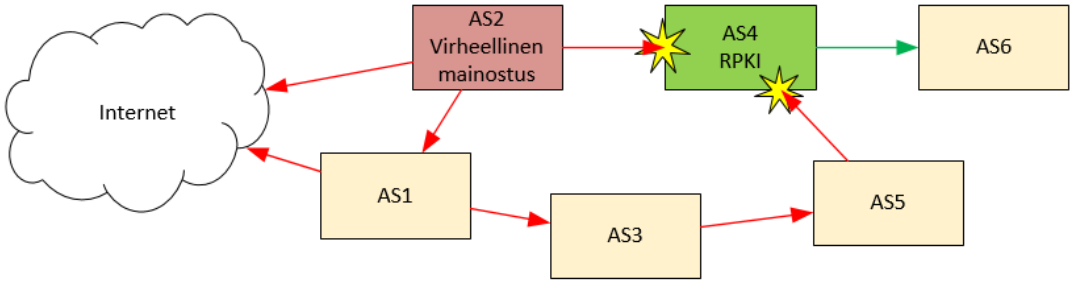
Interdomain Routing Validation varmistaa reittitiedon alkuperäisen mainostajan RPKI:n ROA-objekteja tai S-BGP:n AA-sertifikaatteja muistuttavalla digitaalisesti allekirjoitetulla todistuksella. Valitettavasti varsinaista spesifikaatiota kyseisestä todistuksesta ei ole määritetty. Vertailun vuoksi oletetaan, että käytetty todistus hyödyntää RPKI:n ROA-objekteja, jolloin ne perustuvat osoite- ja AS-alueresursseja myöntävään hierarkiaan ja alueelliset RIR:t ylläpitäisivät niitä.

Mainostuksessa mainitun AS-polun oikeellisuuden tarkastaminen hoituu AS-alueelle paikallisen IRV-palvelimen toimesta, jolle reunareitittimet kertovat saaneensa mainostukset. Jokaisella AS-alueella on oma IRV-palvelimensa, joka IBGP-naapuruuden avulla pitää tiedossaan kaikki reunareitittimien tekemät EBGP-mainostuksia. Mainostuksen vastaanottavan AS-alueen IRV-palvelin kysyy jokaisen AS-polussa mainitun AS-alueen IRV-palvelimelta, onko kyseinen alue mainostanut reittitietoa ulospäin. Mikäli on, niin tulee tarkistaa vielä, että mainostuksen kohde vastaa AS-polun tietoja. Koska IRV-palvelin on tietoinen kaikista AS-alueensa reunareitittimistä IBGP-naapuruuksien

takia, voidaan IRV:n avulla mainostanut reititin sitoa tiettyyn AS-alueeseen toisin kuin RPKI:ssa ja soBGP:ssä.

6.2 Asteittainen implementointi

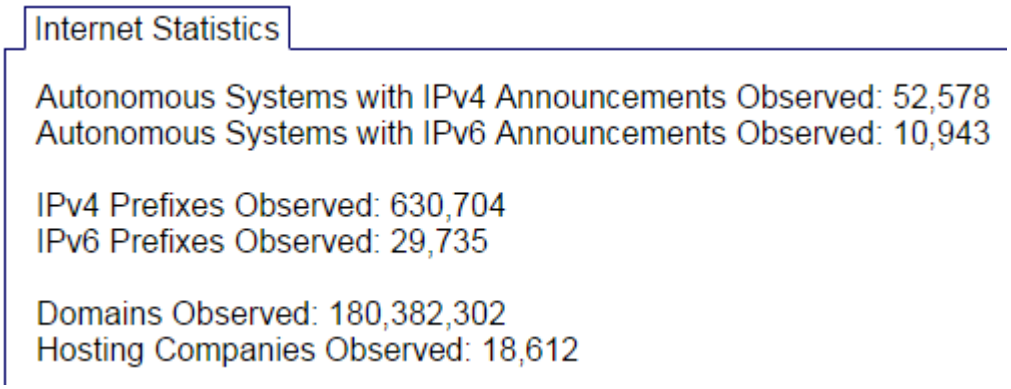
RPKI-ympäristön implementaatiossa kävi ilmi, että jokaisen globaalin toimijan ei tarvitse ottaa käyttöön BGP-reitittimilleen Prefix Origin Validationia. Mikäli virheellinen mainostus hylätään jonkin AS-alueen kohdalla, parantaa se epäsuorasti myös loogisesti kyseisen AS-alueen takana olevia toimijoita, koska mainostus ei tule missään vaiheessa heille asti. Täten RPKI:ta hyödyntämättömät AS-alueet eivät joudu tilanteeseen, jossa virheellinen mainostus voisi valikoitua parhaaksi reitiksi. Ongelmalta ei kuitenkaan vältytä, mikäli nämä AS-alueet vastaanottavat virheellisen mainostuksen kuvion 50 mukaisesti joltain muulta ei-RPKI-naapurilta. AS4 onnistuu suojelemaan omalla RPKI-implementaatiollaan vain AS6-aluetta. Virheellinen mainostus on voinut valikoitua jossain kohtaa verkkoa parhaaksi mainostukseksi tiettyyn verkkoon korvaten aidon mainostuksen. Näin AS4 ei koskaan saa aitoa mainostusta ja hylkäämällä virheellisen mainostuksen sen reititystauluun ei tule kyseistä verkkoa BGP:llä opittuna mistään. Ongelmaa ei pääse syntymään, mikäli kaikki verkon toimijat maailmanlaajuisesti hyödyntäisivät RPKI:ta.



Kuvio 50. Virheellisen BGP-mainostuksen leviäminen

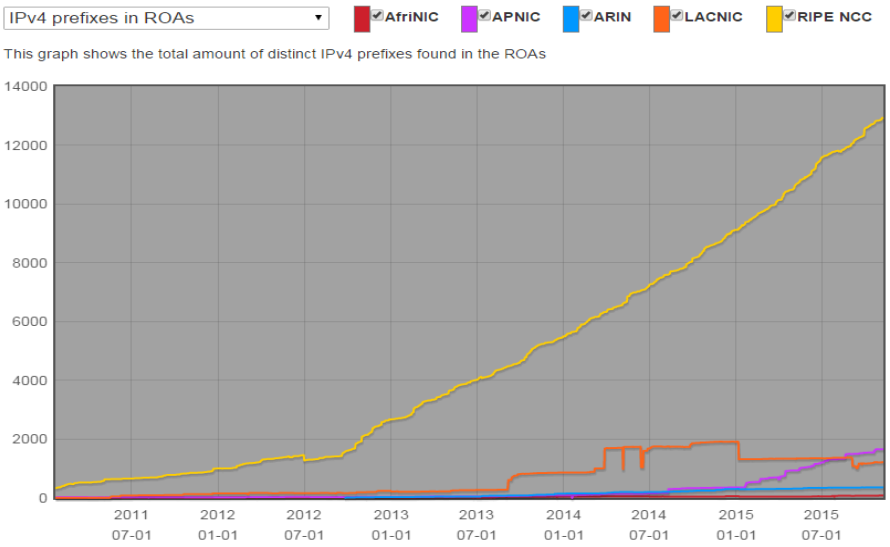
Enemmän painoarvoa on sillä, että liittyykö RPKI-infrastruktuuriin tarpeeksi monta palveluntarjoajaa maailmanlaajuisesti. Kuten toteutusosiossa havaittiin, tällä hetkellä

palvelin ja sitä myötä BGP-reitittimet oppivat Prefix Origin Validation-tilan 16370 IPv4-verkkoalueelle. Kuviossa 51 näkyy, kuinka palveluntarjoaja Hurricane Electric oppii BGP-naapureiltaan 630704 IPv4-verkkoaluetta. Globaalissa ympäristössä tämä tarkoittaa, että tällä hetkellä Prefix Origin Validation-tiloja voidaan hyödyntää 2,60 prosentille kaikista mainostetuista IPv4-verkoista. Hurricane Electricin sivuilta saatu lukema ei välttämättä käsitä kaikkia maailman mainostettuja julkisia verkkoalueita, mutta se tekee kokoluokan selväksi.



