jamk

Verkon monitorointi ja reititystaulujen valvonta Nagios XI:llä

Erkki Honkaniemi

Opinnäytetyö, AMK
Huhtikuu 2022
Tietojenkäsittely ja tietoliikenne
Insinööri (AMK), tieto- ja viestintätekniikka

jamk | Jyväskylän ammattikorkeakoulu
University of Applied Sciences

jamk

Kuvailulehti

Honkaniemi Erkki

Verkon monitorointi ja reititystaulujen valvonta Nagios XI:llä

Jyväskylä: Jyväskylän ammattikorkeakoulu. Huhtikuu 2022, 57 sivua

Tietojenkäsittely ja tietoliikenne. Tieto- ja viestintätekniikan tutkinto-ohjelma. Opinnäytetyö AMK.

Julkaisun kieli: suomi

Verkkojulkaisulupa myönnetty: kyllä

Tiivistelmä

Työn toimeksiantajana oli Puolustusvoimien Järjestelmäkeskus, jonka vastuulla on teknisten järjestelmien elinkaaren ja tilannekuvien hallinta sekä poikkeusolojen toimintavalmius.

Toimeksiantona oli tarkastella verkon monitoroinnin toiminnallisuutta ja sen merkitystä nykyajan organisaatiossa. Työssä tutkittiin Nagios XI:n soveltuvuutta verkonvalvontasovelluksena ja tarkasteltiin erityisesti, kuinka sillä pystyttäisiin valvomaan reitittimen reititystaulun muutoksia. Työssä myös suunniteltiin ja toteutettiin toimiva monitorointijärjestelmä toimeksiantajan verkolle.

Tutkimus eroteltiin kahteen osaan. Ensimmäisessä osassa hyödynnettiin avointa aineistoa, joista koostettiin tutkimuksen määritysten mukainen pohja työlle alan artikkeleita ja kirjallisuutta hyödyntämällä. Toisessa osassa esiteltiin toteutettu ratkaisu ongelmaan, keskityttiin teknisiin ratkaisuihin ja esiteltiin toteutustapa, jolla toimeksianto suoritettiin.

Opinnäytetyön perusteella saatiin selville, että nykyajan verkolla tulee olla luotettava sekä hyvin optimoitu valvontajärjestelmä, jotta organisaatiolle voidaan taata turvalliset ja tuottavat olosuhteet toimia. Varsinkin, kun palvelut siirtyvät yhä enemmän sähköiseen muotoon, vaikuttavat katkokset yrityksen palveluissa negatiivisella tavalla sen imagoon. Työn tuloksia voidaan hyödyntää organisaation verkonvalvonnan suunnittelussa ja kehittämisessä. Tutkimus antoi tietoa Nagioksen tarjoamista ominaisuuksista yhtenä osana verkonvalvonnassa.

Avainsanat (asiasanat)

Verkonvalvonta, Nagios XI, monitorointi, reititystaulu

Muut tiedot (salassa pidettävät liitteet)

jamk

Description

Honkaniemi Erkki

Monitoring network and routing tables with Nagios XI

Jyväskylä: JAMK University of Applied Sciences, April 2022, 57 pages.

Information and Communication Technologies. Programme in Information and Communications Technology. Bachelor's thesis.

Permission for web publication: Yes

Language of publication: Finnish

Abstract

The employer for this study was the Joint Systems Center of the Finnish Defence Forces, whose responsibility is to maintain the lifecycle and situational awareness of technical systems, as well as their readiness in martial law.

The assignment was to inspect the functionality of monitoring a network, and its importance in a modern organization. In this project, the aim was to investigate the feasibility of Nagios XI as a network monitoring application and especially how it could be used to monitor the changes in routing table of a router. In this project, a functioning monitoring system was designed and implemented for the employer's network.

The study investigation was divided in two parts. In the first part, open data was utilized to construct a basis for the project according to the set limitations of the study, based on the field's articles and literature. In the second part, the implemented solution was presented, along with discussion about the technical solutions and presentation on the method of implementing the assigned task.

Based on this thesis, it was discovered that a modern network must have a reliable and well-optimized monitoring system to guarantee safe and productive conditions to operate in. Especially now that services are being digitalized in an ever-accelerating manner, outages in the services of the company have a more negative effect on its image. The results of this study can be utilized in designing and developing the network monitoring of an organization. The research also provided information about the features of Nagios as a part of network monitoring.

Keywords/tags (subjects)

Network monitoring, Nagios XI, routing table

Miscellaneous (Confidential information)

Sisältö

Kuviot

Lyhenneluettelo

API	Application Programming Interface
CCM	Core Config Manager
CPU	Central Processing Unit
FCAPS	Fault, Configuration, Accounting, Performance and Security
FQDNS	Fully qualified domain name
GUI	Graphical user interface
ICMP	Internet Control Message Protocol
ISO	International Organization for Standardization
LDAP	Lightweight Directory Access Protocol
MD5	Message-Digest
MIB	Management Information Base
NCPA	Nagios cross-platform agent
NMS	Network Management Station
NRPE	Nagios Remote Plugin Executor
NTP	Network Time Protocol
OID	Object Identifier
OSI	Open Interconnection Model
PDU	Protocol Data Unit
PERL	Practical Extraction and Report Language
PVLOGL	Puolustusvoimien logistiikkalaitos
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SSL	Secure Sockets Layer
TCO	Total cost of ownership
TCP/IP	Transmission Control Protocol / Internet Protocol
UDP	User Datagram Protocol
WAN	Wide Area Network

1 Johdanto

1.1 Työn taustat

Verkonvalvonnan tarkoitus on parantaa tietoliikenneverkon luotettavuutta, käytettävyyttä ja suorituskykyä ja havaita hallittavassa verkossa tapahtuvia muutoksia, jotta ylläpito voi reagoida muutoksiin tarvittavalla tavalla. Varsinkin suurikokoisessa verkossa on tärkeää olla oikein määritelty valvontaohjelmisto, jotta vianmääritys olisi nopeampaa ja näin käyttökatkokset pienempiä. Valvonta on automatisoitua toimintaa, jossa verkon laitteet ilmoittavat tilastaan ja muutoksista eri tekniikoita käyttäen. Ongelman sattuessa vika on paremmin kohdistettu ja vähentää näin ihmisen tekemää selvitystyötä.

Tämän työn tavoitteena oli tutkia monitorointi- ja valvontaohjelmistojen toiminnallisuutta ja niiden tärkeyttä nykyajan tietoliikenneverkoissa. Tehtävänä oli myös pystyttää toimeksiantajan verkolle monitorointi- ja valvontaohjelmisto ja rajata se informoimaan haluttuja toiminnallisuuksia. Verkolle haluttiin myös lisätä ominaisuus, joka valvoisi tiettyjen reitittimien reititystauluja ja ilmoittaisi niiden muutoksista. Työssä käsiteltiin erityisesti Nagios XI:n toimintaa ja sen soveltuvuuksia. Valmista työtä voitaisiin hyödyntää nykyisessä ympäristössä ja se olisi yhtenä työkaluna verkon valvonnassa. Palvelua olisi mahdollista myös lisä kehittää tulevaisuudessa.

1.2 Opinnäytetyön toteutus

Asiaa lähestyttiin palvelun kehittämisen näkökulmasta soveltavana tutkimuksena. Soveltavan tutkimuksen tavoitteena on tuottaa olemassa olevan uuden tiedon avulla jokin käytännön sovellus. Soveltavassa tutkimuksessa pääpaino on käytännön ongelmien ratkomisessa ja ratkaisujen löytäminen niihin. Sen tarkoituksena on lähtökohtaisesti pyrkiä saamaan teknistä sekä taloudellista hyötyä. (Tutkimus- ja kehittämistoiminta n.d.) Tutkimuskysymyksiksi rajattiin, mitä tehokas verkon valvonta sisältää, kuinka tärkeää verkon valvonta ja -monitorointi on ja kuinka hyvin Nagios XI soveltuu osana sitä.

Tutkimus on eroteltu kahteen osaan. Ensimmäinen osan tietoperustana hyödynnetään jo olemassa olevaa avointa aineistoa, joista koostetaan tutkimuksen määritysten mukainen pohja työlle.

Aineisto koostuu aiheeseen liittyvistä artikkeleista ja kirjallisuudesta. Ensimmäisessä osassa kuvataan verkon toiminnallisuutta lukijalle yleisesti ja asiaa lähestytään monitoroinnin ja valvonnan näkökulmasta. Toinen osa on käytännön osuus, jossa esitellään toteutettu ratkaisu ongelmaan. Toisessa osassa keskitytään teknisiin ratkaisuihin ja esitetään toteutustapa, jolla toimeksianto suoritettiin. Toimeksiantaja tarjosi valmiiksi ympäristön ja laitteiston, jossa työ toteutettiin. Työn toiminnallisuutta testattiin myös samassa ympäristössä, johon se on asennettu.

Opinnäytetyön ollessa julkinen, kohdeverkko esitellään yleisellä tasolla jättäen salassa pidettävät tiedot pois. Työssä näkyvät kuviot muutetaan myöskin vastaamaan sellaista muotoa, ettei niistä aiheudu haittaa organisaatiolle. Työ on kuitenkin tehty vastaamaan todellista tilannetta.

1.3 Toimeksiantaja

Opinnäytetyön toimeksiantajana toimi Puolustusvoimien Järjestelmäkeskuksen ilmajärjestelmäosasto. Järjestelmäkeskus kuuluu osana Puolustusvoimien logistiikkalaitosta (PVLOGL), joka on jakaantunut seitsemään eri hallintoyksikköön, jotka vastaavat puolustusvoimien yhteisistä logistiikan palveluista, suorituskyvyn rakentamisesta, sekä niiden ylläpidosta ja käytöstä. Logistiikkalaitos on pääesikunnan alainen yksikkö ja sen esikunta toimii Tampereella. Sillä on toimipaikkoja ympäri Suomea noin 40 eri kunnassa ja se työllistää yli 2 200 työntekijää. (Puolustusvoimien logistiikkalaitos 2021.)

Järjestelmäkeskuksen vastuulla on teknisten järjestelmien koko elinkaaren ja tilannekuvien hallinta ja poikkeusolojen toimintavalmius. Järjestelmäkeskus johtaa puolustusvoimien merkittävimpiä materiaaliprojekteja. Se jakaantuu neljään eri toimipisteeseen Tampereelle, Turkuun, Riihimäelle, sekä Tikkakoskelle. Yhteensä näissä toimipisteissä on henkilökuntaa noin 500. (Järjestelmäkeskus n.d.)

2 Verkon monitorointi

Nykyajan verkot ovat kehittyneet huomattavasti eteenpäin niistä ajoista, jolloin niissä oli vain kourallinen tärkeimpiä elementtejä. Kun verkot alkavat skaalautua langallisiin, langattomiin ja

virtuaalisiin ympäristöihin, muuttuu hallinta vaativammaksi ja yhä monimutkaisemmaksi. Palvelimissa ilmenevät ongelmat ja niistä johtuvat katkot voivat tulla organisaatiolle kalliiksi ja vaikuttaa negatiivisella tavalla imagoon asiakkaiden näkökulmasta. (Andini 2022.) Tutkimusten perusteella on arveltu, että verkon katkosten keskihinta nousee jopa 5600 dollariin minuutilta. Rahalliset tappiot vaihtelevat, kun otetaan huomioon tulot, toimiala, katkon todellinen kesto, vaikutuspiiriin vaikuttaneiden ihmisten määrä ja kellonaika. (Lerner 2014.)

Verkon seurantatyökalujen avulla järjestelmänvalvojat voivat tietää verkon yleisestä tilasta, suorituskyvystä ja mahdollisista ongelmista. Verkon valvonnan ja hallinnan helpottamiseksi reaaliaikaiset verkkotilastot ovat erittäin tärkeitä. Verkonvalvontaprotokollien tehtävänä on tarjota olennaisia tilastoja ja tärkeitä tietoja erilaisista verkon toiminnoista. Ne on suunniteltu helpottamaan verkkolinkkien (isäntä ja asiakas) ja sieltä lähtevän tiedon ja liikenteen seurantaa. Verkonvalvontatyökalujen vakioprotokollia käyttäen keräämät tiedot näytetään graafisesti, joita järjestelmänvalvojat voivat käyttää verkon toiminnan hallinnassa. (Definitive guide to network monitoring n.d; Network monitor success success in 8 easy steps n.d.)

Markkinoilla on useita eri verkonhallinta vaihtoehtoja, joka aiheuttaa yrityksille paljon päänvaivaa. Yritykset maksavat suuria summia pakettimuotoisista verkonhallintajärjestelmistä, joissa on useita olennaisia toimintoja, mutta myös ominaisuuksia, joita yritys ei todennäköisesti koskaan tule käyttämään. Budjetissa pysyäkseen verkonvalvojat todennäköisesti etsivät edullisempia ratkaisuja sen sijaan, että ostaisivat "yritystason verkkopakettijärjestelmiä", jotka veloittavat enemmän. Yksi parhaista tavoista on investoida verkonhallintatyökaluihin, joilla on korkea sijoitetun pääoman tuotto ja alhaiset kokonaiskustannukset. (Network Monitoring: Protocols, Best Practices, and Tools 2020.) IT-yritykset käyttävät tässä apuna Total cost of ownership -laskentamallia (TCO), jossa lasketaan ostohintaa ja käyttökustannuksia (Twin 2021). Siksi onkin erityisen tärkeää tehdä suunnitelma mitä aiotaan valvoa ja miksi.

2.1 Valvonnan osa-alueet, FCAPS -malli

Valvonnan muututtua enemmän proaktiiviseksi, kehiteltiin sille standardi kuvastamaan koko verkonhallintaa, johon on yhdistetty sekä verkonhallinta, että -valvonta. Tämän standardin loi International Organization for Standardization (ISO) Open Interconnection Model (OSI) -ryhmän

johtamana jo 1980-luvulla. Nykymuotoonsa sen on muokannut ITU-T (International Telecommunication Union) ja standardi kulkee nimellä X.700. (ISO/IEC 7498-4:1989, 1-3.) Standardi on määritelty televerkkojen kehysrakenteeseen, mutta pätee silti tietoverkoissa yleisesti.

ITU-T:n standardissa käytetään FCAPS-mallia, jota kutsutaan usein myös termeillä ISO network management model. Se jaottelee valvonnan viiteen eri osa-alueeseen, jotka ovat seuraavat (ISO/IEC 7498-4:1989, 4.):

Fault Management – Vikatilanteiden hallinta

Vikatilanteet jaetaan ohimeneviin ja pysyviin tapahtumiin. Siinä järjestelmä ei pysty täyttämään sille määriteltyjä tavoitteitaan ja näin ollen syntyy virhe (engl. error). Vianhallinta lähtee ongelman havaitsemisesta, sen rajoittamisesta ja loppuen ympäristön korjaamiseen. Hallinta vaatii tietoliikenteen jatkuvaa seuraamista, jotta virheistä voidaan ilmoittaa.

Configuration Management – Asetuksien/kokoonpanon hallinta

Tärkeimpänä tehtävänä on kokoonpanon laitteiden asetusten hallinta. Se ylläpitää laitteiden välisiä riippuvuuksia, varmistaa niiden asentamisen, jatkuvan toiminnan ja päivittämisen sekä poiston. Kokoonpanon hallinta on yksi tärkeimmistä ominaisuuksista, sillä se luo perustan kaikille muille toiminnoille verkonhallinnassa.

Accounting management – Kirjanpito/Käytön laskenta

Käyttäjätietojen seuranta tilastointiin ja analysointiin. Näitä tietoja käytetään yleisesti laskutuksessa asiakkaalle. Voittoa tavoittelemattomissa organisaatioissa "hallinta" korvaa "kirjanpidon". Hallinnan tavoitteena on hallita käyttäjien salasanoja ja käyttöoikeuksia sekä hallita laitteiden toimintaa, kuten suorittamalla ohjelmistojen varmuuskopiointia ja synkronointia.

Performance management – Suorituskyvyn hallinta

Nimensä mukaisesti keskittyy hallitsemaan verkon suoritustehoa. Suorituskyvyn hallinta sisältää toimintoja, joilla kerätään tilastotietoja, ylläpidetään lokeja järjestelmän tilasta ja kerätään tietoa välityskyvystä, käyttöasteesta, liikennemääristä ja vasteajoista. Pyrkimyksenä optimointi, ennakointi ja mahdollisten pullonkaulojen todentaminen.

Security management – Turvallisuuden hallinta

Turvallisuuden hallinnan tarkoituksena on tukea tietoturvapolitiikan soveltamista toimintojen avulla, jotka sisältävät turvapalvelujen ja -mekanismien luomisen, poistamisen ja valvonnan. Se vastaa myös turvallisuuden kannalta olennaisten tietojen jakelun ja turvallisuuden kannalta merkittävien tapahtumien raportoimisen.

2.2 Monitoroitavat kohteet

Valvontajärjestelmät keräävät ja käyttävät tietoa verkkoelementeistä erilaisiin valvontaan liittyviin toimintoihin. Verkot tarvitsevat myös jatkuvaa valvontaa varmistaakseen, että ongelmat havaitaan ennen kuin ne aiheuttavat verkkokatkoksia. Jatkuva seuranta johtaa suurien tietomäärien kerääntymiseen, joka voi johtaa valvontaratkaisun suorituskyvyn hidastumiseen, koska analysoitavaa tietoa on enemmän tarvittavien raporttien luomiseksi. Se lisää myös valvontatietojen tallentamiseen vaadittavaa kapasiteettia, mikä lisää kokonaiskustannuksia. Myöskin vianetsintä on loogisesti hitaampaa, mitä enemmän dataa tallennetaan. Tässä esiin nousee, kuinka tärkeää on suunnitella se, mitä halutaan monitoroida, jotta valvonta pysyy tehokkaana.

Solarwindsin julkaisun mukaan yleisimpiä verkon monitoroitavia kohteita ovat palvelimien prosessorien, muistien ja tallennustilojen käyttöasteet. Reitittimien ja kytkimien monitoroitavia aiheita ovat tyypillisesti käytettävyysaika, saatavuus ja vasteaika, jotka muodostavat yleiskuvan palvelun suorituskyvystä. (Ultimate guide to network monitoring 2019). Kohteiden seurantamenetelmät taas luokitellaan kahteen eri kategoriaan, aktiiviseen ja passiiviseen monitorointiin (Charbonneau 2020).

2.3 Raportointi ja hälytykset

Verkkovalvonnassa raportointi tarkoittaa tiedon keräämistä, kerättyjen tietojen käsittelyä ja niiden esittämisestä käyttäjälle ymmärrettävässä muodossa. Raportointi auttaa verkon valvojaa hahmottamaan suorituskykyä, verkon nykyisen tilan ja sen, mikä verkossa on normaalia. Raporttien tietojen avulla järjestelmänvalvoja voi tehdä tietoon perustuvia päätöksiä kapasiteetin suunnittelusta, verkon ylläpidosta, vianmäärityksestä ja verkon turvallisuudesta.

Pelkkä raportointi ei auttaisi järjestelmänvalvojaa ylläpitämään tehokasta verkkoa. Verkonvalvojan täytyy tunnistaa ympäristönsä mahdolliset ongelmakohdat. Raportit auttavat ymmärtämään, mikä on normaalia ja verkon nykyistä tilaa, mutta kynnysarvoihin (engl. thresholds) perustuvat *hälytykset* auttavat verkonvalvojaa tunnistamaan mahdolliset suorituskykyyn ja turvallisuuteen liittyvät verkko-ongelmat ennen kuin ne kaatavat koko verkon. Hälytykset ja raportit täydentävät toisiaan siten, että hälytykset kertovat järjestelmänvalvojalle mahdollisista ongelmista ja raportit tarjoavat tietoa verkko-ongelmien perimmäisen syyn tunnistamiseksi. (All About Network Alerts 2020.)

Hälytykset tarjoavat kattavat tiedot siitä, mikä on mennyt pieleen ja miksi. Ihanteellisessa tilanteessa verkon monitorointityökalu antaa mahdollisimman paljon tietoa havaittuun ongelmaan liittyen, mikä antaa verkonvalvojalle mahdollisuuden ryhtyä välittömiin toimenpiteisiin ongelman ratkaisemiseksi. Älykkäiden hälytysten tulee sisältää tiedot siitä, mikä ongelma on, missä ongelma on, milloin ongelma alkoi ja mihin verkon osa-alueisiin se vaikuttaa. Näiden älykkäiden hälytysten tulee myös olla selkeitä ja ymmärrettäviä, jotta korjaustoimenpiteet olisivat nopeita. Hälytykset voivat jopa sisältää ehdotuksia toimenpiteistä, joita voidaan tehdä ongelman ratkaisemiseksi. (All About Network Alerts 2020.)

Monille yrityksille oikean työkalun löytäminen voi minimoida verkkoilmoitusten määrää heikentämättä verkon suorituskykyä ja turvallisuutta. Verkonvalvontahälytystyökalun tulisi tukea kriittisiä ja porrastettuja hälytyksiä. Ne antavat valvojalle mahdollisuuden priorisoida hälytyksiä, joita pidetään "kriittisinä" tai jotka ylittävät tietyn ennalta määritetyn kynnysarvon. Tehokkaan ohjelmiston tulisi luokitella hälytykset niiden tärkeystason perusteella ja erottaa ei-kriittiset ongelmat kriittisistä ongelmista, jotta priorisointi olisi asianmukaista. Järjestelmät, joissa on porrastettu hälytys, määrittävät ongelmat yhteen useista luokista. Hälytykset lähetetään sen mukaan, kuinka tärkeä

luokka on. Tämä vähentää todennäköisyyttä, että tarpeettomat tai vähemmän tärkeät verkkoilmoitukset vievät tilaa oikeilta ongelmilta. Tyypillisimmät hälytyksien aiheuttajat liittyvät kaistanleveyden käyttöön, siirtonopeuksiin, pakettien tippumiseen, viiveeseen ja saatavuuteen. (All About Network Alerts 2020.)

3 Protokollat

3.1 Yleistä

Maailmassa on monia eri teknisiä laitteita, joilla on samat tarkoitusperät, mutta jotka toimivat hieman eri tavalla. Tarvitaan siis jokin universaali kieli, jotta eri valmistajien laitteet pystyisivät kommunikoimaan keskenään maailmanlaajuisesti. Tätä kieltä voidaan kutsuta standardeiksi. Niiden tarkoitus on mahdollistaa keskustelu verkossa huolimatta siitä, millainen yksittäisen verkon rakenne ja sisäiset prosessit ovat. Näitä standardien sisältämiä sääntöjä voidaan kutsua protokolliksi, jotka siis määrittelevät kuinka data liikkuu laitteiden välillä. Verkkoprotokollilla on siten ratkaiseva rooli nykyaikaisessa digitaalisessa viestinnässä. (Macpherson 2021.)

Verkkoprotokollat ottavat vastaan suuria prosesseja ja jakavat ne pieniin, erityisiin tehtäviin tai toimintoihin, eli paketteihin. Tämä tapahtuu verkon kaikilla tasoilla ja jokaisen toiminnon on tehtävä yhteistyötä kullakin tasolla suorittaakseen käsillä olevan suuremman tehtävän. Topologiasta riippumatta jokainen verkko noudattaa yleisesti tiettyä vertailumallia, jota kutsutaan OSI-malliksi. The Open Systems Interconnections (OSI) on kehitetty jo vuonna 1980-luvulla ja se määrittää perusperiaatteet, joiden mukaan tietoverkot rakennetaan. OSI-malli on suunniteltu varmistamaan eri laitteiden yhteensopivuus, riippumatta siitä, minkä valmistajan rakentamia ne ovat. (Medhi & Ramasamy 2018, 14.)

OSI-malli koostuu seitsemästä eri kerroksesta, joista jokaisella on oma osansa prosessissa ja ne kuvastavat tiettyä funktiota, kun data tai paketit liikkuvat verkossa. Alemmat kerrokset käsittelevät sähköisiä signaaleja, binääritietojen paloja ja näiden tietojen reitittämistä verkkoihin. Korkeammat tasot kattavat verkkopyynnöt ja vastaukset, tietojen esittämisen ja verkkoprotokollat käyttäjän näkökulmasta katsottuna. Mallissa data siirtyy aina yhden kerroksen kerrallaan ja jokainen kerros

käyttää aina alemman kerroksen palveluja ja tarjoaa niitä myös kerrosta ylöspäin. (Medhi & Ramasamy 2018, 15-19).

Kuvio 1 esittää OSI-mallin toimintaperiaatetta. Esimerkiksi kun dataa halutaan lähettää verkkoon, liikkuu se seitsemännestä kerroksesta ensimmäiseen kerrokseen ja loogisesti vastaanottaessa dataa liikkuu se päinvastaisella järjestyksellä.

Layer	Function	Example
Application (7)	Services that are used with end user applications	SMTP,
Presentation (6)	Formats the data so that it can be viewed by the user Encrypt and decrypt	JPG, GIF, HTTPS, SSL, TLS
Session (5)	Establishes/ends connections between two hosts	NetBIOS, PPTP
Transport (4)	Responsible for the transport protocol and error handling	TCP, UDP
Network (3)	Reads the IP address form the packet.	Routers, Layer 3 Switches
Data Link (2)	Reads the MAC address from the data packet	Switches
Physical (1)	Send data on to the physical wire.	Hubs, NICS, Cable

Kuvio 1 OSI-malli (Valtierra 2017)

Verkon valvonnan kannalta OSI-mallin yleisimmin käytetyt kerrokset ovat Siirtokerros (2), verkkokerros (3) ja sovelluskerros (7). Valvontajärjestelmät käyttävät näitä kerroksia löytämään verkon laitteet, kuinka ne ovat liittyneet toisiinsa, luomaan topologiakarttoja ja valvomaan liikennettä. Seuraavissa kappaleissa esitellään valvonnan kannalta yleisimmät protokollat.

3.2 Simple Network Management Protocol

SNMP (Simple Network Management Protocol) on verkonvalvonta sovelluksien yleisimmin käytetty tietoliikenneprotokolla. Protokollan tehtävänä on kysellä verkossa olevien laitteiden tilaa ja raportoida niihin tulevista virheistä. Se on sovelluskerroksen protokolla (OSI-mallin kerros 7), joka tyypillisesti käyttää UDP-siirtoprotokollan porttia 161 ja 162.(Scarpati 2021.)

Vaikka SNMP on protokollana hyvinkin vanha, jo vuonna 1988 ensimmäisen versionsa saaneena, se on upotettu lähes kaikkiin lähiverkon laitteisiin kuten reitittimiin, kytkimiin, palomuureihin ja palvelimiin. Protokollasta on kehitetty kolme versiota, joista kolmas eli uusin versio SNMPv3 on vuodelta 2002. Uusin versio on käytössä nykyään eniten, sillä sen tietoturvaa on kehitetty huomattavasti aiempia versioita pidemmälle. SNMP:tä käytetään paljon lähinnä sen takia, että se on tehty toimimaan todella yksinkertaisesti ja sillä pystytään hallitsemaan isoja määriä eri laitteita. (Jaakohuhta 2005, 260.)

SNMP:n hallintaympäristö määrittyy tyypillisesti seuraaviin komponentteihin:

- **Hallinta-asema (NMS, Network Management Station)**, eli järjestelmä, jota käytetään verkon valvontaan
- **Agentti**, eli moduuli, joka on asennettu hallittavaan laitteeseen, kuten kytkimeen, reitittimeen tai palvelimeen
- **Hallintatietokanta (MIB, Management Information Base),** eli tietokanta, johon SNMP-agentit keräävät ja ylläpitävät tietoja verkkolaitteesta. Kannan rakenne on puumainen ja sen yläosan ominaisuudet ovat standardointijärjestön määrittämät. Alemmat tasot kuuluvat eri organisaatioille ja laitetoimittajille.

Yleensä prosessi lähtee siitä, että NMS haluaa agentilta jotain tietoa laitteesta. Tämän jälkeen agentti etsii vastauksen MIB:stä ja toimittaa sen takaisin NMS:lle. Agentti voidaan myös määrittää niin, että se antaa itsenäisiä tilatietoja ilman NMS:n kyselyitä. (Scarpati, 2021.)

SMNP-managerin ja SNMP-agentin välinen viestintä tapahtuu yleensä User Datagram Protocol (UDP)- tai Transmission Control Protocol/Internet Protocol (TCP/IP) -protokollan kautta, ja niitä kutsutaan protokollatietoyksiköiksi (PDU). Kuvio 2 kuvaa eri viestintätyylejä, joita on nykyään yhteensä seitsemän (Scarpati 2021.):

1. **GetRequest**, komento hakee yhden tai useamman arvon.