Kuvio 51. Hurricane Electricin BGP:llä oppimat IPv4-verkkoalueet (Hurricane Electric - Internet Statistics 2015)

Huolestuttavaa on, että RIPE:n ylläpitämä statistiikka ROA-objektien sisältämien IPv4-verkkoalueiden kasvusta näyttää hyvin maltilliselta. Koska alueelliset RIR:t synkronoivat ROA-tietokantoja keskenään, käy kuviosta 52 ilmi myös muiden toimijoiden määrät. Voidaan huomata, että ainoastaan RIPE:n asiakkaat ovat kasvavassa määrin luoneet ROA-objekteja omista verkkoalueistaan. Itse asiassa reilusti suurin osa 16370 objektista kuuluu tähän joukkoon. Suuri syy tähän on mahdollisesti se, että RIPE tarjoaa dokumentaatiota ja tässäkin työssä hyödynnetyn API:n RPKI-ympäristön pystytyksen ja testauksen avuksi. Viimeisen 12 kuukauden aikana IPv4-prefiksien määrä ROA-objekteissa on kasvanut vain noin 5000 kappaleella.



Kuvio 52. RIPE:n ylläpitämä statistiikka ROA-objektien verkkoalueista (RPKI Statistics 2015)

Kaikkien RIR-toimijoiden julkaisemien ROA-objektien määrä ja varsinkin uusien julkaisujen kiihtyvyyden määrän puute ei viittaa siihen, että RPKI-infrastruktuuria voitaisiin ottaa lähivuosina käyttöön. Tilanne voi toki muuttua räjähdysmäisesti uusien virheellisistä mainostuksista johtuvien reititysongelmien tapahtuessa tai mikäli jokin johtava taho alkaa ajamaan RPKI:n käyttöönottoa globaalilla tasolla.

Oletetaan skenaario, jossa osa palveluntarjoajista implementoisi RPKI-infrastruktuurin tuotantoverkkoonsa pitäen reitittimien muistissa kaikki julkaistut 16370 tila-arvoa. ROA-objektien määrän asettamien rajoitusten takia he pelkästään painottaisivat Valid-tilaisia reittitietoja muiden ylitse. Nykytilanteessa BGP-reitittimien toiminta ei juuri heikentyisi, sillä tila-arvojen tarkistamisen ei havaittu vaativan juuri laskentatehoa ja niiden säilyttäminen vei noin 3,6 megatavua keskusmuistia. Sitä mukaa, kun uusia ROA-objekteja julkaistaan muiden palveluntarjoajien toimesta, kasvaa muistissa olevan taulukon varaama keskusmuistimäärä. Ei tule myöskään unohtaa uusien IPv4- ja IPv6-verkkoalueiden allokoimisesta johtuvaa reitittimen globaalin reititystaulun varaaman keskusmuistin määrän kasvamista, joka on RPKI:sta riippumaton seikka. Yhdessä nämä kaksi muistissa säilöttyä taulukkoa voivat

ylittää reitittimen kriittisen pisteen, mikä johtaa koko laitteen kaatumiseen ja sitä myötä BGP-naapureiden reittitietojen mahdolliseen muuttumiseen, mikä jatkuu ketjureaktiona verkon läpi. Kriittinen piste riippuu jokaisen palveluntarjoajan käyttämästä laitekannasta.

Yksi RPKI:n ja Prefix Origin Validation -tilojen ongelma on se, että niillä ei voida kieltää yhtään AS-aluetta mainostamasta tiettyä verkkoaluetta. Toki verkkoalueen omistajuus voidaan yksiselitteisesti määrittää jollekin AS-alueelle ROA-objektilla, mutta silloin mainostusoikeus jää vielä kyseiselle AS-alueelle. Yksi kiertotapa tähän on määrittää verkkoalueen omistajuus AS-alueelle 0, joka ei ole hyväksyttävä AS-aluenumero. Täten yksikään oikea AS-alue ei voi mainostaa verkkoaluetta ilman, että Prefix Origin Validation -tilaksi tulee Invalid.

Secure BGP:n ja Secure Origin BGP:n tapauksessa implementaatioista ei ole vielä olemassa kuin teoreettista spekulaatiota. Molemmat näistä laajennuksista tulevat mahdollisesti pohjautumaan RPKI:n luomaan pohjaan, joten aiemmin käydyt rajoitukset tulee pitää mielessä. Esimerkiksi Secure BGP:n käyttämät AA-objektit ovat toiminnallisuudeltaan hyvin lähellä RPKI:n ROA-objekteja. Lisäksi RPKI määrittää palvelimien ja reitittimien välisen RPKI To Router -protokollan sekä mainostavan AS-alueen oikeellisuuden tiloja kuvaavan Prefix Origin Validationin.

Secure BGP vaatii naapuruuksien välille IPsec-tunnelin, mikä tarkoittaa, että globaalien palveluntarjoajien tulisi saada sovittua aikaikkunat toteutuksen implementointiin. Pahimmillaan puhutaan eri aikavyöhykkeillä ja eri kieltä käyttävistä elimistä. Lisäksi IPsec-yhteyden muodostamisen ja BGP-naapuruuden tunnelin läpi muodostamisen jälkeen vanhat yhteydet tulisi purkaa, mistä aiheutuu lyhyt liikennekatkos olettaen, että molempien osapuolien konfiguraatiot ovat kunnossa.

Isompi ongelma on, että Secure BGP vaatii toimiakseen AA- ja RA-objektit. Osoitealueen omistaja myöntää ensin AA-objektin haluamalleen toimijalle, jonka se valtuuttaa mainostamaan reittiä. Tämän jälkeen alkuperäisenä mainostajana toimiva reititin luo RA-objektin ja liittää sen reittimainostukseensa, jonka jälkeen vastaanottava reititin

luo RA-objektin ja niin edelleen. Route Origin Authorization -objektien hitaasta yleistymisestä voidaan päätellä, että AA-objektien yleistyminen ei tule tapahtumaan yhden yön aikana. Varsinkin, kun AA- ja ROA-objektit ovat rinnastettavissa toisiinsa joten molempien yleistyminen vaikuttaa epätodennäköiseltä. S-BGP:stä ei saada täyttä hyötyä, mikäli kaikki AS-polun AS-alueet eivät implementoi sitä. Reittimainostus voi kulkeutua S-BGP:tä käyttämättömälle toimijalle, jolloin siitä eteenpäin polun oikeellisuutta ei voida enää tarkistaa puuttuvan RA-tietueen takia.

Secure Origin BGP:n asteittainen implementointi ei juuri eroa RPKI:sta, mutta sen EntityCert-sertifikaattien leviäminen perustuu verkostomalliin eikä keskitettyyn malliin. Niinpä sellaiset tilanteet ovat mahdollisia, joissa mainostuksen vastaanottava soBGP-reititin ei pysty tarkistamaan AuthCert-sertifikaatin oikeellisuutta, koska siltä puuttuu pääsy tarvittavaan sertifikaattiin. Tässä tapauksessa reitittimen tulee hylätä reittitieto.

Asteittain implementoituna soBGP:n topologiakartasta saadaan täysi hyöty vain niiden reittien kohdalla, joiden AS-polku käsittää pelkästään soBGP-toimijoita. Mikäli kohdeverkon saavutettavuutta ei voida varmasti todentaa topologiakartalla, kuinka paljon arvoa saadaan sillä, että AS-polku hieman normaalia todennäköisemmin vie kohdeverkkoon. Sama pitää osittain paikkaansa myös alkuperäisen mainostajan tarkistamisen kohdalla, mutta siinä ollaan riippuvaisia vain kyseisestä toimijasta eikä kaikista AS-polun AS-alueista. Myös soBGP:n ongelmana on ROA-objekteissa huomattu yleistymisen hitaus, mikä rajoittaa vahvasti soBGP:n sertifikaattien verrattain nopeamman leviämisen todennäköisyyttä.

IRV:n iso ongelma on sen dokumentaation epätarkkuus. Avoimia kysymyksiä ovat esimerkiksi AS-alueen ja verkko-osoitteen sidoksen implementointi, mihin ei oteta alkuperäisessä dokumentissa mitään kantaa. Samoin IRV-palvelimien ja BGP-reunareitittimien välistä kommunikaatiota ei kuvata. Näistä välittämättä jälleen kerran asteit-

taisesta implementaatiosta saatava hyöty on suoraan verrannollinen IRV:tä käyttävien toimijoiden määrään. Kuten soBGP:n tapauksessakin, mikäli IRV-AS-alueiden ketju katkeaa AS-polun aikana, ei polun oikeellisuutta voida täysin varmentaa.

Toinen kompastuskivi liittyy IRV:n tapaan varmentaa AS-polun oikeellisuus. Sen toiminta vaatisi sitä, että kaikkien AS-alueiden IRV-palvelimet suostuisivat jakamaan mahdollisesti yrityksen tekemiin sopimuksiin liittyviä mainostustietoja niitä tiedustelevalle IRV-palvelimelle. Kysyvä palvelin ei välttämättä sijaitse viereisessä AS-alueessa, joten ei voida tietää liittyykö kysely varmasti mainostukseen. Jokaisen palveluntarjoajan tulisi myös olla halukas jakamaan UPDATE-viestinsä sisältämät verkot muille palveluntarjoajille, jotta IRV-ympäristö toimisi. IRV-palvelimien ylläpito kuuluu palveluntarjoajille ja on mahdollista, että kaikki IRV-palvelimet eivät ole aina tavoitettavissa. Dokumentaatio ei ota kantaa, kuinka näissä tilanteissa tulisi toimia.