2. **GetNextRequest**, noutaa seuraavan objektitunnisteen (OID) arvon MIB-puussa. OID on tunniste, jota käytetään nimeämään ja osoittamaan objekti MIB-hierarkiassa. Jokaisella verkkolaitteella on oma MIB (joka sisältää tietoja, kuten järjestelmän tila-, saatavuus- ja suorituskykytiedot). Jokainen tämän tiedon osa tunnetaan objektina ja tunnistetaan tietyllä OID:lla, alkaen aina 0:sta.

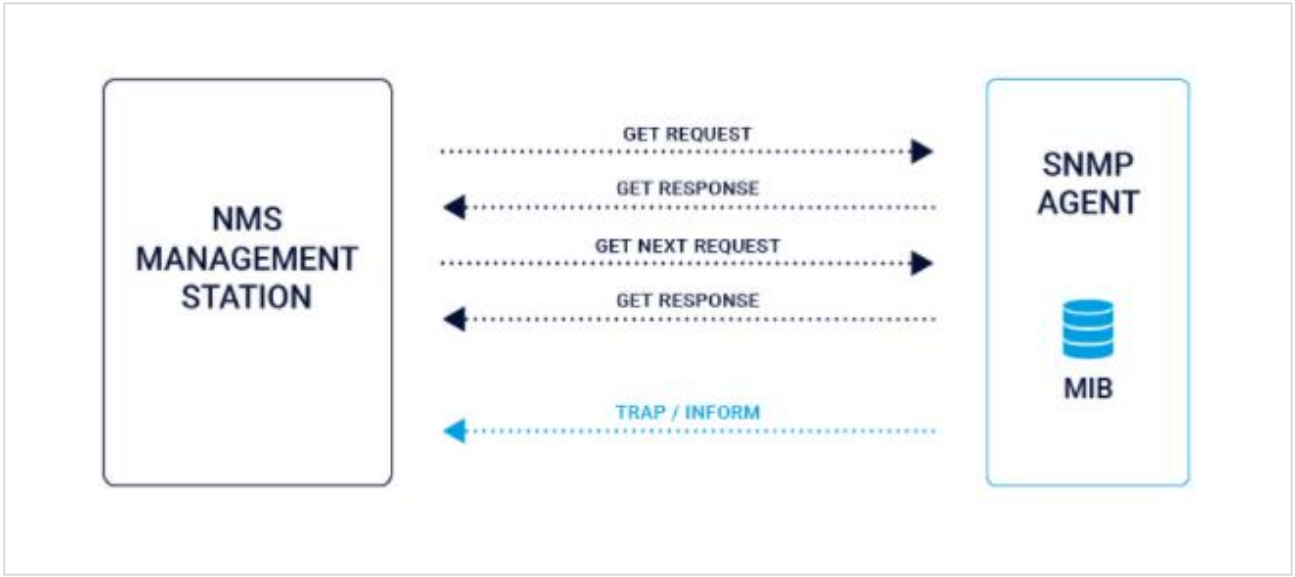
3. **GetBulkRequest**, komento hakee suuria määriä tietoa.

4. **SetRequest**, hallinta-asema voi muokata hallinta-agenttien objektitunnisteiden arvoja.

5. **Traps**, agentti, eli verkkolaite lähettää hallinta-asemalle viestejä ilman, että hallinta-asema ottaa ensin yhteyden.

6. **InformRequest**, ominaisuus, jonka avulla SNMP-agentit voivat lähettää viestejä hallinta-asemalle. Samantyylinen kuin trapit, mutta tässä lähetetään viestejä niin kauan, kunnes agentti saa kuittauksen.

7. **Response**, palauttaa hallinta-aseman pyyntöjen arvoja tai signaaleja.



Kuvio 2 SNMP viestit (Tolliver 2020)

Kuten aikaisemmin mainittiin, SNMPv3 on uusin versio protokollasta ja se otettiin käyttöön vuonna 2002. Siinä missä SNMPv1 ja SNMPv2 kärsivät pahoista tietoturva-aukoista, SNMPv3:n suurimmat erot aiempiin versioihin on sen paranneltu turvallisuus. Kolmanteen versioon lisätty todennus, kulunvalvonta ja salattu data olivat joitakin avainkomponentteja, joita käytettiin parantamaan merkittävästi SNMP:n turvallisuutta. (Emmit 2020.) Viimeisimmässä versiossa on myös eri suojaustasot, joita on kolme. Ensimmäisenä on viestintä ilman mitään todennusta ja yksityisyyttä (noAuthNoPriv). Tämä tarkoittaa, että viesteille ei sovelleta suojausta ja kaikki tekstit näkyvät selkokielellä. Toisena on viestintä todennuksella, mutta ei yksityisyyttä (AuthNoPriv). Tässä viestit todennetaan, eli ne menevät halutulla tunnuksella, mutta niillä ei ole yksityisyyttä eli viestit edelleen selkokielellä luettavissa. Kolmantena ja turvallisimpana menetelmänä on viestintä todennuksen ja tietosuojan kanssa (AuthPriv), jossa kaikki viestit on todennettu ja salattu. (Harrington, Presuhn & Wijnen 2002, 28.)

3.3 Internet Control Message Protocol

ICMP (Internet Control Message Protocol) on virheraportointiin käytettävä protokolla. Sen avulla voidaan määrittää, onko data saavuttanut määränpäänsä kahden laitteen välillä ja missä ajassa se tapahtui. ICMP sijoittuu OSI-mallissa verkkokerrokselle, eli kolmannelle kerrokselle. ICMP on olennainen osa Internet protokollaa, ja jokaisen IP-moduulin on otettava se käyttöön. (Postel 1981, 1-5.)

ICMP-paketin tyyppi jakautuu kahteen osaan, tyyppiin ja koodiin. ICMP paketti sisältää kokonaisviestin, joka viestin on tarkoitus välittää. Esimerkiksi kuviota 3 tarkastellessa, tyypin arvo 3 tarkoittaa, että vastaanottaja ei ole tavoitettavissa. Joillekin tyypeille on useita koodiarvoja, joiden tarkoituksena on antaa lisätietoja. Esimerkiksi tyypin 3 ICMP-sanoma, jonka koodi on 0, viittaa ongelmiin kohdeverkossa, kun taas koodi 1 tarkoittaa, että ongelma on, että tietty isäntä ei ole tavoitettavissa. (Postel 1981, 1-5.) Kuviossa 3 löytyy listattuna enemmän paketin eri tyyppejä sekä koodeja.

Type	Code	Description
0 – Echo Reply	0	Echo reply
3 – Destination Unreachable	0	Destination network unreachable
	1	Destination host unreachable
	2	Destination protocol unreachable
	3	Destination port unreachable
	4	Fragmentation needed and DF flag set
	5	Source route failed
5 – Redirect Message	0	Redirect datagram for the Network
	1	Redirect datagram for the host
	2	Redirect datagram for the Type of Service and Network
	3	Redirect datagram for the Service and Host
8 – Echo Request	0	Echo request
9 – Router Advertisement	0	Use to discover the addresses of operational routers
10 – Router Solicitation	0	
11 – Time Exceeded	0	Time to live exceeded in transit
	1	Fragment reassembly time exceeded
12 – Parameter Problem	0	Pointer indicates error
	1	Missing required option
	2	Bad length
13 – Timestamp	0	Used for time synchronization
14 – Timestamp Reply	0	Reply to Timestamp message

Kuvio 3 ICMP-paketin arvoja (Belding 2020)

ICMP-protokollaan liittyy vahvasti komennot ”ping” ja ”traceroute”. Traceroutea käytetään kahden laitteen välisen reitityspolun selvittämiseen. Reitityspolku on kytkettyjen reitittimien fyysinen polku. Reitittimen ja toisen välinen matka tunnetaan "hyppynä" (engl.hop), ja traceroute ilmoittaa myös kunkin hypyn vaatiman ajan. Tämä voi olla hyödyllistä määritettäessä verkon viiveen lähteitä. Pingiä käytetään määrittelemään kahden laitteen välisen yhteyden nopeutta ja raportoimaan tarkalleen, kuinka kauan datapaketin saapuminen määränpäähänsä ja takaisin lähettäjän laitteelle kestää. (CloudFlare 2021.)

3.4 Syslog

Syslog-protokollan avulla voidaan lähettää lokiin kirjautuneita tapahtumia eteenpäin, esimerkiksi hallintajärjestelmään tai erilliselle palvelimelle. Syslog on määritelty IETF:n RFC:ssä 5424. Syslog-sanoma koostuu kolmesta osasta: PRI (laskettu prioriteettiarvo), HEADER (tunnistetiedot) ja MSG (itse viesti). Syslog-protokollan kautta lähetettävät PRI-tiedot tulevat kahdesta numeerisesta

arvosta, jotka auttavat luokittelemaan viestin. Ensimmäinen on toimitila- eli facility-arvo. Tämä arvo on yksi 15 ennalta määritetystä arvosta tai useista paikallisesti määritellyistä arvoista, jos kyseessä on arvo väliltä 16-23. Nämä arvot luokittelevat viestin tyypin. Syslog-viestin toinen otsikko luokittelee viestin tärkeyden tai vakavuuden asteikolla 0-7. Näin lasketaan viestin prioriteetti ja mitä pienempi PRI-arvo, sitä korkeampi prioriteetti sillä on. (Gerhard 2009, 9-11.)

Syslog-protokolla voi tuottaa paljon viestejä. Syslog yksinkertaisesti välittää viestit niin nopeasti kuin se luo ne. Tämän seurauksena syslog-palvelimen tärkein ominaisuus on kyky suodattaa oikealla tavalla saapuvat lokitiedot ja reagoida niihin.

4 Nagios XI

Nagios on 2000-luvun taitteessa kehitetty avoimen lähdekoodin verkon valvonta- ja monitorointisovellus. Alun perin ohjelmisto kulki ”Netsaint” nimellä, mutta vuonna 2002 nimi muutettiin Nagiokseen Netsaintin tavaramerkkiongelmien vuoksi. Nagios on valittu useaan otteeseen parhaaksi monitorointisovellukseksi suositun linuxquestions -sivuston äänestyksessä. (History of Nagios n.d.)

Se perustuu suurimmilta osin virallisiin ja käyttäjien tekemiin lisäosiin ja plugineihin, joita voi kuka tahansa ladata ja ne ovat koottuna heidän verkkosivuilleen. Nagios tarjoaa neljää eri tuotettaan kertamaksu periaatteella, lisenssin ostamisen jälkeen ei vaadita enää kuukausimaksuja tai muita käyttömaksuja. (Serie 2021.)

Nagios XI luotiin Nagios Coren rinnalle tuomaan monitorointipalveluun lisää visuaalisuutta ja helppokäyttöisyyttä. Siinä missä Core vaatii enemmän teknistä osaamista luettavuudessa, Nagios XI tehtiin enemmän yrityskäyttöön tuomaan monipuolisuutta laajemmilla ominaisuuksilla. Kaavioiden ja raporttien, muokattavien taulukkojen, integroidun tietokannan, backend API:n, multi-tenancyn (usean käyttäjän samanaikainen käyttö) ja ohjattujen toimintojen ansiosta Nagios XI ylittää huomattavasti kaiken, mitä Core voi tehdä käytettävyyden ja nopeuden suhteen. (Serie 2021.)

4.1 Arkkitehtuuri

Nagioksen toimintaperiaate on seuraavanlainen. Se toimii isäntäpalvelimella yleensä palveluprosessina (engl. Daemon) ja kommunikoi sieltä hostien kautta kahdella eri tavalla riippuen kelle kysely on osoitettu. Nämä kaksi tapaa ovat agenttipohjainen valvonta ja protokollapohjainen valvonta. (Velimirovic 2021.)

Agenttipohjaisessa valvonnassa Nagios XI esimerkiksi kysyy joltain hostilta sen CPU:n tilatietoja. Agentti palauttaa informaation takaisin isäntäpalvelimelle, jossa Nagios muuttaa tiedot visuaalisiksi graafiseen käyttöliittymään ja voi luoda tiedoista hälytyksen, jos ne poikkeavat ennalta määritetyistä arvoista. Protokollapohjaisen tavan valvonnassa käytetään esimerkiksi SNMP:tä. Siinä Nagios voi kysyä kytkimeltä onko jokin sen porteista ylhäällä vai ei. Se lähettää kyselyn, johon kytkin vastaa palauttaen sen takaisin Nagiokselle. (Mackin 2017.)

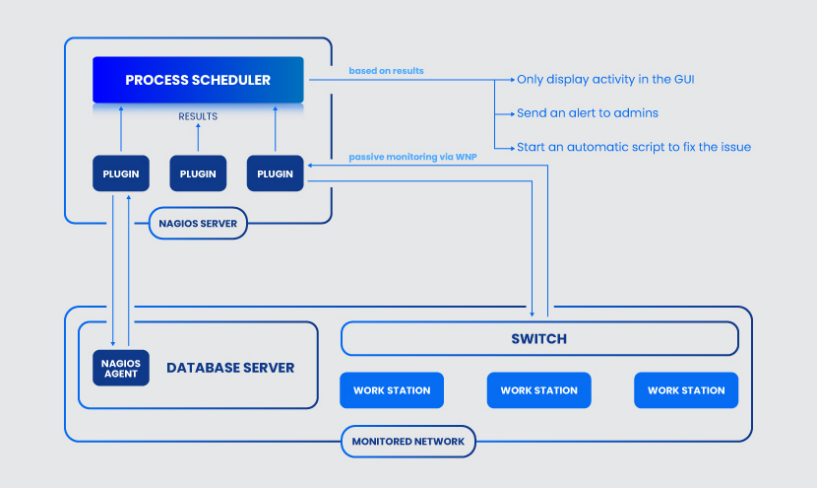
Arkkitehtuuri koostuu kolmesta eri osatekijästä (ks. kuvio 5). Ensimmäisenä on prosessiajastin (eng. process scheduler), joka toimii nimensä mukaisesti tehtävien aikatauluttajana. Sen tehtävänä on käynnistää prosesseja määritellyin väliajoin ja suorittaa toimintoja sille tulleiden tulosten perusteella, kuten esimerkiksi hälytysten lähetys. (Velimirovic 2021.)

Toisena ovat liitännäiset (eng. plugins). Ne ovat binääritiedostoja tai komentosarjoja, joiden rooli on keskustelu Nagioksen ja hostien välillä. Ne sijaitsevat Nagioksen palvelimella ja joita scheduler ohjaa tekemään haluttuja toimintoja. Pluginit jaetaan kolmeen eri kategoriaan, viralliset lisäosat, yhteisön lisäosat ja mukautetut lisäosat. Nagios ylläpitää n. 50 virallista lisäosaa, mutta yhteisön tekemiä lisäosia on yli 3000, joita jaetaan https://exchange.nagios.org/ -sivustolla. Kuitenkin riippumatta siitä, onko lisäosa virallinen vai ei, noudattaa se tiettyä status koodia, jotka näkyvät kuviossa 4. (Velimirovic 2021.)

EXIT CODE	STATUS	DESCRIPTION
0	OK	The system is working fine
1	WARNING	The system continues to operate but requires attention
2	CRITICAL	The system is not working correctly
3	UNKNOWN	The plugin cannot assess the status of the host or service

Kuvio 4 Lisäosien statuskoodit (Velimirovic 2021)

Kolmas osatekijä on graafinen käyttöliittymä (GUI). Se on muokattava käyttöliittymä, joka tarjoaa käyttäjälle yleiskatsauksen kaikista hosteista, palveluista ja verkkolaitteista. Aloitusnäkymä on ensimmäinen sivu mikä avautuu, kun kirjaudutaan sisään käyttöliittymään. Siihen voidaan laittaa yhteenveto tärkeimmistä monitoroitavista palveluista, jotta saadaan heti kuva verkon tilasta. Aloitusvalikko jakaantuu pienempiin alavalikoihin. Ensimmäisenä on yksityiskohtaisempi valikko, jossa näytetään kaikki monitoroitavat asiat jokaiselta hostilta ja niitä voidaan tarkastella tarkemmin. Grafiikkanäkymässä voidaan esittää eri kaavioita, jotka piirtävät reaaliaikaista tietoa verkosta. Karttaosiossa voidaan piirtää prosessi, jossa visualisoidaan kaikki verkossa olevat laitteet, kuinka ne on yhdistetty ja kuinka koko verkko on rakennettu. Verkkokartta antaa yleensä tietoja siitä, toimiiko verkko oikein tai onko jossain tietyssä laitteessa ongelmia. Tapahtumanhallinta eli hälytyslista sisältää yhteenvedon kaikista ilmoituksista ja hälytyksistä ja niiden kriittisyydestä. (Velimirovic, 2021.) Aloitusnäkymän jälkeen on views -näkymä, joka tarjoaa nopean pääsyn ulkoisten sivustojen tietoihin käyttöliittymän kautta. Seuraavana on Dashboard -valikko, johon voidaan kerätä tärkeimmät tiedot palveluista yhteen paikkaan. Reports välilehdellä voidaan luoda koostettuja tilastoja ja graafisia raportteja isännistä ja palveluista tietyn ajanjakson aikana. Konfigurointi valikossa voidaan määrittää uusia valvontaprosesseja, joista lisää myöhemmin. Admin valikossa voidaan konfiguroida käyttäjätunnuksiin ja käyttöliittymään liittyviä asetuksia. Viimeiset valikot Tools ja Help tarjoavat ohjeita Nagioksen käytössä. (Nagios XI User Interface n.d.)



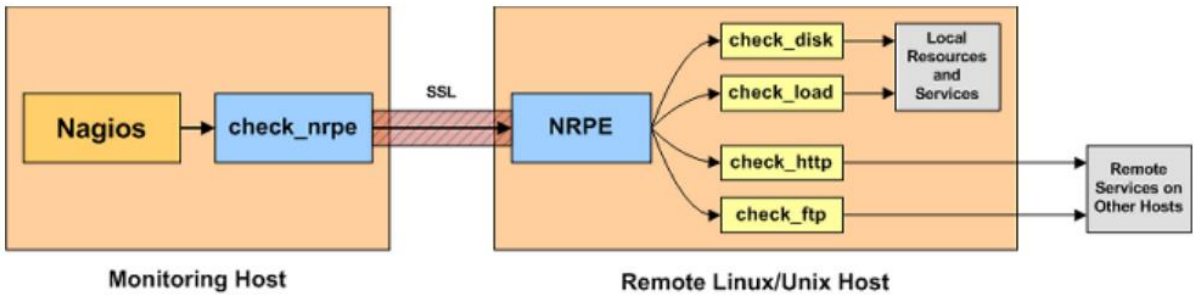
Kuvio 5 Nagioksen arkkitehtuuri (Velimirovic 2021)

4.2 Agenttipohjainen valvonta

Valvonnassa Nagios hyödyntää eniten avoimen lähdekoodin agentteja, jotka tarjoavat käyttäjille joustavamman ja ymmärrettävän tavan monitoroida. Seuraavaksi esitellään kahden agentin toimintatapaa, jotka ovat tärkeimpiä ja myös työn kannalta oleellisimpia agentteja. Nämä ovat Nagios Cross-Platform Agent (NCPA) ja Nagios Remote Plugin Executor (NRPE).

NCPA koostuu kahdesta osasta, jotka ovat NCPA Listener ja NCPA Passive. Yhdessä ne muodostavat yhden valvonta-agentin. NCPA Listener tehtävänä on ylläpitää yhteyksiä verkkokäyttöliittymään, käsitellä API-pyyntöjä (mukaan lukien aktiivisten tarkistusten pyynnöt) sekä tarjota graafiseen käyttöliittymään reaaliaikaista grafiikkaa. NCPA Passive taas huolehtii passiivisten tarkastusten suorittamisesta sekä toimittaa tulokset valvottavasta kohteesta itsenäisesti. Tämä kahden palvelun eriyttäminen mahdollistaa sen, että agentti voi suorittaa passiivisia tarkistuksia ilman, että sen tarvitsee sallia yhteyksiä verkkokäyttöliittymään tai ulkoisia kutsuja API:lle. Useimmiten valvonnassa käytetään aktiivisia tarkistuksia kuin passiivisia. Suurimpana erona näiden kahden välillä on se, että aktiivisessa tarkistuksessa tarkastuksien käynnistämisen ja suorittamisen hoitaa Nagios, kun taas passiivisessa se on toisinpäin. Passiivista tarkistusta käytetään yleensä hajautetussa valvonnassa, jossa pyritään vähentämään Nagios-palvelimen liikaa kuormittamista. (NCPA n.d.)

NRPE-lisäosa on suunniteltu mahdollistamaan Nagios-laajennusten suorittaminen Linux/Unix-käyttöjärjestelmissä. NRPE koostuu sekin kahdesta osasta, check_nrpe plugin ja NRPE daemon. Check_nrpe plugin sijaitsee Nagioksen palvelimella ja NRPE daemon on asennettuna monitoroitavassa laitteessa. Kun Nagios haluaa saada tietoa Linux/Unix-etäkoneesta, se antaa käskyn check_nrpe-laajennuksen ja kertoo sille, mikä palvelu on tarkistettava. Check_nrpe-laajennus ottaa yhteyttä etäisännän NRPE-daemoniin (valinnaisesti) suojatun yhteyden SSL:n kautta. NRPE-daemon suorittaa käskyn tarkistaakseen palvelun tai resurssin ja välittää tulokset NRPE-daemonista takaisin check_nrpe-laajennukseen, joka sitten palauttaa tarkistustulokset Nagiokselle (ks. kuvio 6). (Galstad 2017.)



Kuvio 6 NRPE:n toiminta (Galstad 2017)

4.3 Nagios Core Config Manager (CCM)

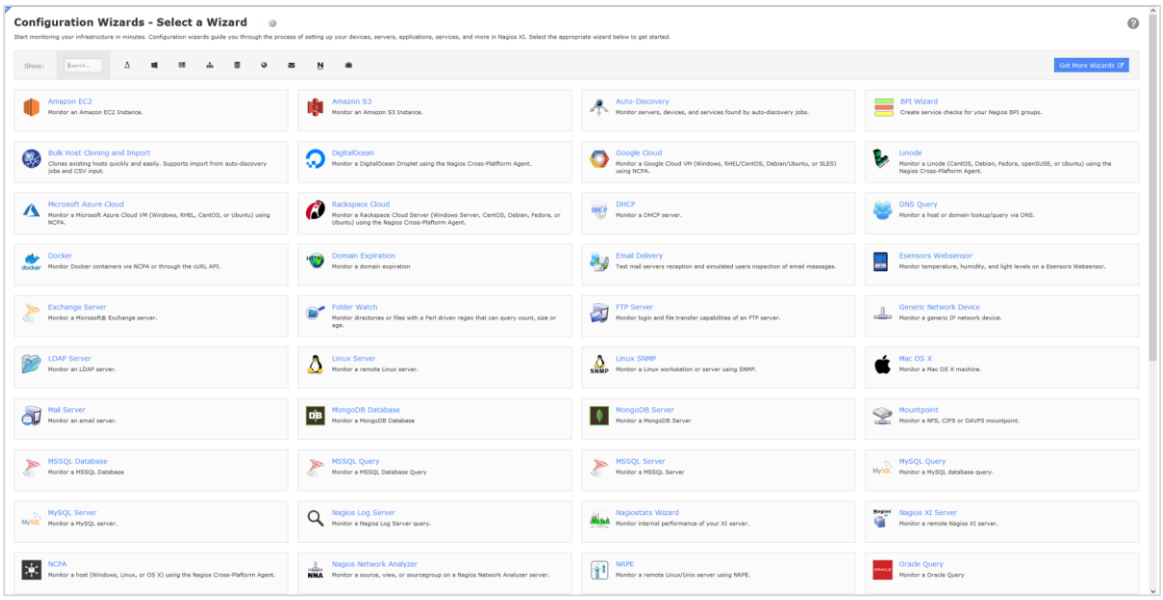
Nagios XI on rakennettu aikaisemmin kehitetyn Nagios Coren päälle selkeyttämään ja tekemään monitoroinnista helppokäyttöisempää. Se ohittaa käyttäjien tarpeen ymmärtää komentorivikoodia käyttäjäystävällisellä käyttöliittymällä, joka on suunniteltu ei-teknisille käyttäjille. Tämä käyttöliittymä helpottaa huomattavasti uusien työntekijöiden kouluttamista Nagios XI:ssä ja antaa järjestelmänvalvojille mahdollisuuden siirtää ohjat vähemmän teknisille käyttäjille päivittäisten tehtävien hoitamisessa. Kun muut työntekijät hoitavat päivittäisiä tehtäviä, järjestelmänvalvojalla on enemmän aikaa keskittyä monimutkaisiin projekteihin, jotka vaativat hänen teknistä asiantuntemustansa. (Using The Core Config Manager For Host Management 2018.)

Vaikka Nagios Core on tehokas moottori, sen konfigurointi on aikaa vievää ja monimutkaista. Varsinkin suuremmassa verkossa se syö huomattavasti tehokkuutta, kun jokainen valvottava palvelu

on yksittäinen tekstitiedosto ja useiden palveluiden muokkaaminen kerralla on sekavaa. Tästä syystä Nagios XI:lle tehtiin ominaisuus, joka toimii viestin välittäjänä Coren ja XI:n välillä. Tämä ominaisuus on Core Config Manager (CCM). CCM:llä pystytään tekemään konfiguraatiot graafista käyttöliittymää käyttäen yhdestä paikasta, mikä tekee siitä huomattavasti helppokäyttöisemmän kuin mitä Core on. Kun CCM:llä tehdään muutos, esimerkiksi lisätään uusi laite monitorointiin, tallentaa se muutokset ensin tietokantaan, mutta ei vielä järjestelmän konfigurointitiedostoihin. Vasta kun määritykset halutaan ottaa käyttöön, CCM siirtää muutokset tietokannasta Corelle, jossa ne kirjataan ja otetaan käyttöön. CCM tallentaa tietokantaan myös aiempien konfiguraatioiden palautuspisteitä, eli niin kutsuttuja checkpointteja. Jos jokin uusista määrityksistä on virheellisesti tehty, ei monitorointi keskeydy, vaan CCM ottaa käyttöön aikaisemman toimivan checkpointin. (Using The Core Config Manager For Host Management 2018.)

4.4 Configuration Wizards

Wizardit ovat lisäosia, joiden avulla laitteiden ja palveluiden lisääminen on tehty helpommaksi, kun käyttäjän ei tarvitse ymmärtää niin tarkasti, miten Nagios toimii taustalla (ks. kuvio 7). Ohjatut konfigurointitoiminnot tarjoavat käyttäjille helpot, vaiheittaiset ohjeet uusien hostien ja palveluiden seuraamiseen. Wizardin kautta lisäämisen etuna on se, että se on nopeampaa ja automatisoidumpaa, mutta ne eivät sisällä yhtä tarkasti säädettäviä parametreja, kuin mitä Core Config Manager tarjoaa. (Understanding And Using Configuration Wizards In Nagios XI 2017.)



Kuvio 7 Erilaisia Configuration Wizardeja

5 Toteutus

5.1 Kohdeverkko

Kohdeverkko koostui Laajaverkosta (WAN) sekä sen päälle VPN-tunneleilla tehdyistä LAN-verkoista. Näiden välillä ei ole pääsyä toisiinsa. Nagios palvelin asennettiin LAN-verkkoon, jossa myös valvonta ja monitorointi tapahtui. Itse verkko sisälsi useita solmuja, kuten reitittimiä, kytkimiä, palvelimia ja palomuureja.

5.2 Alkuasennukset

Kuten koko tuotetta mainostetaan helppokäyttöisenä, ei ohjelmiston asennus tee siinä poikkeusta. Järjestelmävaatimukset eivät ole kovinkaan suuret, 2+ GHz prosessori, 1GB RAM:ia sekä 40 GB muistia. Työssä Nagios asennettiin offline-ympäristöön VMwaren virtuaalikoneelle käyttöjärjestelmänään Centos7. Virallisilla sivuilla suositellaan vahvasti järjestelmän asentamista täysin uuteen kerneliin virheiden välttämiseksi. Nagioksen asentaminen onnistuu suoraan hakemalla virallisilta sivuilta halutun version ja ajamalla sen omalla palvelimellaan.

cd /tmp/rpms

tar xzf nagiosxi-haluttu-versio.tar.gz

cd nagiosxi

./fullinstall

5.2.1 Hostname ja verkkoasetukset

Ennen graafisen käyttöliittymän asetuksien määrittämistä muokattiin ensin vielä virtuaalikoneen asetuksia. Ensimmäisenä vaihdettiin hostname oletusnimestä halutuksi ja muutettiin verkkoasetukset kohdeverkkoon sopiviksi. Verkkoasetukset löytyvät työn liitteestä 1. Kernelille tehtiin muutamia kovennuksia, kuten esimerkiksi ylimääräisten tunnusten tekemisen estämistä, ipv6 käytöstä poistamista, bannerin lisäämistä kirjautumiseen ja pakettien uudelleenohjaamisen estämistä oletuksena. Kovennuksien tarkoituksena on parantaa palvelimen turvallisuutta ja näin vähennetään haavoittuvuuksia.