6.3 Ohjelmistovaatimukset yleistymiselle

RPKI tarjoaa palveluntarjoajille suhteellisen yksinkertaisen implementaation hyödyntäen RIPE:n käyttämää työkalua, sillä salaisista avaimista ja sertifikaateista ei tarvitse palveluntarjoajan huolehtia ollenkaan. Tämä laskee implementaatiokynnystä varsinkin pienemmillä toimijoilla, mutta ei ole varsinaisesti asia, joka voidaan laskea RPKI:n eduksi. Tosimaailman toteutuksessa varsinkin isommat palveluntarjoajat haluavat ladata ROA-objektit omassa ylläpidossa olevalle tietokantapalvelimelleen hyödyntäen erillistä kommunikaatioyhteyttä alueellisen RIR-toimijan tai kolmannen osapuolen tietokantaan.

Tämän jälkeen ne voivat jakaa ROA-objektien Prefix Validation State-arvoja BGP-reitittimilleen hallitummin ja niitä voidaan suodattaa. Implementoitu RPKI Validator API-ohjelma nimittäin jakaa kaikki julkaistut ROA-verkkoalueet reitittimille ilman suodattamista. Lisäksi palveluntarjoajat todennäköisesti haluavat luoda oman Trust Anchor-yksikön, joka pystyy julkaisemaan paikallisesti merkittäviä ROA-objekteja,

joilla voidaan varmentaa AS-alueen käyttämien privaattiosoitteiden mainostuksia. Kyseiset ROA-objektit voidaan tämän jälkeen ladata tietokantapalvelimelle.

RPKI vaatii, että reitittimet tukevat RPKI To Router -protokollaa ROA-objektien tiivistelmien kommunikointiin palvelimen ja reitittimen välillä. Suurista laitevalmistajista tuki löytyy todennetusti jo Juniperilta ja Ciscolta. Varsinaiseen BGP-protokollaan ei tarvita muutoksia infrastruktuurin hyödyntämisessä EBGP-yhteksissä. IBGP-naapureiden välisessä Prefix Origin Validation -tilojen vaihtamisessa yksittäisen UPDATE-viestin koko kasvaa 8 oktettia, sillä tieto kuljetetaan Non-Transitive Opaque Extended Community -kehyksessä.

Secure BGP:n käyttämien RA-objektien allekirjoittaminen aiheuttaa sen, että BGP-reitittimien tulee tukea salausmoduuleita, jotka käyttävät prosessoritehoa. Lisäksi vastaanotetut objektit tulee säilöä reitittimen muistiin, joka on vielä kriittisempi resurssi. Muistissa tulee pitää myös AA-objektit, vaikka niitä ei välitetäkään BGP:n UPDATE-viesteissä. Uhkakuvaa voidaan verrata RPKI:n aiheuttamaan muistin loppumiseen, mutta huomattavasti kiihdytettynä. Kent, Lynn, Mikkelson ja Seo (2000, 11) arvioivat, että, mikäli S-BGP:n toimintaa ei optimoida, niin reitittimen muistissa pitämät sertifikaatit tulisivat viemään tuotantokäytössä 32 megatavua muistia, AA-objektit 10 ja RA-objektit 17. RA-objektien muistinkäyttö on ilmoitettu per naapuruus. Yhteensä tästä syntyisi nykyiseen BGP-toteutukseen verrattuna 40 megatavua eli 200 prosenttia lisää muistinkäyttöä yhdellä naapurilla. (Kent ym. 2000, 11.)

S-BGP tarvitsee toimiakseen lisäyksen BGP-protokollaan, sillä se määrittää uuden Optional Transitive Path Attribute-tyypin nimeltään ATTEST. IANA:n tulee myös määrittää sille numeroarvo ennen kuin sitä voidaan käyttää. Tämän lisäksi reitittimet tarvitsevat ohjelmistotuen AA- ja RA-objekteille ja niiden kanssa toimimiselle. Kirjoitushetkellä kaikki mainitut asiat ovat toteuttamatta.

Secure Origin BGP puolestaan vaatii toimiakseen kokonaan uuden BGP:n SECURITY-viestin, jonka tukeminen ohjelmistopuolella vaatii huomattavasti enemmän kuin uuden Path Attribute-kehyksen (S-BGP) tai uuden Extended Community-arvon (RPKI)

implementoiminen. Tähän saakka on BGP:n synnystä asti käytetty pääsääntöisesti vain OPEN-, UPDATE-, KEEPALIVE- ja NOTIFICATION-viestityyppejä. Tämä itsessään lisää implementoinnin kynnystä huomattavasti.

Reitittimien tulee Secure Origin BGP:n tapauksessa pystyä pitämään muistissaan AuthCert-, EntityCert-, ASPolicyCert- sekä PrefixPolicyCert-sertifikaatit. Vaikka tämä ei suoraan aiheuttaisi ongelmia muistinkäytön ja salaukseen käytettävien laskennallisten resurssien takia, voidaan niiden lukumäärää verrata RPKI-reitittimien muistissa pitämiin ROA-objektien Prefix Origin Validation -tilojen listaukseen. Vertaus on osuva, sillä näiden kahden laajennuksen tuottamat tietoturvaparannukset BGP:hen eivät juuri eroa toisistaan.

IRV ei vaadi toimiakseen päivityksiä tai muutoksia BGP-protokollaan toisin kuin muut tässä dokumentissa käsitellyt laajennukset, vaan se jättää kaiken laskennan IRV-palvelimille. Se luultavasti tarvitsee rautavalmistajilta kuitenkin jonkin asteista tukea, millä reitittimet ja IRV-palvelimet pystyvät juttelemaan keskenään. Koska kyseinen laajennus on julkaistu hyödyntämättä alan yleisesti käyttämää julkaisurajapintaa ja Request For Comments -dokumentteja, on laitevalmistajien tuki epätodennäköinen. Varsinkin, kun dokumentaatio ei ota kantaa siihen, mitä kyseisiltä valmistajilta vaadittaisiin ominaisuuksina. IRV:n vaatimiin reititinresursseihin ei oteta puutteellisen spesifikaation takia kantaa.

7 POHDINTA

7.1 Työn tavoitteet ja tulokset

Opinnäytetyön tavoitteena oli vertailla työn tekijän valitsemia BGP:n tietoturvalaajennuksia toisiinsa teoriapohjaan perustuen ja toteuttaa käytännön implementaatio soveltuvissa määrin. Tutkintakohteiksi valikoituivat Resource Public Key Infrastructure, Secure BGP, Secure Origin BGP sekä Interdomain Routing Validation, jotka keskittyvät BGP-protokollan alkuperäisen mainostajan ja AS-polun oikeellisuuden varmistamiseen. Ainoastaan RPKI-laajennus pystyttiin toteuttamaan testausympäristössä laitevalmistajien puuttuvan ohjelmistotuen takia. Ympäristön mahdollisti JYVSECTECin tarjoamat virtualisointiresurssit.

RPKI:n kohdalla saatiin todistettua käytännössä, että Route Origin Authorization -objektien arvoista johdetuilla Prefix Origin Validation -tiloilla voidaan vaikuttaa BGP-reitittimien tekemiin reitityspäätöksiin. Kyseisiin tiloihin perustuvilla säännöillä autonominen järjestelmä 65400 saatiin onnistuneesti suojattua AS-alueiden 3549 ja 3556 virheellisiltä reittimainostuksilta. RPKI:n toiminta ja konfiguraatio saatiin toteutettua Cisco-, Juniper- ja QuaggaSRx-reitittimillä. Tutkimus voi toimia arvokkaana tietopohjana vastaavan ympäristön pystyttäjälle.

Tutkimuksessa löydettiin myös epäkohtia laitevalmistaja Juniperin RFC:hen 6811 pohjautuvassa RPKI-impelementaatiossa. Prefix Origin Validation -tilasta Not Found käytettiin testatussa Junos OS-käyttöjärjestelmän versiossa 14.1R2.12 nimitystä Unknown. Lisäksi löydettiin RPKI-ympäristön toimintaan vaikuttava virhe Juniperin ohjeistuksessa BGP:n Extended Community-arvon määrittämiseksi. Oikea arvo saatiin selvitettyä soveltamalla.