5.2.2 NTP

Palvelimelle tulee määritellä aikapalvelu, jotta sen aika on synkronissa muiden verkossa olevien laitteiden kanssa. Protokollaa toteuttaa chrony ja Network Time Protocol (NTP) ja tässä työssä päädyttiin käyttämään jälkimmäistä ohjelmistoa. Tämän seurauksena käytiin ensin sammuttamassa chrony palvelu.

 # systemctl stop chronyd
 # systemctl disable chronyd

Sen jälkeen käytiin konfiguroimassa ntpd asetuksia lisäämällä sinne seuraavat parametrit:

 # vi /etc/ntp.conf
server xxx.xxx.xxx.xxx prefer iburst burst
server 127.127.1.0 iburst burst #local clock
fudge 127.127.1.0 stratum 10

Iburst-tila on konfiguroitavissa oleva vaihtoehto, ei oletustoiminto. Jos NTP-palvelin ei vastaa, iburst-tila jatkaa toistuvien kyselyjen lähettämistä, kunnes palvelin vastaa ja ajan synkronointi

alkaa. Tämän jälkeen ntpd-palvelu käynnistettiin vielä uudelleen:

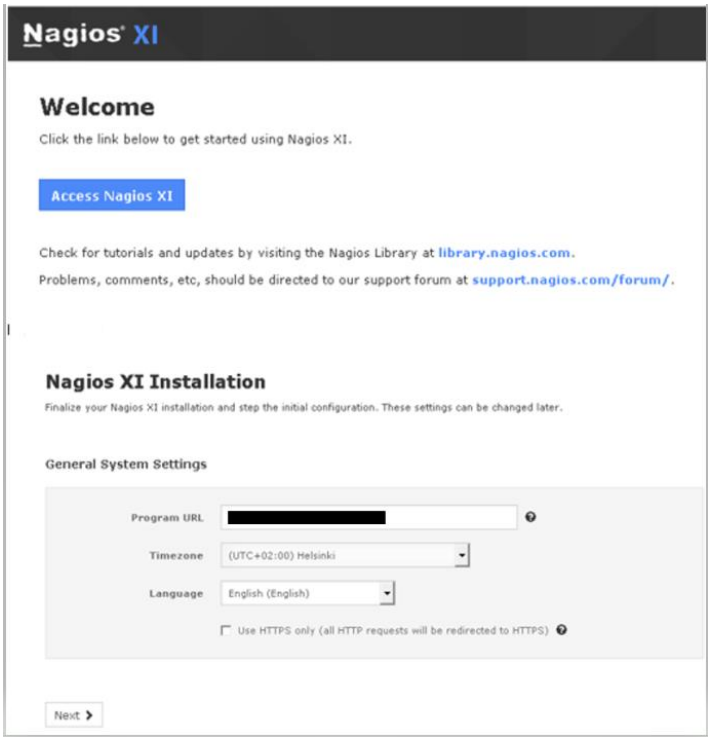
 # systemctl restert ntpd.service

NTP-asetuksien konfiguroinnin jälkeen vaihdettiin vielä aikavyöhyke oikeaksi ja tarkistettiin, että aika vastasi todellisuutta:

 # sudo timedatectl set-timezone Europe/Helsinki
 # timedatectl

5.3 Graafinen käyttöliittymä (GUI) ja lisensointi

Virtuaalipalvelimen asetusten jälkeen avattiin itse graafinen käyttöliittymä palvelimen IP-osoitteella, jossa asetetaan yleisiä asetuksia kuten kielimääritykset URL-osoite, aikavyöhyke, sekä oletus ylläpitäjän käyttäjänimi ja salasana (ks. kuvio 8). Tuotteen lisensoinnissa Nagios tarjoaa 30 päivän kokeilujaksoa, ilmaista palvelua rajoittaen valvottavien hostien määrän seitsemään tai itse lisensointia eri hinnoittelulla lähtien 100 hostin lisenssistä aina rajoittamattomaan määrään. Toimeksiantajalla oli valmiiksi tarjota Nagiokseen lisenssi, joten se lisensoitiin tässä vaiheessa.



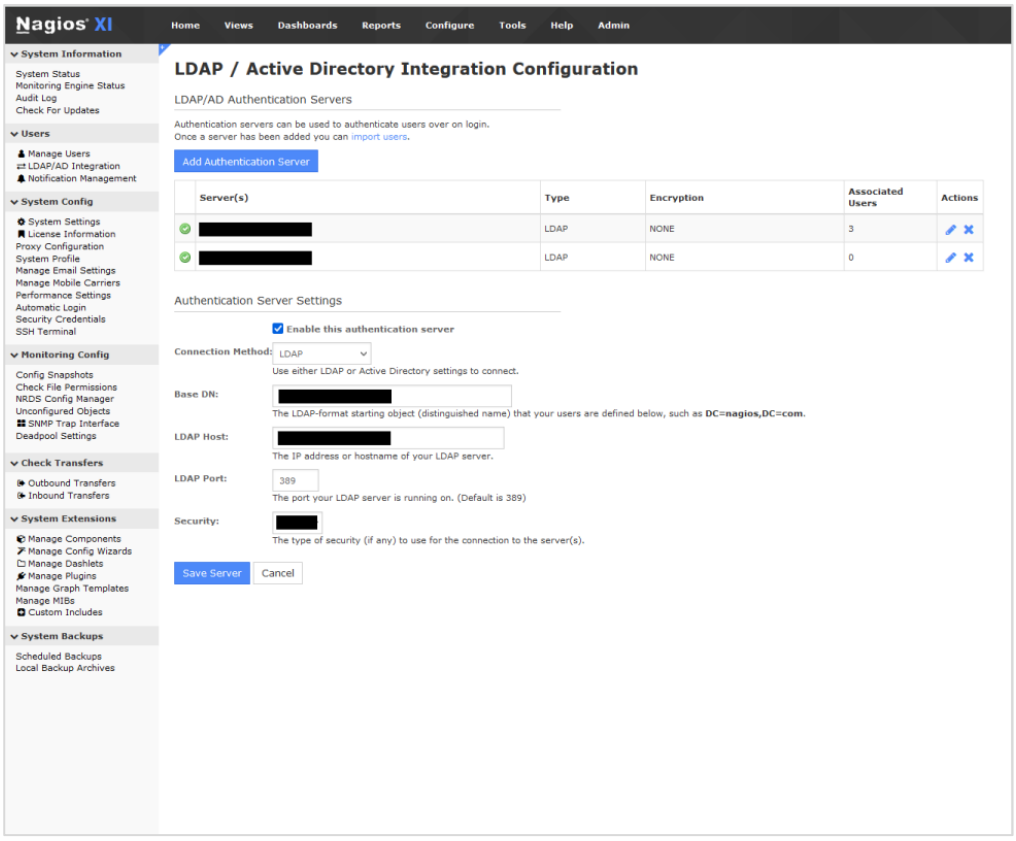
Kuvio 8 Graafisen käyttöliittymän alkuasennus

5.4 LDAP

Pelkän admin tunnuksen käyttäminen ohjelmistossa, jota käyttäisi useampi kuin yksi henkilö, olisi huono ratkaisu selkeyden ja tietoturvan kannalta. Tämän vuoksi Nagiokseen liitettiin osaksi keskitettyä käyttäjän hallintaa LDAP-protokollalla. Lightweight Directory Access Protocol (LDAP) määrittää käyttäjien todennusta ja valtuutusta palvelimiin verkkolaitteisiin ja tiedostoihin. Tämä vähentää useiden käyttäjien luomista eri ohjelmistoihin, kun yhdellä käyttäjätunnuksella henkilö pääsee kaikkiin paikkoihin, johon hänellä on oikeudet. (What Is LDAP & How Does It Work? n.d.)

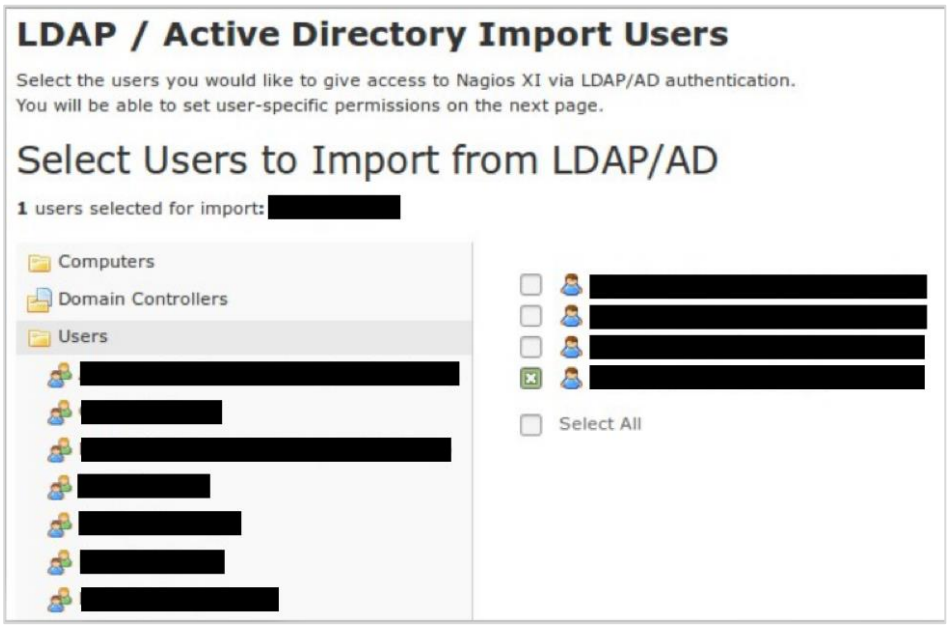
Nagioksessa LDAP konfiguraatio on tehty yksinkertaiseksi (ks. kuvio 9). Asetuksista löytyy ”Authentication Server Settings”, josta valitaan yhteystavaksi LDAP. Base DN kohtaan merkitään paikka, jota LDAP-palvelin käyttää etsiessään käyttäjien todennusta hakemistosta. LDAP-palvelin, jota vastaan Nagios XI voi todentaa, merkitään kohtaan LDAP Host. Tämä voi olla lyhyt IP-osoite nimi tai täydellinen verkkotunnus. LDAP kommunikoi oletuksena portista 389. Viimeisenä kysytään, mitä

menetelmää käytetään yhteyden salaamiseksi. Suositeltavaa on käyttää salattua tiedonsiirtoja LDAP-palvelimen ja Nagioksen välillä.



Kuvio 9 LDAP konfiguraatio

LDAP:n lisäämisen jälkeen pystyttiin nyt lisäämään käyttäjiä käyttäen keskitettyä palvelua. Lisääminen tapahtui ”Manage Users” välilehdeltä valitsemalla ”Add users from LDAP/AD” (ks. kuvio 10). Listasta voidaan sen jälkeen valita halutut käyttäjät, jotka halutaan integroida Nagiokseen. Jokaiselle käyttäjälle on vielä määriteltävä omat asetukset, jotka liittyvät kieli- ja aika-asetuksiin, sekä käyttäjien oikeuksiin.



Kuvio 10 LDAP käyttäjien lisääminen

5.5 Laitteiden lisääminen

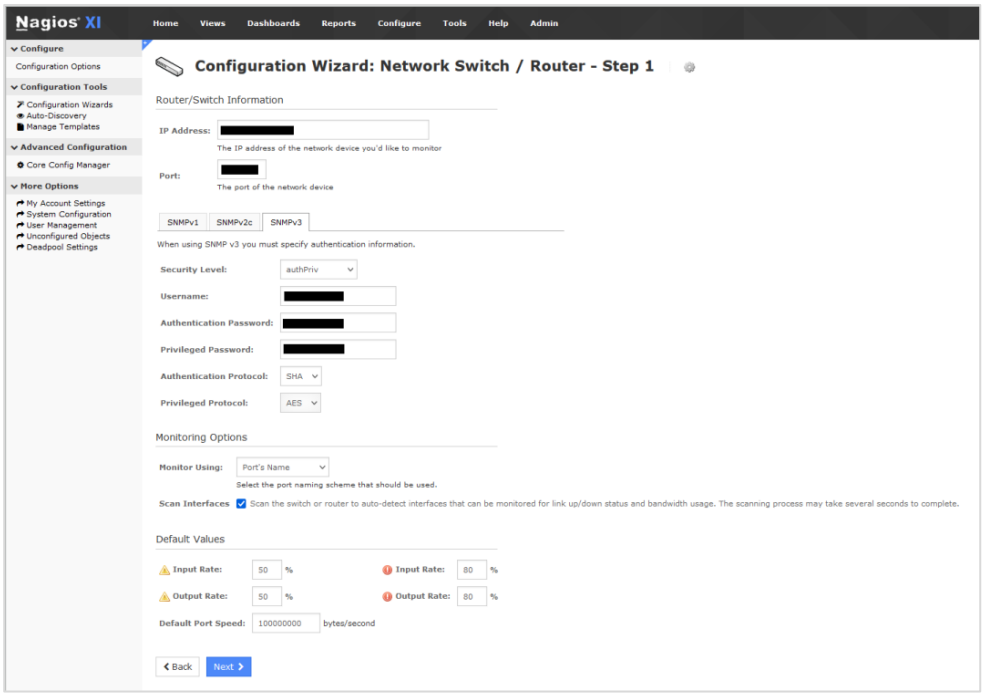
Kuten aiemmin mainittiin, laitteita ja palveluita voidaan lisätä Nagios XI Core Config Managerin (CCM) tai Configuration wizardin kautta. Toimeksiannossa kokeiltiin molempia tapoja, mutta hyvin pian huomattiin, että on nopeampaa alkuun lisätä palvelu käyttäen wizardia ja sen jälkeen muokata sitä tarpeen vaatiessa uudestaan CCM:n kautta.

5.5.1 Reitittimet ja kytkimet

Valvontaan haluttujen reitittimien ja kytkimien lisääminen toteutettiin Network Switch / Router wizardilla. Lisäosalla pystyi monitoroimaan laitteen saatavuutta, porttien tilaa, sekä niiden kaistaleveyttä. Laitteiston suuren määrän takia ei olisi järkevää monitoroida aivan jokaisen portin liikennettä, koska kaikki portit eivät joka laitteessa ole välttämättä edes käytössä ja näin aiheuttaisi turhia hälytyksiä ja kuormitusta.