7.2 Parannuskohteet ja puutteet

Tehtyä tutkimusta ei voida kuitenkaan pitää kokonaisvaltaisena RPKI-implementaationa, sillä autonominen järjestelmä 65400 ei ylläpitänyt omaa repositoriotaan, vaan tähän hyödynnettiin RIPE:n kehittämää RPKI Validator API-ohjelmaa. AS-alueen sisäisen Trust Anchor-palvelimen pystyttäminen jäi myös toteutuksen ulkopuolelle, joten AS-alue 65400 ei voinut luoda omia paikallisesti merkittäviä ROA-objektejaan. Kokonaisuutena implementaatiossa ei keskitytty PKI-infrastruktuuriin ollenkaan vaan pääpaino pidettiin laitevalmistajien BGP-ohjelmistotuen tutkimisessa. Painotuksesta huolimatta QuaggaSRx-ohjelmiston toimivuutta eri skenaarioissa ei tarkasteltu yhtä syvällisesti kuin Juniper- tai Cisco-reitittimen kohdalla. Yksi suurimmista puutteista oli virheellisen mainostuksen vastaanottamisen tutkiminen EBGP-naapurilta, sillä käytetyssä topologiassa Invalid-tilaiset reittitiedot eivät tavoittaneet kyseistä ohjelmistoreititintä CiscoRPKI:n ja JunRPKI:n tekemien alkusuodatusten takia.

Laajennusten välinen vertailu saatiin tehtyä suurimmilta osilta. Käytännön ohjelmistotuesta johtuen RPKI:sta saatiin parasta dataa, jonka perusteella päätelmiä ja perusteluita voitiin tehdä. Esimerkiksi julkaistujen ROA-objektien statistiikka ja toteutettu ympäristö olivat arvokkaita tiedonlähteitä. Kolmen muun laajennuksen kohdalla vertailu perustui enimmäkseen teoriapohjaan ja kirjoittajan tietämykseen alasta. Osa päätelmistä voitiin tosin pohjata RPKI:sta saatuun dataan, sillä varsinkin S-BGP ja soBGP muistuttavat perustasolla kyseistä laajennusta.

Laajennusten alkuperäinen valinta eli tutkimuksen lähtökohta oli suurimmaksi osaksi onnistunut. Jälkikäteen ajateltuna Interdomain Routing Validation -arkkitehtuuri olisi kannattanut joko sivuuttaa tai korvata jollain muulla tuotteella. Tarkemmassa tarkastelussa kyseisen laajennuksen dokumentaatiossa oli liikaa puutteita ja se pohjautui liian vahvasti S-BGP:n tekemiin valintoihin. RPKI, S-BGP ja soBGP puolestaan toimivat

hyvinä tutkimuskohteina tarjoten eri lähestymistapoja ja päämääriä. RPKI:n toiminnasta olisi mahdollista toteuttaa kokonaan oma tutkimustyönsä, sillä kaikkia osa-alueita ei ehditty työn laajuuden takia käsitellä.

LÄHTEET

ARIN - Resource Public Key Infrastructure N.d. RESOURCE PUBLIC KEY INFRASTRUCTURE (RPKI). Viitattu 17.11.2015. https://www.arin.net/resources/rpki/

BGP-Origin AS Validation N.D. Cisco BGP-Origin AS Validation. Viitattu 7.11.2015. http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_bgp/configuration/xe-3s/irg-xe-3s-book/irg-origin-as.pdf

Example: Configuring Origin Validation for BGP 2013. Example: Configuring Origin Validation for BGP. Viitattu 7.11.2015. https://www.juniper.net/techpubs/en_US/junos12.2/topics/topic-map/bgp-origin-as-validation.html

Fuller, V. & Li, T. 2006. Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan. Viitattu 11.11.2015. https://tools.ietf.org/html/rfc4632

Gaurab, R. N.d. BGP Best Practices for ISPs Viitattu 18.10.2015. http://archive.apnic.net/meetings/22/docs/tut-routing-pres-bgp-bcp.pdf

Goodell, G., Aiello, W., Griffin, T., Ioannidis, J., McDaniel, P. & Rubin, A. 2003. Working around BGP: An Incremental Approach to Improving Security and Accuracy in Interdomain Routing. Viitattu 13.10.2015. http://www.internetsociety.org/doc/working-around-bgp-incremental-approach-improving-security-and-accuracy-interdomain-routing

Heffernan, A. 1998. Protection of BGP Sessions via the TCP MD5 Signature Option. Viitattu 11.11.2015. https://tools.ietf.org/html/rfc2385

Huawei - Feature Description 2014. Huawei Secospace USG2100/2200/5100 BSR/HSR & USG2000/5000 V300R001 Feature Description. Luku 01-06 IP Routing. Kappale 6.7.4 BGP. Viitattu 18.10.2015. http://support.huawei.com/enterprise/docinforeader.action?contentId=DOC1000010135&partNo=100122

Hurricane Electric - 193.34.198.0/24. Hurricane Electric Network 193.34.198.0/24. Viitattu 18.10.2015. http://bgp.he.net/net/193.34.198.0/24

Hurricane Electric Internet Statistics. Hurricane Electric Internet Statistics. Viitattu 9.11.2015. http://bgp.he.net/report/netstats

JYVSECTEC - Tietoa meistä 2015. JYVSECTEC - Tietoa meistä. Viitattu 14.10.2015. http://jyvsectec.fi/fi/tietoa-meista/

JYVSECTEC - RGCE - Kybertoimintaympäristö 2015. RGCE - Kybertoimintaympäristö. Viitattu 14.10.2015. http://jyvsectec.fi/fi/kyberymparisto/

Kent, S., Lynn, C., Mikkelson, J. & Seo, K. 2000. Secure Border Gateway Protocol (S-BGP) - Real World Performance and Deployment Issues. Viitattu 10.11.2015. http://users.ece.cmu.edu/~adrian/731-sp04/readings/KLMS-SBGP.pdf

Kent, T. N.d. Securing the Border Gateway Protocol. Viitattu 12.10.2015. http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_6-3/securing_bgp_s-bgp.html

Lepinski, M., Chi, A. & Kent, S. 2012. Signed Object Template for the Resource Public Key Infrastructure (RPKI). Viitattu 12.10.2015. https://tools.ietf.org/html/rfc6488

Lepinski, M. & Kent, S. 2012. An Infrastructure to Support Secure Internet Routing. Viitattu 17.10.2015. http://tools.ietf.org/html/rfc6480

Lepinski, M., Kent, S. & Kong, D. 2012. A Profile for Route Origin Authorizations (ROAs). Viitattu 12.10.2015. https://tools.ietf.org/html/rfc6482

Lynn, C., Mikkelson, J. & Seo, K. 2003. Secure BGP (S-BGP) Viitattu 10.11.2015. https://www.ietf.org/archive/id/draft-clynn-s-bgp-protocol-01.txt

Lonvick, C. 2004. RADIUS Attributes for soBGP Support. Viitattu 12.10.2015. https://tools.ietf.org/html/draft-lonvick-sobgp-radius-04

McCullagh, D. 2008. How Pakistan knocked YouTube offline (and how to make sure it never happens again). Viitattu 18.10.2015. http://www.cnet.com/news/how-pakistan-knocked-youtube-offline-and-how-to-make-sure-it-never-happens-again/

Mohapatra, P., Patel, K., Scudder, J., Ward, D. & Bush, R 2015. BGP Prefix Origin Validation State Extended Community. Viitattu 8.11.2015. https://tools.ietf.org/html/draft-ietf-sidr-origin-validation-signaling-05#section-2

Mohapatra, P., Scudder, J., Ward, D., Bush, R. & Austein, R. 2013. BGP Prefix Origin Validation. Viitattu 12.10.2015. https://tools.ietf.org/html/rfc6811

Murhpy, S. 2006. BGP Security Vulnerabilites Analysis. Viitattu 18.10.2015. https://www.ietf.org/rfc/rfc4272.txt

Ng, J. 2004. Extensions to BGP to Support Secure Origin BGP (soBGP). Viitattu 12.10.2015. https://tools.ietf.org/html/draft-ng-sobgp-bgp-extensions-02

Postel, J. 1981a. Internet Protocol. Viitattu 1.8.2015. https://www.ietf.org/rfc/rfc791.txt

Postel, J. 1981b. Assigned Numbers. Viitattu 1.8.2015. https://www.ietf.org/rfc/rfc790.txt

Postel, J. 1981c. Transmission Control Protocol. Viitattu 1.8.2015. https://tools.ietf.org/html/rfc793