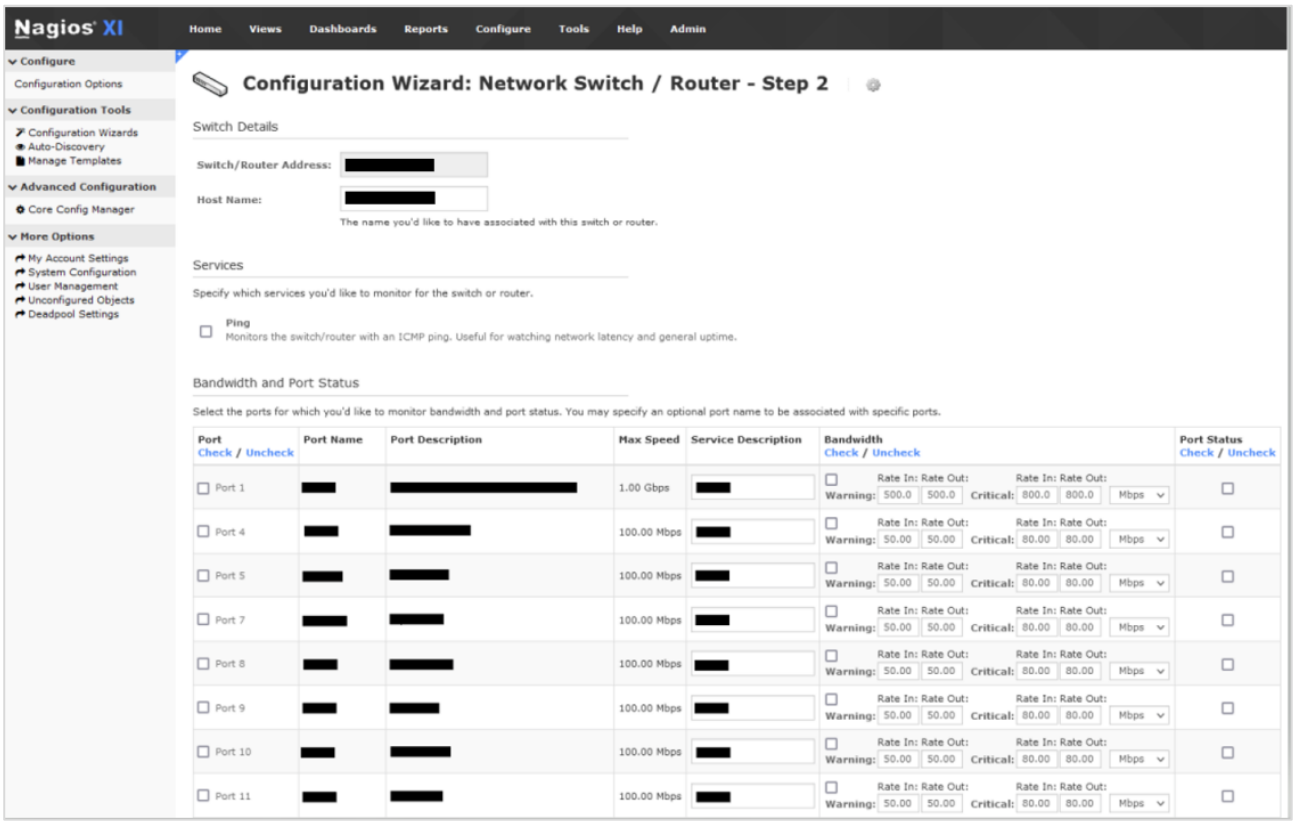
Laitteen lisääminen wizardilla on viisivaiheinen ja siihen tarvitaan laitteen hallinta IP-osoitetta, eli osoitetta johon laite vastaa, sekä siinä tulee olla valmiiksi konfiguroituna SNMP asetukset. Työssä

käytettiin SNMPv3:sta, joka oli sen versioista tietoturvallisin. Wizard hakee SNMP:n avulla listan laitteen kaikkien porttien tiedoista ja antaa ne Nagiokselle (ks. kuvio 11).



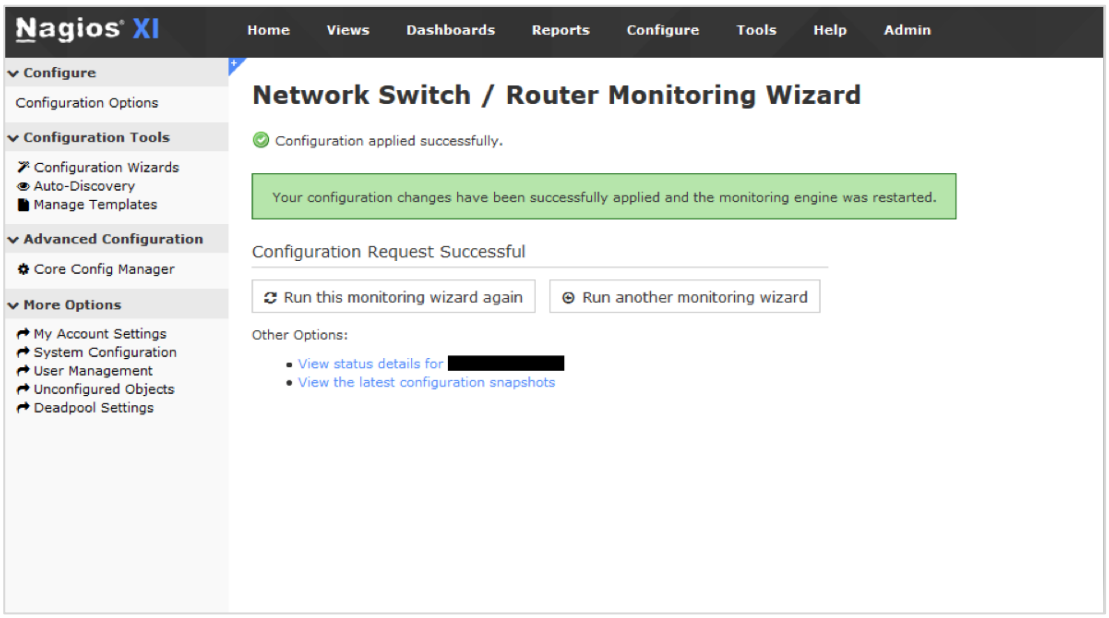
Kuvio 11 Wizardin IP- ja SNMP asetuksia

Seuraavassa vaiheessa nimetään laite, sekä valitaan ICMP pingien lähetys laitteeseen, jotta voidaan valvoa sen saatavuutta (ks. kuvio 12). Samalla valitaan mitä portteja halutaan monitoroida ja työssä päädyttiin porttien osalta monitoroimaan laitteen oleellisimpien porttien statusta sekä niiden kaistanleveyttä.



Kuvio 12 Ote wizardin vaiheesta 2

Kolmessa viimeisessä vaiheessa konfiguroidaan sen hälytysasetuksia, eli kuinka usein laitteelta kysytään sen tietoja, kuinka nopeasti lähetetään virheilmoitus ja kelle kaikille järjestelmänvalvojille ilmoitus lähetetään. Näillä asetuksilla voidaan minimoida vääriä ja aiheettomia hälytyksiä, jotka ovat nopeasti ohimeneviä. Tässä on kuitenkin oltava tarkkana ja osattava määritellä, mitkä virheet ovat harmittomia ja mitkä eivät, jotta valvonta pysyy luotettavana. Virheilmoitusten kohdistaminen tehostaa myös valvontaa, koska sillä voidaan isossa verkossa jakaa hyvin eri vastuualueita järjestelmänvalvojien välillä. Onnistuneen lisäämisen jälkeen Nagios ilmoittaa kuvion 13 tavalla, että se on ottanut muutokset käyttöön ja aloittaa kohteen monitoroinnin.

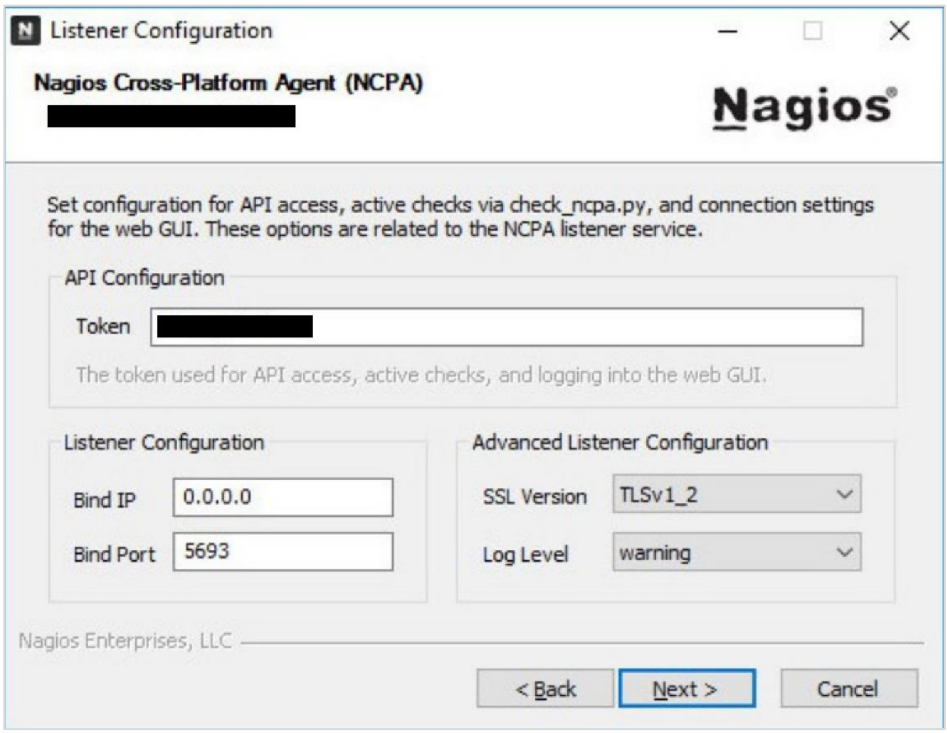


Kuvio 13 Laite lisätty onnistuneesti

5.5.2 NCPA

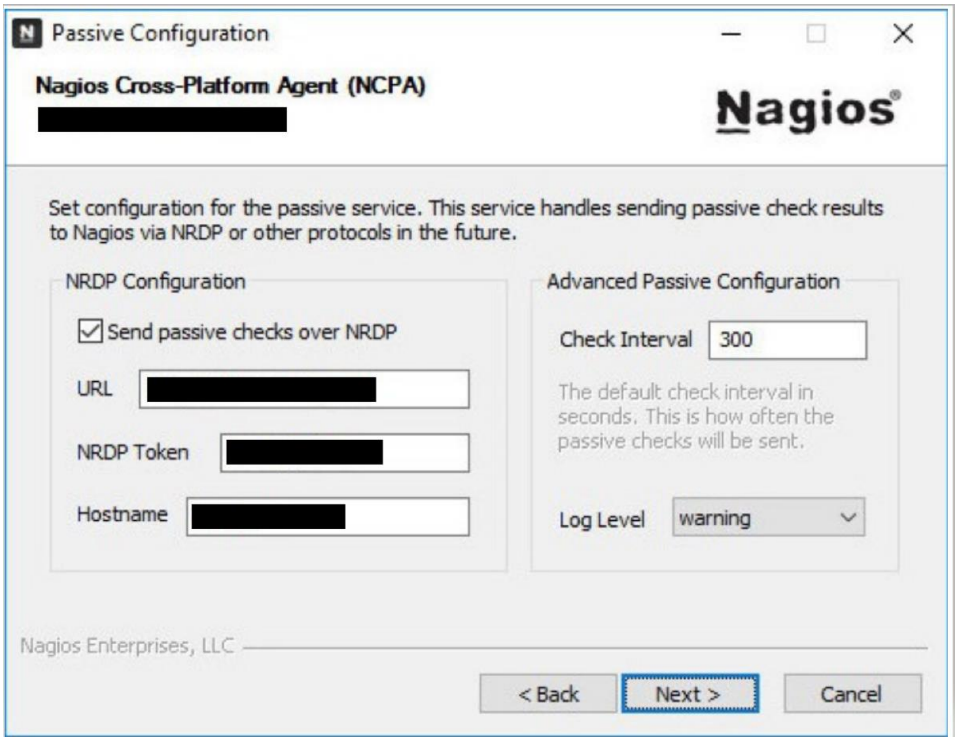
Windows pohjaisten palvelimien monitoroinnissa käytettiin NCPA agenttia. Ennen kuin Nagioksella pystyi käyttämään asennus wizardia, täytyi NCPA käydä asentamassa valvottavaan kohteeseen.

NCPA tarjoaa valmiita asennuspaketteja eri alustoille, ja tässä työssä käytettiin Windowsille suunnattua EXE-pakettia. Asennus sisälsi kolme eri vaihetta, jotka olivat yksinkertaisia ja ohjattuja, joten asennus sujui helposti vailla ongelmia. Lisenssisopimuksen hyväksymisen jälkeen tuli kuvion 14 mukainen vaihe. Siinä täytyi määritellä NCPA:lle Token, joka on käytännössä salasana, jota Nagios käyttää todentaakseen NCPA:n. Bind IP:llä voidaan valikoivasti valita, mitä IP-osoitteita palvelinprosessi kuuntelee. Jättäessä osoitteen muotoon 0.0.0.0, kuuntelee se kaikkia saatavilla olevia. SSL Version ja Log Level kohdissa määritellään mitä suojausprotokollaa käytetään ja mitä lokitasoa käytetään. Nämä jätetiin vastaamaan oletusarvoja.