Rekhter, Y. & Li, T., Hares, S. 2006. A Border Gateway Protocol 4 (BGP-4). Viitattu 1.8.2015. https://www.ietf.org/rfc/rfc4271.txt

Rosen, E. & Rekhter, Y. 2014. IANA Registries for BGP Extended Communities. Viitattu 10.11.2015. https://tools.ietf.org/html/rfc7153

Router Configuration 2014. RIPE Router Configuration. Viitattu 7.11.2015. https://www.ripe.net/manage-ips-and-asns/resource-management/certification/router-configuration

RPKI Statistics 2015. RIPE RPKI Statistics - Alasvetovalikosta IPv4 prefixes in ROAs. Viitattu 9.11.2015. https://certification-stats.ripe.net/

Toonk, A. 2010. Chinese ISP hijacks the Internet. Viitattu 18.10.2015. http://www.bgpmon.net/chinese-isp-hijacked-10-of-the-internet/

Touch, J. 2007. Defending TCP Against Spoofing Attacks. Viitattu 18.10.2015. https://tools.ietf.org/html/rfc4953

Weis, B. 2006. Secure Origin BGP (soBGP) Certificates. Viitattu 12.10.2015 https://tools.ietf.org/html/draft-weis-sobgp-certificates-04

White, R. N.d. Securing BGP Through Secure Origin BGP. Viitattu 12.10.2015. http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_6-3/securing_bgp_sobgp.html

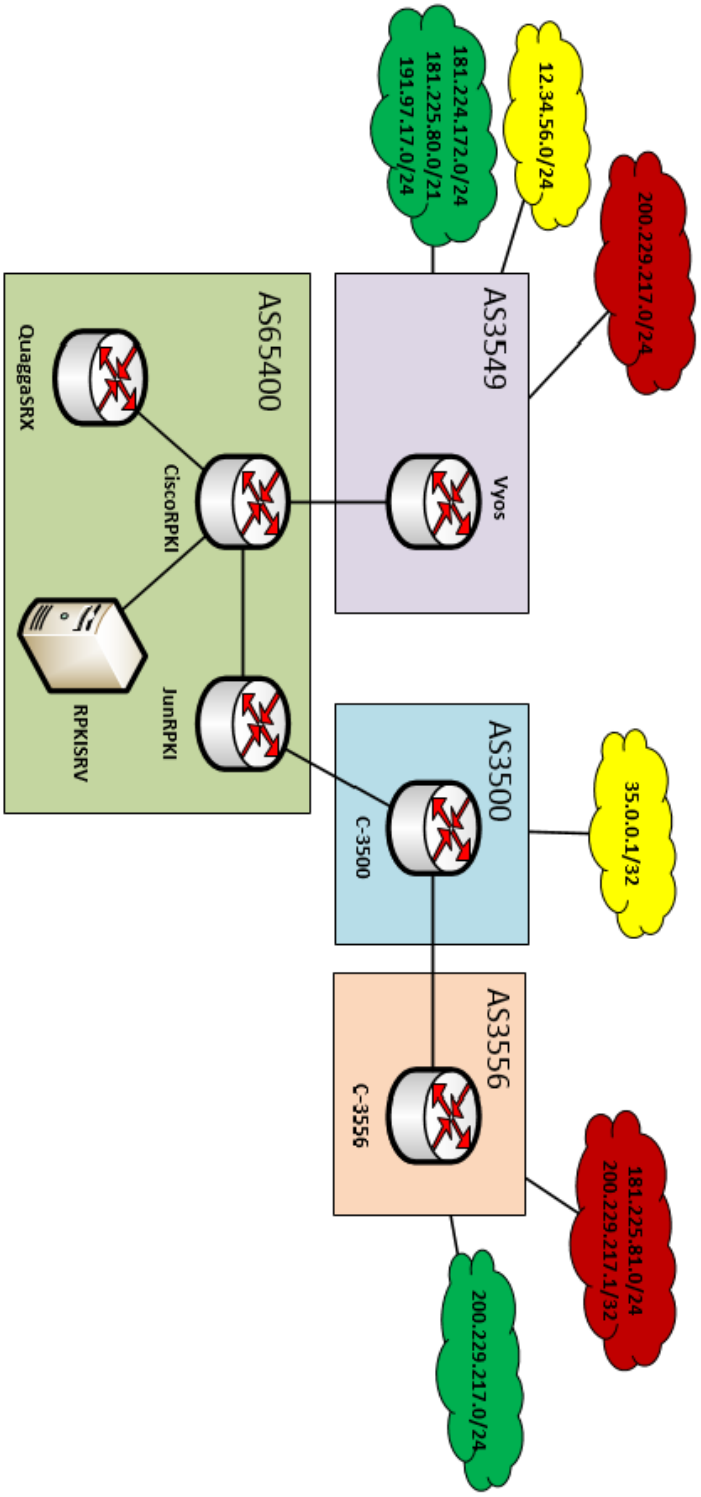
White, R. 2006. Architecture and Deployment Considerations for Secure Origin BGP (soBGP). Viitattu 12.10.2015. https://tools.ietf.org/html/draft-white-sobgp-architecture-02

LIITTEET

Liite 1. Implementaatiossa käytetty IP-osoitteistus

Laite	Rajapinta	Osoite	Maski	Vastapää
CiscoRPKI	Gi1	172.16.0.2	255.255.255.252	JunRPKI eM3
	Gi4	172.16.0.6	255.255.255.252	QuaggaSRX eth0
	Gi5	10.0.1.2	255.255.255.252	vyos eth0
	Gi6	192.168.0.254	255.255.255.0	RPKISRV eth0
JunRPKI	eM2	10.0.2.1	255.255.255.252	C-3500 Gi1
	eM3	172.16.0.1	255.255.255.252	CiscoRPKI Gi1
QuaggaSRX	eth0	172.16.0.5	255.255.255.252	CiscoRPKI Gi4
C-3500	Gi1	10.0.2.2	255.255.255.252	JunRPKI eM2
	Gi2	10.0.3.2	255.255.255.252	C-3556 Gi1
	lo0	35.0.0.1	255.255.255.255	Ei
C-3556	Gi1	10.0.3.1	255.255.255.252	C-3500 Gi2
	lo0	200.229.217.1.	255.255.255.255	Ei
vyos	eth0	10.0.1.1	255.255.255.252	CiscoRPKI Gi5
	eth1	200.229.217.1	255.255.255.0	Ei
	eth3	191.97.17.1	255.255.255.0	Ei
	eth4	12.34.56.1	255.255.255.0	Ei
	eth6	181.224.172.1	255.255.255.0	Ei
	eth8	181.225.80.1	255.255.248.0	Ei
RPKISRV	eth0	192.168.0.1	255.255.255.0	CiscoRPKI Gi6

Liite 2. Testausverkon topologia ja BGP-mainostukset



Liite 3. RPKISRV:n RPKI Validator API:n konfiguraatio

```
# You can edit this file to override default settings of the RPKI Validator, for example
# to use a different port for the HTTP and router interface, or the location of resources
# used by this application.
#
# By default the start script will expect this file at the following location:
#    conf/rpki-validator.conf
#
# Override the default name and location of this configuration file using the -c flag:
#    ./rpki-validator.sh start -c /path/to/my-configuration.conf


# Start the web user interface on the specified port.
ui.http.port=8080

# In kiosk mode the application will be accessible read-only to anyone, but any action or
# update will require authentication with a username and password.
ui.kiosk.enable=false
ui.kiosk.user=admin
ui.kiosk.pass=admin

# The delay after validating publishing objects for a trust anchor before the next validation
# occurs.
# Other values can be of the form "30m", "2d", "1w".
validation.interval = 10m

# Defines the usage of loose validation
# http://tools.ietf.org/html/draft-huston-rpki-validation-01
validation.loose=true

# Interval between runs of the job cleaning old objects
# from the local cache store. In case this setting is omitted
# the default is 7 days.
validation.remove_old_objects.interval = 7d

# Allow RPKI-capable routers to connect on the specified port.
rtr.port=8282

# Stop this application from sending 'notify' messages to the router when it has updates.
# When set to true, routers will fetch new data at the interval specified on the device.
rtr.send-notify=true

# Stop this application from closing connections when it receives fatal errors.
rtr.close-on-error=true

# Change the location of any of the files and working directories this application uses.
# All paths are relative to where the rpki-validator.sh script is installed.
locations.workdir=tmp
locations.datadir=data
locations.rsyncdir=data/rsync
locations.taldir=conf/tal
locations.trusted.ssl.dir=conf/ssl
locations.libdir=lib
locations.pidfile=validator.pid

logging.application.file=log/validator.log
logging.rtr.file=log/rtr.log
```

```
logging.access.file=log/access.log

# Use the following settings to change JVM parameters
#
# Change the minimum and maximum memory for the JVM
#
# Notes:
# - 1GB of memory is needed for the current size of the combined RPKI repositories
# - You may want to raise this value if you see 'out of memory' errors in the log
# - A higher maximum will allow the JVM to use more system memory and spend less time on
#   garbage collection (slight speed improvements possible)
jvm.memory.initial=1024m      # -Xms jvm option -> initial memory claimed by the jvm
jvm.memory.maximum=1024m      # -Xmx jvm option -> maximum memory for the jvm

# Proxy settings are used by the JVM when fetching data for the BGP Preview and
# notifications about new RPKI Validator releases.
#
# Notes:
# - rsync is used as an external program and will not use this proxy
# - you should only specify one type of proxy, if you specify both 'socks' is preferred
jvm.proxy.socks.host=''        # leave empty if you don't use a socks proxy
jvm.proxy.socks.port=''        # leave empty if you don't use a socks proxy

jvm.proxy.http.host=''         # leave empty if you don't use a http proxy
jvm.proxy.http.port=''         # leave empty if you don't use a http proxy
```