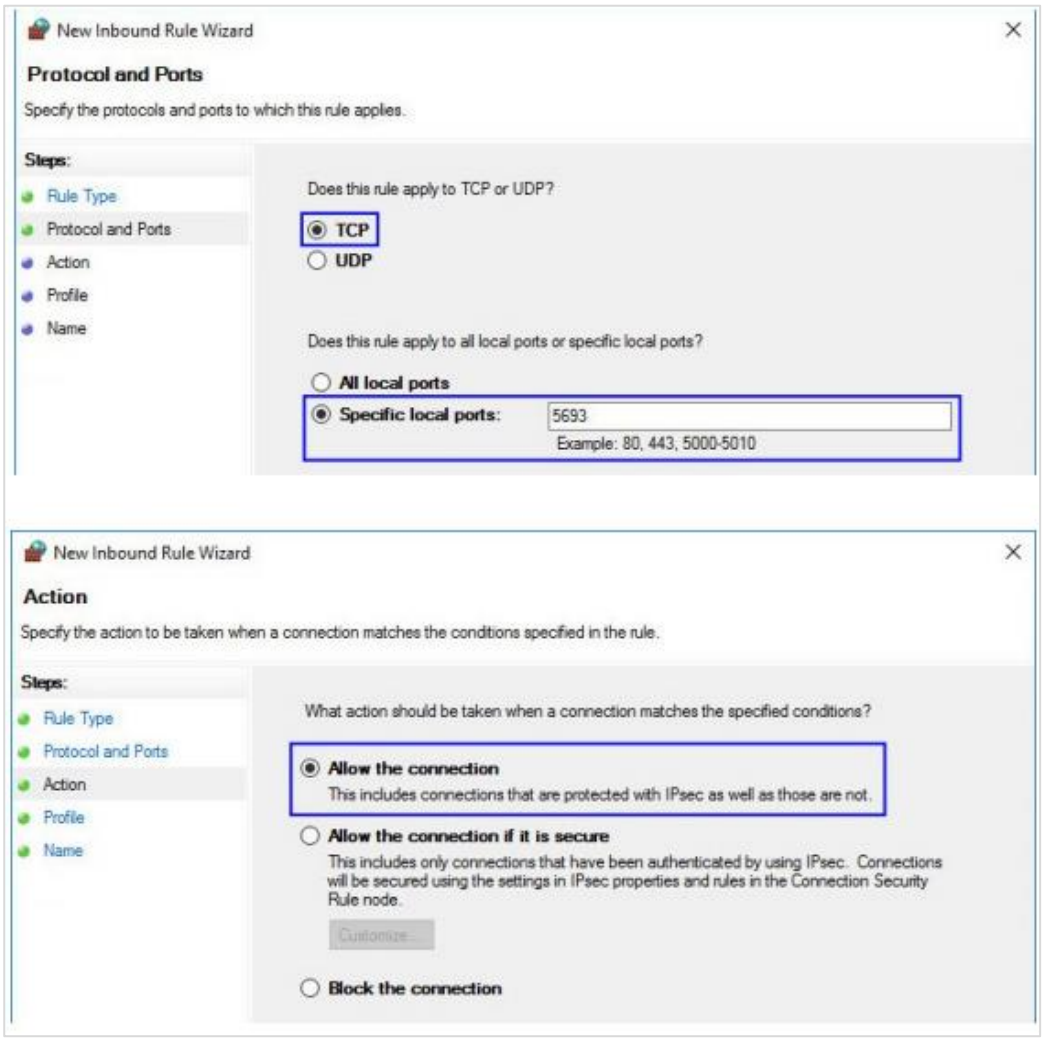
Kuvio 14 NCPA listenerin konfigurointia

Kuten aiemmin kerrottiin NCPA:ta voidaan käyttää aktiiviseen ja passiiviseen valvontaan. Kuvio 15 kuvaa passiivisen konfiguroinnin vaihetta, jossa täytyi myös määritellä token autentikointiin NRDP:lle. NRDP tokenin löytää Nagioksen graafisesta käyttöliittymästä Admin > Check Transfers > Inbound Transfers alasvetovalikosta, johon se on automaattisesti generoitu jo valmiiksi. Muissa kohdissa asetetaan NRDP:n URL-osoite, hostname, jolle tulokset kuuluvat sekä tarkistusväli ja lokitaso. Tässä ei päädytty käyttämään passiivisia tarkastuksia, koska katsottiin ettei palveluja ole niin paljoa, että ne kuormittaisivat Nagios-palvelinta liikaa.



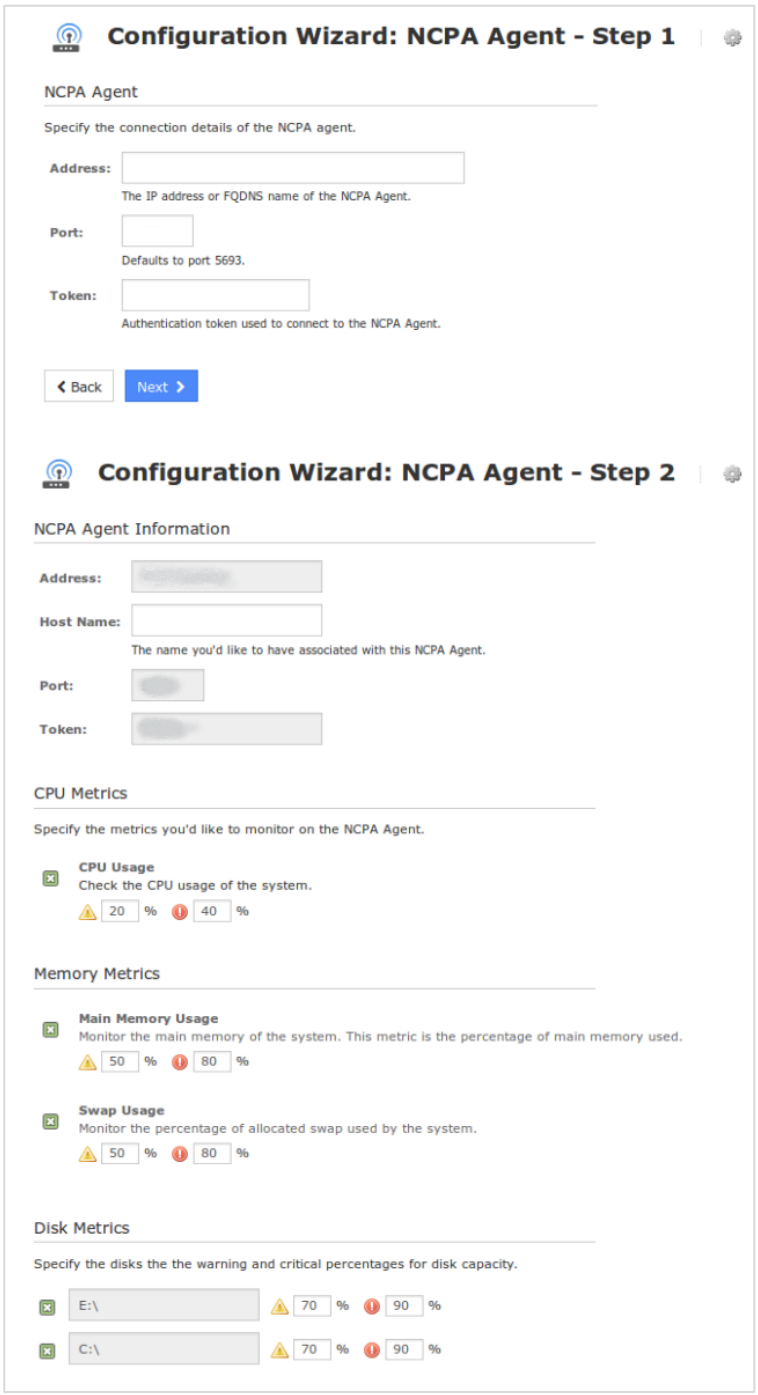
Kuvio 15 Passiivisen valvonnan asetukset

Jotta liikennöinti Nagioksen ja hostin välillä toimisi, täytyi windows palvelimella käydä tekemässä vielä palomuurille uusi sääntö. NCPA käyttää TCP porttia 5693, joten sääntönä oli sallia saapuva TCP-liikenne (engl. inbound rule) portista 5693 (ks. kuvio 16).



Kuvio 16 Palomuurisäännön luominen

Nyt kun NCPA löytyy myös hostilta, voidaan Nagioksella käyttää NCPA wizardia. Vaiheissa määritellään hostin IP-osoite tai sen FQDNS (Fully Qualified Domain Name), lisätään sama token, joka oli määriteltynä aiemmin ja valitaan haluttavat kohteet monitorointiin, sekä niiden kynnysarvot (ks. kuvio 17). Tämän jälkeen Nagios hakee monitoroitavalta laitteelta tiedot ja aloittaa valvonnan.

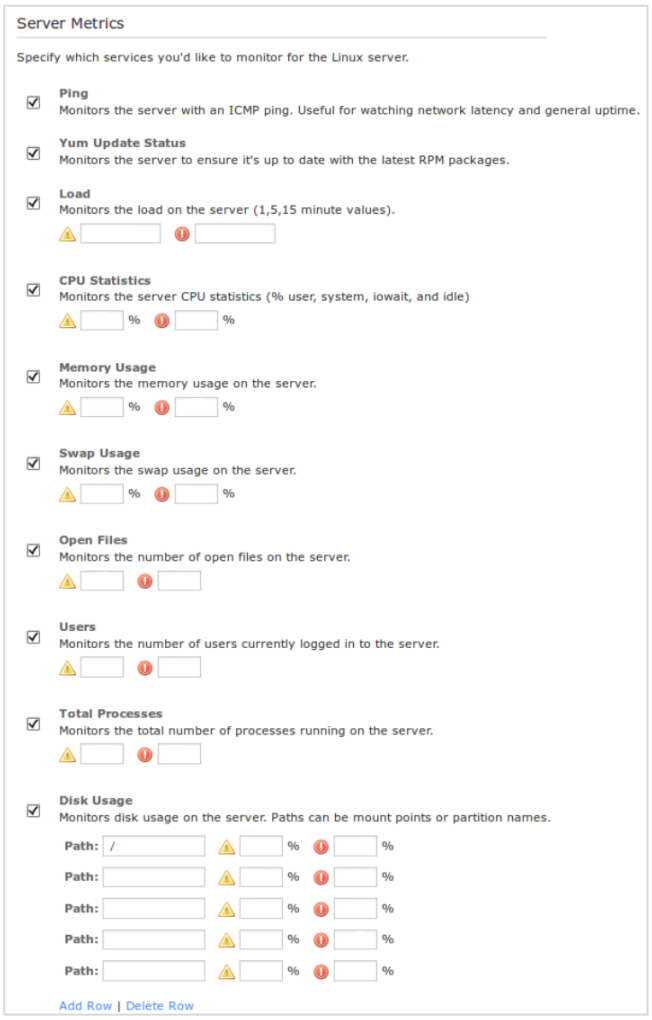


Kuvio 17 NCPA configuration wizard

5.5.3 NRPE

NRPE:tä käytettiin Linux pohjaisten palveluiden monitoroinnissa. NRPE käyttäytyy peruspiirteiltään hyvin samantyyppisesti kuin NCPA. Agentti täytyi käydä asentamassa sekä monitoroitavaan kohteeseen, sekä konfiguroida wizardin avulla Nagiokselle. NRPE:n erona on kuitenkin se, että asennuksessa linuxille ei tarvitse erikseen muodostaa palomuurisääntöjä tai tunnisteita, vaan asennusohjelmisto suorittaa kaiken valmiiksi automaattisesti. Skripti kysyi asennuksen aikana ainoastaan nagios palvelimen ip-osoitteen, jolla keskustelu sallitaan daemonin ja pluginin välillä.

Seuraavaksi konfiguroitiin vielä Nagios GUI:n puolella kohteelle haluttavat parametrit valvontaan käyttäen NRPE wizardia. Wizard tarvitsi kohteen ip-osoitteen ja pudotusvalikosta valittiin oikea käyttöjärjestelmä, joka siellä on käytössä. Toisessa vaiheessa päätettiin, mitä kohteita haluttiin valita valvontaan ja asetettiin niille raja-arvot hälytyksiä varten. NRPE:n tarjoamat monitorointikohteet näkyvät kuviossa 18.



Kuvio 18 Wizardin monitorointi vaihtoehtoja

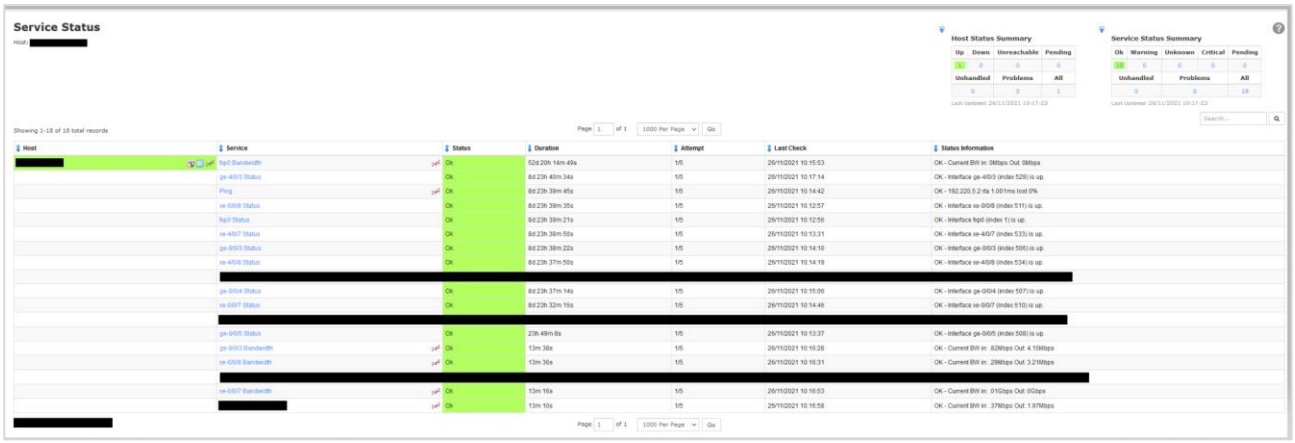
Wizardin konfiguroinnin jälkeen Nagios otti yhteyden monitoroitavaan kohteeseen ja haki sieltä valittujen kohteiden tiedot kappaleen 4.2 selitettyjen toimintamallien avulla. Tietojen hakemisen jälkeen Nagios katsoi, olivatko ne aikaisemmin määriteltyjen rajojen sisällä ja teki päätökset niiden mukaan.