Liite 4. QuaggaSRX:n konfiguraatiotiedostot

SRx Server (/etc/srx_server.conf)

```
verbose  = true;
loglevel = 5;
#log     = "/var/log/srx_server.log";
sync    = true;
port    = 17900;

console: {
  port = 17901;
  password = "x";
};

rpki: {
  host = "192.168.0.1";
  port = 8282;
};

bgpsec: {
  host = "localhost";
  port = 50002;
};

mode: {
  no-sendqueue = true;
  no-receivequeue = false;
};

#mapping: {
#The configuration allows 255 pre-xonfigurations. client_0 is invalid
#  client_1  = "2";
#  client_10 = "10.0.0.1";
#  client_25 = "10.1.1.2";
#};
```

SRx Crypto API (/etc/srxcryptoapi.conf)

```
# Contains the name of the library that will be loaded.
#library_conf="bgpsec_openssl";
library_conf="testlib";

# Allows to specify the default key volt. Can be overwritten programatically.
key_volt = "/var/lib/bgpsec-keys/";

#debug type
debug-type = 3;

# this is the mapping information of the library to be loaded
bgpsec_openssl: {
  library_name = "libSRxBGPSecOpenSSL.so";

#
```

```
# The following method mappings allow to customize the mapping. in case the
# mapping is disables using the hash tag, misspelled, or missing at all the
# default mapping of method names as specified in the header file will be used.
# in case the default specified method names are not implemented, the
# SRxCryptoAPI wrapper functions are mapped.
#

# Minimum required functions to operate
  method_validate           = "validate";
  method_sign_with_key      = "sign_with_key";

# In case public key management is available
  method_isExtended         = "isExtended";
  method_extValidate        = "extValidate";
  method_registerPublicKey  = "registerPublicKey";
  method_unregisterPublicKey  = "unregisterPublicKey";

# In case private key management is available
  method_isPrivateKeyStorage  = "isPrivateKeyStorage";
  method_sign_with_id       = "sign_with_id";
  method_registerPrivateKey   = "registerPrivateKey";
  method_unregisterPrivateKey = "unregisterPrivateKey";
};

# Some other example configuration
testlib: {
  library_name="libSRxCryptoTestlib.so";

# Minimum required functions to operate
  method_validate           = "validate";
  method_sign_with_key      = "sign_with_key";

# In case public key management is available
  method_isExtended         = "isExtended";
  method_extValidate        = "extValidate";
  method_registerPublicKey  = "registerPublicKey";
  method_unregisterPublicKey  = "unregisterPublicKey";

# In case private key management is available
  method_isPrivateKeyStorage  = "isPrivateKeyStorage";
  method_sign_with_id       = "sign_with_id";
  method_registerPrivateKey   = "registerPrivateKey";
  method_unregisterPrivateKey = "unregisterPrivateKey";
};
```

QuaggaSRX (/etc/bgpd.conf)

```
hostname bgpd
password --POISTETTU--
enable password --POISTETTU--
log stdout
!
router bgp 65400
 bgp router-id 172.16.0.5
 neighbor 172.16.0.6 remote-as 65400

 ! SRx Basic Configuration Settings
 srx set-proxy-id 172.16.0.5
```

```
 srx set-server 127.0.0.1 17900
 srx keep-window 900
 srx evaluation origin_only
 srx extcommunity 1 only_ibgp
 srx display

 ! SRx Evaluation Configuration Settings
 srx set-origin-value undefined
 srx set-path-value undefined
 srx policy local-preference valid 200
 srx policy local-preference notfound 100
 srx policy prefer-valid
 srx policy ignore-invalid

 ! Connect to SRx-server
 srx connect
!
line vty
!
end
```

Liite 5. CiscoRPKI:n konfiguraatio

```
version 15.4
service timestamps debug datetime msec
service timestamps log datetime msec
no platform punt-keepalive disable-kernel-core
platform console auto
!
hostname CiscoRPKI
!
boot-start-marker
boot-end-marker
!
!
no logging console
enable password root
!
no aaa new-model
!
!

no ip domain lookup

!
!
!
!
subscriber templating
!
multilink bundle-name authenticated
!
!
---- JÄTETTY POIS VIRTUAALIREITITTIMEN LISENSSI ----
spanning-tree extend system-id
!
username admin privilege 15 secret 5 ---POISTETTU---
!
redundancy
 mode none
!
!
!
!
!
!
!
interface GigabitEthernet1
 description to JunRPKI
 ip address 172.16.0.2 255.255.255.252
 negotiation auto
!
interface GigabitEthernet4
 description To Quagga
 ip address 172.16.0.6 255.255.255.252
 negotiation auto
!
interface GigabitEthernet5
 description to VYOS
```

```
 ip address 10.0.1.2 255.255.255.252
 negotiation auto
!
interface GigabitEthernet6
 description to RPKI VALIDATOR
 ip address 192.168.0.254 255.255.255.0
 negotiation auto
!
router bgp 65400
 bgp log-neighbor-changes
 bgp rpki server tcp 192.168.0.1 port 8282 refresh 600
 redistribute connected
 neighbor 10.0.1.1 remote-as 3549
 neighbor 10.0.1.1 route-map rpkistate in
 neighbor 172.16.0.1 remote-as 65400
 neighbor 172.16.0.1 next-hop-self
 neighbor 172.16.0.1 send-community extended
 neighbor 172.16.0.1 announce rpki state
 neighbor 172.16.0.1 route-map rpkistate in
 neighbor 172.16.0.5 remote-as 65400
 neighbor 172.16.0.5 route-reflector-client
 neighbor 172.16.0.5 next-hop-self
 neighbor 172.16.0.5 send-community extended
 neighbor 172.16.0.5 announce rpki state
 neighbor 172.16.0.5 route-map rpkistate in
!
!
virtual-service csr_mgmt
!
ip forward-protocol nd
!
no ip http server
ip http secure-server
!
!
route-map rpkistate permit 10
 match rpki valid
 set local-preference 200
!
route-map rpkistate permit 20
 match rpki not-found
 set local-preference 100
!
!
!
control-plane
!
!
line con 0
 stopbits 1
line vty 0 4
 password --POISTETTU--
 login
 transport input telnet
!
!
end
```

Liite 6. JunRPKI:n konfiguraatio

```
version 14.1R2.12;
system {
    host-name JunRPKI;
    root-authentication {
        encrypted-password  --POISTETTU--
    }
    login {
        user remote {
            uid 2000;
            class super-user;
            authentication {
                encrypted-password --POISTETTU--
            }
        }
    }
    services {
        telnet;
    }
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any notice;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
    }
}
interfaces {
    em2 {
        description toAS3500;
        unit 0 {
            family inet {
                address 10.0.2.1/30;
            }
        }
    }
    em3 {
        description toCiscoRPKI;
        unit 0 {
            family inet {
                address 172.16.0.1/30;
            }
        }
    }
}
routing-options {
    autonomous-system 65400;
    validation {
        group rpkisrv {
            session 192.168.0.1 {
                refresh-time 600;
                hold-time 800;
```

```
            port 8282;
            local-address 172.16.0.1;
        }
    }
  }
}
protocols {
  bgp {
    group int {
      type internal;
      import RPKI;
      export ibgp;
      neighbor 172.16.0.2;
    }
    group ext {
      type external;
      import RPKI;
      export direct;
      peer-as 3500;
      neighbor 10.0.2.2;
    }
  }
}
policy-options {
  policy-statement RPKI {
    term valid {
      from {
        protocol bgp;
        validation-database valid;
      }
      then {
        local-preference 200;
        validation-state valid;
        community add origin-validation-state-valid;
        accept;
      }
    }
    term notfound {
      from {
        protocol bgp;
        validation-database unknown;
      }
      then {
        local-preference 100;
        validation-state unknown;
        community add origin-validation-state-unknown;
        accept;
      }
    }
    term invalid {
      from protocol bgp;
      then reject;
    }
  }
  policy-statement direct {
    term networks {
      from protocol direct;
      then accept;
    }
```

```
    }
    policy-statement ibgp {
        term connected {
            from protocol direct;
            then {
                next-hop self;
                accept;
            }
        }
        term nexthop {
            then {
                next-hop self;
                accept;
            }
        }
    }
    community origin-validation-state-unknown members 0x4300:65400:1;
    community origin-validation-state-valid members 0x4300:65400:0;
}
```

Liite 7. vyos:n konfiguraatio

```
interfaces {
    ethernet eth0 {
        address 10.0.1.1/30
        description ToCiscoRPKI
        hw-id 00:50:56:84:29:5d
    }
    ethernet eth1 {
        address 200.229.217.1/24
        description BogusFromAS3556
        hw-id 00:50:56:84:50:26
    }
    ethernet eth2 {
        hw-id 00:50:56:84:42:7a
    }
    ethernet eth3 {
        address 191.97.17.1/24
        description MaxLengthMorePrecise
        hw-id 00:50:56:84:49:fd
    }
    ethernet eth4 {
        address 12.34.56.1/24
        description RandomPrefix
        hw-id 00:50:56:84:7d:c8
    }
    ethernet eth5 {
        hw-id 00:50:56:84:6b:d7
    }
    ethernet eth6 {
        address 181.224.172.1/24
        description ValidPrefix
        hw-id 00:50:56:84:7d:8e
    }
    ethernet eth7 {
        hw-id 00:50:56:84:78:20
    }
    ethernet eth8 {
        address 181.225.80.1/21
        description ValidPrefix
        hw-id 00:50:56:84:14:db
    }
    ethernet eth9 {
        hw-id 00:50:56:84:24:f0
    }
}
protocols {
    bgp 3549 {
        neighbor 10.0.1.2 {
            remote-as 65400
            route-map {
            }
        }
        network 181.225.82.0/24 {
        }
        redistribute {
            connected {
            }
```

```
        }
    }
    static {
        route 181.225.82.0/24 {
            next-hop 181.225.82.1 {
            }
        }
    }
}
service {
    telnet {
        port 23
    }
}
system {
    config-management {
        commit-revisions 20
    }
    console {
        device ttyS0 {
            speed 9600
        }
    }
    login {
        user vyos {
            authentication {
                encrypted-password --POISTETTU--
                plaintext-password --POISTETTU--
            }
            level admin
        }
    }
    ntp {
        server 0.pool.ntp.org {
        }
        server 1.pool.ntp.org {
        }
        server 2.pool.ntp.org {
        }
    }
    package {
        repository community {
            components main
            distribution helium
            url http://packages.vyos.net/vyos
        }
    }
    syslog {
        global {
            facility all {
                level notice
            }
            facility protocols {
                level debug
            }
        }
    }
}
```

Liite 8. C-3500:n konfiguraatio

```
version 15.4
service timestamps debug datetime msec
service timestamps log datetime msec
no platform punt-keepalive disable-kernel-core
platform console auto
!
hostname C-3500
!
boot-start-marker
boot-end-marker
!
!
no logging console
enable password --POISTETTU--
!
no aaa new-model
!
!
!

no ip domain lookup

!
!
!
!
!
subscriber templating
!
multilink bundle-name authenticated
!
-----POISTETTU VIRTUAALIREITITTIMEN LISENSSI------
!
spanning-tree extend system-id
!
username admin privilege 15 secret ---POISTETTU---
!
redundancy
 mode none
!
!
!
!
!
!
!
interface Loopback0
 ip address 35.0.0.1 255.255.255.255
!
interface GigabitEthernet1
 description to JunosRPKI
 ip address 10.0.2.2 255.255.255.252
 negotiation auto
!
interface GigabitEthernet2
 description to AS3556
```

```
 ip address 10.0.3.2 255.255.255.252
 negotiation auto
!
interface GigabitEthernet3
 no ip address
 shutdown
 negotiation auto
!
router bgp 3500
 bgp router-id 35.0.0.1
 bgp log-neighbor-changes
 network 35.0.0.1 mask 255.255.255.255
 redistribute connected
 neighbor 10.0.2.1 remote-as 65400
 neighbor 10.0.3.1 remote-as 3556
!
!
virtual-service csr_mgmt
!
ip forward-protocol nd
!
no ip http server
ip http secure-server
!
!
!
!
control-plane
!
!
line con 0
 stopbits 1
line vty 0 4
 password --POISTETTU--
 login
 transport input telnet
!
!
end
```

Liite 9. C-3556:n konfiguraatio

```
version 15.4
service timestamps debug datetime msec
service timestamps log datetime msec
no platform punt-keepalive disable-kernel-core
platform console auto
!
hostname C-3556
!
boot-start-marker
boot-end-marker
!
!
no logging console
enable password --POISTETTU--
!
no aaa new-model
!
!
no ip domain lookup
!
!
!
!
!
subscriber templating
!
multilink bundle-name authenticated
!
-----POISTETTU VIRTUAALIREITITTIMEN LISENSSI-------
!
spanning-tree extend system-id
!
username admin privilege 15 secret 5 --POISTETTU--
!
redundancy
 mode none
!
!
!
!
!
interface Loopback0
 ip address 200.229.217.1 255.255.255.255
!
!
interface GigabitEthernet1
 description to AS3500
 ip address 10.0.3.1 255.255.255.252
 negotiation auto
!
interface GigabitEthernet2
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet3
```

```
 no ip address
 shutdown
 negotiation auto
!
router bgp 3556
 bgp log-neighbor-changes
 redistribute connected
 redistribute static
 neighbor 10.0.3.2 remote-as 3500
!
!
virtual-service csr_mgmt
!
ip forward-protocol nd
!
no ip http server
ip http secure-server
ip route 181.225.81.0 255.255.255.0 Null0
ip route 200.229.217.0 255.255.255.0 Null0
!
!
!
!
control-plane
!
!
line con 0
 stopbits 1
line vty 0 4
 password --POISTETTU--
 login
 transport input telnet
!
!
end
```

Liite 10. Reitittimien reititystaulut

CiscoRPKI

```
CiscoRPKI#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.0.1.0/30 is directly connected, GigabitEthernet5
L        10.0.1.2/32 is directly connected, GigabitEthernet5
B        10.0.2.0/30 [200/0] via 172.16.0.1, 00:01:00
B        10.0.3.0/30 [200/0] via 172.16.0.1, 00:01:00
      12.0.0.0/24 is subnetted, 1 subnets
B        12.34.56.0 [20/1] via 10.0.1.1, 01:58:01
      35.0.0.0/32 is subnetted, 1 subnets
B        35.0.0.1 [200/0] via 172.16.0.1, 00:01:00
      172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C        172.16.0.0/30 is directly connected, GigabitEthernet1
L        172.16.0.2/32 is directly connected, GigabitEthernet1
C        172.16.0.4/30 is directly connected, GigabitEthernet4
L        172.16.0.6/32 is directly connected, GigabitEthernet4
      181.224.0.0/24 is subnetted, 1 subnets
B        181.224.172.0 [20/1] via 10.0.1.1, 01:58:01
      181.225.0.0/21 is subnetted, 1 subnets
B        181.225.80.0 [20/1] via 10.0.1.1, 01:58:01
      191.97.0.0/24 is subnetted, 1 subnets
B        191.97.17.0 [20/1] via 10.0.1.1, 01:58:01
      192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.0.0/24 is directly connected, GigabitEthernet6
L        192.168.0.254/32 is directly connected, GigabitEthernet6
B     200.229.217.0/24 [200/0] via 172.16.0.1, 00:01:00
```

JunRPKI

```
root@JunRPKI> show route

inet.0: 16 destinations, 18 routes (14 active, 0 holddown, 2 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.1.0/30        *[BGP/170] 00:03:57, MED 0, localpref 100
                      AS path: ?, validation-state: unknown
                    > to 172.16.0.2 via em3.0
10.0.2.0/30        *[Direct/0] 03:10:30
                    > via em2.0
                    [BGP/170] 03:10:26, MED 0, localpref 100
                      AS path: 3500 ?, validation-state: unknown
                    > to 10.0.2.2 via em2.0
10.0.2.1/32        *[Local/0] 03:10:30
                      Local via em2.0
10.0.3.0/30        *[BGP/170] 03:10:26, MED 0, localpref 100
                      AS path: 3500 ?, validation-state: unknown
                    > to 10.0.2.2 via em2.0
12.34.56.0/24      *[BGP/170] 00:03:57, MED 1, localpref 100
                      AS path: 3549 ?, validation-state: unknown
                    > to 172.16.0.2 via em3.0
35.0.0.1/32        *[BGP/170] 03:10:26, MED 0, localpref 100
                      AS path: 3500 I, validation-state: unknown
                    > to 10.0.2.2 via em2.0
172.16.0.0/30      *[Direct/0] 03:10:29
                    > via em3.0
                    [BGP/170] 00:03:57, MED 0, localpref 100
                      AS path: ?, validation-state: unknown
                    > to 172.16.0.2 via em3.0
172.16.0.1/32      *[Local/0] 03:10:29
                      Local via em3.0
172.16.0.4/30      *[BGP/170] 00:03:57, MED 0, localpref 100
                      AS path: ?, validation-state: unknown
                    > to 172.16.0.2 via em3.0
181.224.172.0/24   *[BGP/170] 00:03:57, MED 1, localpref 200
                      AS path: 3549 ?, validation-state: valid
                    > to 172.16.0.2 via em3.0
181.225.80.0/21    *[BGP/170] 00:03:57, MED 1, localpref 200
                      AS path: 3549 ?, validation-state: valid
                    > to 172.16.0.2 via em3.0
191.97.17.0/24     *[BGP/170] 00:03:57, MED 1, localpref 200
                      AS path: 3549 ?, validation-state: valid
                    > to 172.16.0.2 via em3.0
192.168.0.0/24     *[BGP/170] 00:03:57, MED 0, localpref 100
                      AS path: ?, validation-state: unknown
                    > to 172.16.0.2 via em3.0
200.229.217.0/24   *[BGP/170] 03:10:26, localpref 200
                      AS path: 3500 3556 ?, validation-state: valid
                    > to 10.0.2.2 via em2.0
```

QuaggaSRX

```
bgpd# show ip bgp
BGP table version is 0, local router ID is 172.16.0.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Validation:    v - valid, n - notfound, i - invalid, ? - undefined
SRx Status:    I - route ignored, D - SRx evaluation deactivated
SRxVal Format: validation result (origin validation, path validation)
Origin codes: i - IGP, e - EGP, ? - incomplete

   Ident    SRxVal SRxLP Status Network          Next Hop            Metric  LocPrf Weight Path
*>iA9871DF2 n(n,-)   100,       10.0.1.0/30      172.16.0.6               0    100a      0 ?
*>i986F076F n(n,-)   100,       10.0.2.0/30      172.16.0.1                    100a      0 i
*>i0F9709BA n(n,-)   100,       10.0.3.0/30      172.16.0.1               0    100a      0 3500 ?
*>i41ADAC47 n(n,-)   100,       12.34.56.0/24    172.16.0.6               1    100a      0 3549 ?
*>i823AE0B4 n(n,-)   100,       35.0.0.1/32      172.16.0.1               0    100a      0 3500 i
*>i14ABD674 n(n,-)   100,       172.16.0.0/30    172.16.0.6               0    100a      0 ?
*>i49478778 n(n,-)   100,       172.16.0.4/30    172.16.0.6               0    100a      0 ?
*>iF0DB6D7A v(v,-)   200,       181.224.172.0/24 172.16.0.6               1    200a      0 3549 ?
*>iC0741CE3 v(v,-)   200,       181.225.80.0/21  172.16.0.6               1    200a      0 3549 ?
*>iAD3FCEB7 v(v,-)   200,       191.97.17.0/24   172.16.0.6               1    200a      0 3549 ?
*>iFDF38377 n(n,-)   100,       192.168.0.0      172.16.0.6               0    100a      0 ?
*>i5354D53E v(v,-)   200,       200.229.217.0    172.16.0.1                    200a      0 3500 3556 ?

Total number of prefixes 12
```

vyos

```
vyos@vyos:~$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

C>* 10.0.1.0/30 is directly connected, eth0
B>* 10.0.2.0/30 [20/0] via 10.0.1.2, eth0, 00:57:12
B>* 10.0.3.0/30 [20/0] via 10.0.1.2, eth0, 00:57:12
C>* 12.34.56.0/24 is directly connected, eth4
B>* 35.0.0.1/32 [20/0] via 10.0.1.2, eth0, 00:57:12
C>* 127.0.0.0/8 is directly connected, lo
B>* 172.16.0.0/30 [20/0] via 10.0.1.2, eth0, 04:26:38
B>* 172.16.0.4/30 [20/0] via 10.0.1.2, eth0, 04:26:38
C>* 181.224.172.0/24 is directly connected, eth6
C>* 181.225.80.0/21 is directly connected, eth8
S   181.225.82.0/24 [1/0] via 181.225.82.1 inactive
C>* 191.97.17.0/24 is directly connected, eth3
B>* 192.168.0.0/24 [20/0] via 10.0.1.2, eth0, 04:26:38
C>* 200.229.217.0/24 is directly connected, eth1
```

C-3500

```
C-3500#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
B        10.0.1.0/30 [20/0] via 10.0.2.1, 00:13:16
C        10.0.2.0/30 is directly connected, GigabitEthernet1
L        10.0.2.2/32 is directly connected, GigabitEthernet1
C        10.0.3.0/30 is directly connected, GigabitEthernet2
L        10.0.3.2/32 is directly connected, GigabitEthernet2
      12.0.0.0/24 is subnetted, 1 subnets
B        12.34.56.0 [20/0] via 10.0.2.1, 00:13:16
      35.0.0.0/32 is subnetted, 1 subnets
C        35.0.0.1 is directly connected, Loopback0
      172.16.0.0/30 is subnetted, 2 subnets
B        172.16.0.0 [20/0] via 10.0.2.1, 03:19:45
B        172.16.0.4 [20/0] via 10.0.2.1, 00:13:16
      181.224.0.0/24 is subnetted, 1 subnets
B        181.224.172.0 [20/0] via 10.0.2.1, 00:13:16
      181.225.0.0/16 is variably subnetted, 2 subnets, 2 masks
B        181.225.80.0/21 [20/0] via 10.0.2.1, 00:13:16
B        181.225.81.0/24 [20/0] via 10.0.3.1, 23:02:45
      191.97.0.0/24 is subnetted, 1 subnets
B        191.97.17.0 [20/0] via 10.0.2.1, 00:13:16
B     192.168.0.0/24 [20/0] via 10.0.2.1, 00:13:16
      200.229.217.0/24 is variably subnetted, 2 subnets, 2 masks
B        200.229.217.0/24 [20/0] via 10.0.3.1, 1d02h
B        200.229.217.1/32 [20/0] via 10.0.3.1, 1d02h
```

C-3556

```
C-3556#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
B        10.0.1.0/30 [20/0] via 10.0.3.2, 00:22:45
B        10.0.2.0/30 [20/0] via 10.0.3.2, 1d02h
C        10.0.3.0/30 is directly connected, GigabitEthernet1
L        10.0.3.1/32 is directly connected, GigabitEthernet1
      12.0.0.0/24 is subnetted, 1 subnets
B        12.34.56.0 [20/0] via 10.0.3.2, 00:22:45
      35.0.0.0/32 is subnetted, 1 subnets
B        35.0.0.1 [20/0] via 10.0.3.2, 1d02h
      172.16.0.0/30 is subnetted, 2 subnets
B        172.16.0.0 [20/0] via 10.0.3.2, 03:29:44
B        172.16.0.4 [20/0] via 10.0.3.2, 00:22:45
      181.224.0.0/24 is subnetted, 1 subnets
B        181.224.172.0 [20/0] via 10.0.3.2, 00:22:45
      181.225.0.0/16 is variably subnetted, 2 subnets, 2 masks
B        181.225.80.0/21 [20/0] via 10.0.3.2, 00:22:45
S        181.225.81.0/24 is directly connected, Null0
      191.97.0.0/24 is subnetted, 1 subnets
B        191.97.17.0 [20/0] via 10.0.3.2, 00:22:45
B     192.168.0.0/24 [20/0] via 10.0.3.2, 00:22:45
      200.229.217.0/24 is variably subnetted, 2 subnets, 2 masks
S        200.229.217.0/24 is directly connected, Null0
C        200.229.217.1/32 is directly connected, Loopback0
```