5.6 Palveluiden visualisointi

Datan visualisoinnin tulee olla ymmärrettävää ja tehokasta, jotta se olisi toimivaa. Monitoroinnissa visualisoinnin pääpisteenä ei ole ulkonäöllisesti kaunis lopputulos vaan visuaalisesti selkeä ja

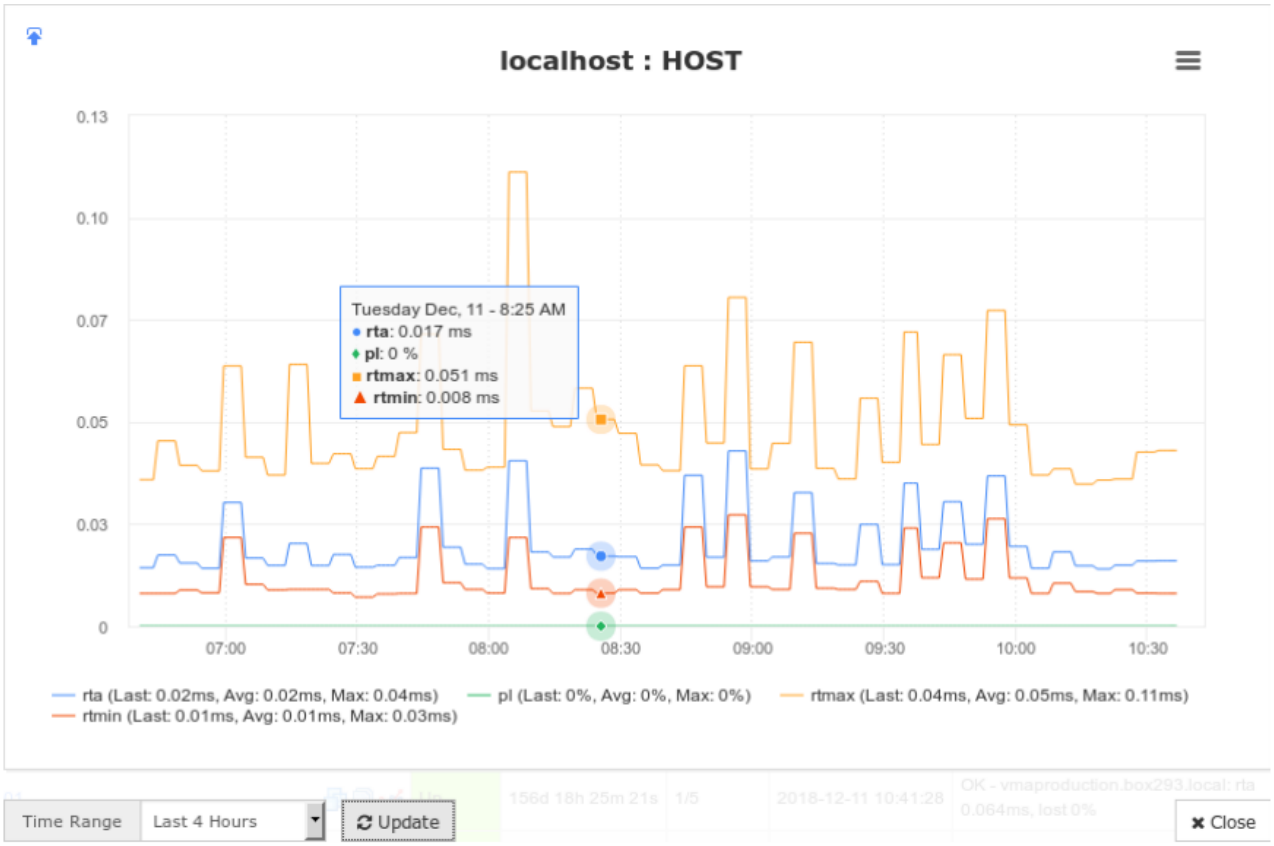
luettava kokonaisuus. Suunnittelijat usein epäonnistuvat luomalla upeita datavisualisointeja, jotka epäonnistuvat niiden päätarkoituksessa, tiedon välittämisessä (Friedman 2008).

Visualisointi jakautui pääpiirteittäin kahteen eri tapaan, suorituskyky- (engl. performance) ja tilatietoihin (engl. status). Tilatietograafit esittävät hostin tietoja kuvion 4 statuskoodien tavalla. Tilatieto voi olla esimerkiksi reitittimen ja sen porttien saatavuudesta. Kuvio 19 esittää erään reitittimen tilatietoja, joissa näkyy sen porttien statuksia ja kaistanleveyksiä, sekä ICMP-pingin avulla sen saatavuus.



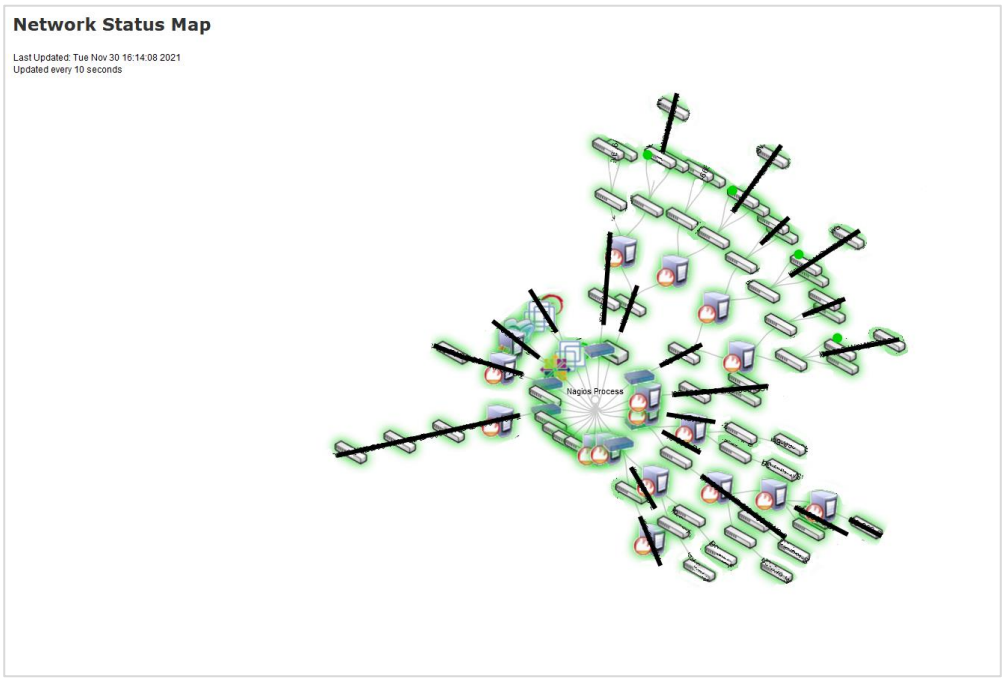
Kuvio 19 Reitittimen tilatietoja

Suorituskykyvalvonta sisältää pääasiassa kaistanleveyden, läpäisykyvyn ja latenssin seurantaa. Dataa esitetään diagrammeilla, josta nähdään helposti palvelun muutoksia eri aikavälillä. Suorituskykytiedoissa kuvaillaan tietoa tarkemmin verrattuna tilatietoihin. Se sisältää enemmän numeraalisia arvoja, esimerkiksi CPU:n prosentuaalista käyttöastetta tai viiveen arvoa millisekunteina. Erilaisia kaavioita pystyi myös kustomoida itse haluamansa näköiseksi Graph Explorerin avulla. Graph Explorerilla pystyi esimerkiksi muuttamaan graafin skaalautuvuutta ja lisäämään yhteen diagrammiin useita eri palveluita tai hosteja vertailuun (ks. kuvio 20).



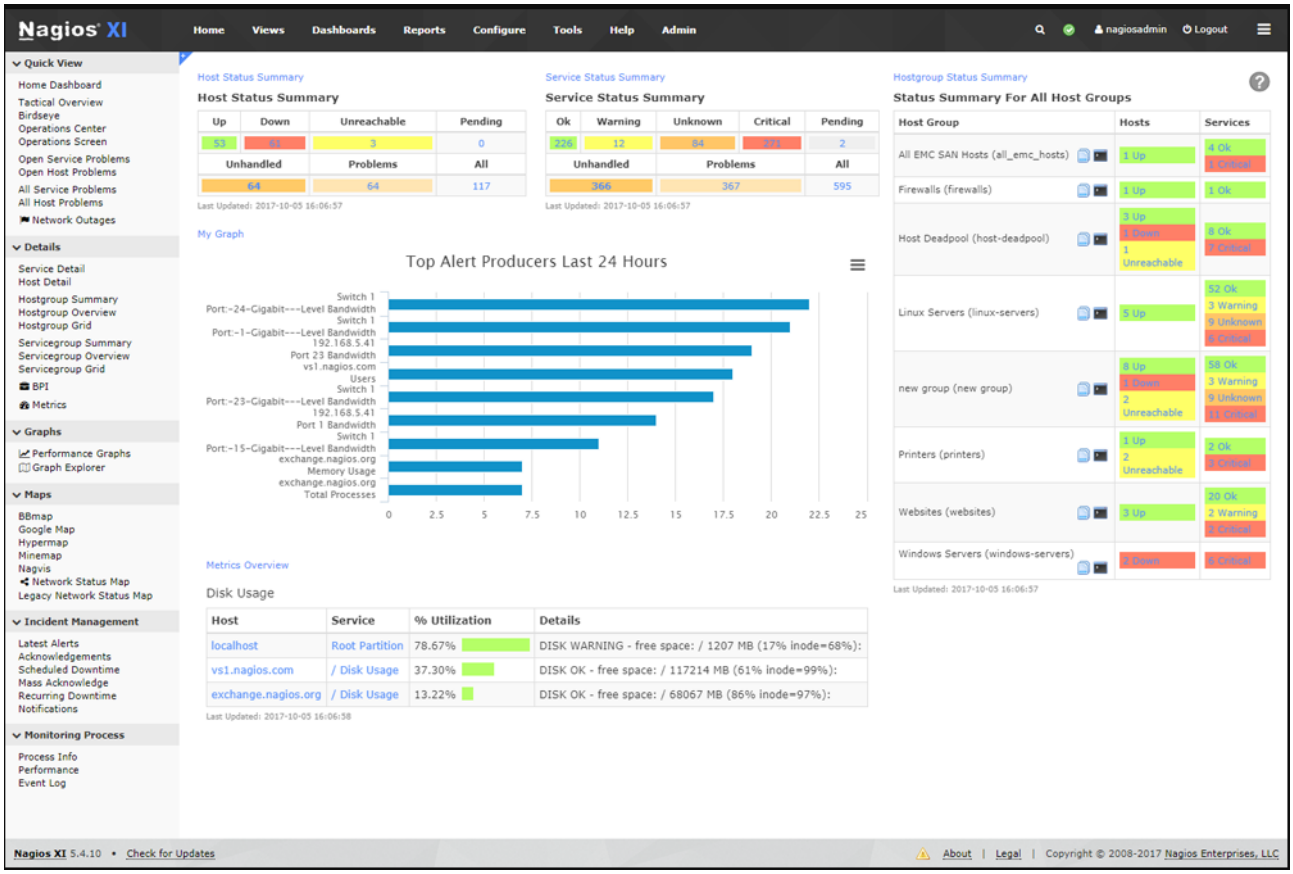
Kuvio 20 Esimerkki suorituskyky diagrammista (Nagios XI n.d)

Verkon kokonaistilanteen seurantaan käytettiin karttamallista visualisointia (ks. kuvio 21). Nagioksella laitteiden väliset suhteet toimivat parent-child-suhteilla. Hierarkiassa aina ylempi laite (parent) on vastuussa alemman tason laitteista (child). Jos Parent hostiin ei saada enää yhteyttä, katkeaa yhteys myös sen alla oleviin child hosteihin, ellei niihin ole muuta korvaavaa reittiä. Hierarkia estää päällekkäisten ilmoitusten syntymisen ja nopeuttaa korjaustoimenpiteitä, kun tiedetään mistä lähdetään vianselvityksessä liikkeelle. Laitteiden väliset suhteet määritetään Nagioksen Core Config Managerissa ja yhdellä laitteella voi useita suhteita sekä ylöspäin että alaspäin.



Kuvio 21 Network status map

Tärkeimmät monitoroitavat kohteet voidaan kerätä yhdelle tai useammalle eri välilehdelle, dashboardille. Dashboardit voivat olla joko jokaiselle käyttäjälle yksilöityjä tai yhteisesti jaettavia kokonaisuuksia. Dashboardilla pystyy esittämään lähes kaikkia Nagioksen piirtämiä graafeja ja se auttaa näyttämään käyttäjälle kaikki oleellisimmat tiedot suoraan yhdestä paikasta helpottaen verkon suorituskyvyn ja kokonaistilanteen valvontaa. Dashboardilla olevat graafit ovat kelluvia ikkunoita, joita pystyy järjestelemään haluamaansa järjestykseen ja niitä voidaan lisätä eri laitteista ja palveluista. Kuviossa 22 on esimerkki dashboardista, jossa on monitoroituna muun muassa levyn käyttöä, sekä eri hostien ja palvelujen statuksia.



Kuvio 22 Esimerkki Dashboard (Nagios XI n.d)

Ongelman sattuessa ja siitä syntyvän hälytyksen aiheutuessa olisi tärkeää, että työnjako viankorjauksessa olisi selkeä, eikä useampi verkonvalvoja tee päällekkäisiä muutoksia voiden aiheuttaa näin vielä suuremman ongelman. Nagioksella hälytyksen kuittaus tapahtuu Mass Acknowledge-komponentin kautta (ks. kuvio 23). Kuittauksen yhteydessä voidaan määritellä palvelun seisokkiaika, jotta vältytään turhilta hälytyksiltä samasta ongelmasta. Hälytykselle voidaan asettaa myös kolme erilaista kuittaustyyppiä, jotka ovat sticky, notify ja persistent. Sticky-kuittaus säilyy, kunnes host tai palvelu palaa takaisin normaalitilaansa ja häviää sen jälkeen ilmoitustaululta. Notify-kuittauksessa lähetetään kaikille yhteyshenkilöille viesti, että hälytys on kuitattu. Persistent-kuittaus on pysyvä ja se säilyy niin pitkään, kunnes se manuaalisesti poistetaan tai kun Nagios palvelin käynnistetään uudelleen.

Kuvio 23 Mass Acknowledge

5.7 Reititystaulujen valvonta

Yhtenä tutkimuskohteena oli testata, kuinka hyvin Nagios XI:llä pystyisi valvomaan reititystauluja. Jotta verkossa lähetettävät paketit löytäisivät perille, tarvitsee sitä kuljettavalla laitteella olla jokin tieto siitä, mihin suuntaan se paketin laittaa eteenpäin. Reititystaulu on lista, jossa on merkattuna kaikki verkot, joiden reitit tunnetaan. Kun esimerkiksi reititin vastaanottaa paketin, se tutkii paketin kohde IP-osoitetta ja etsii omasta reititystaulustaan siihen sopivia reititystietoja. Sopivien tietojen löydyttyä se lähettää paketin eteenpäin parasta katsomaansa reittiä pitkin. (Cisco Networking Academy 2014, 64.)

Parhaan polun määrittämiseen kuuluu useiden reittien arviointi samaan kohdeverkkoon ja optimaalisen tai lyhimmän polun valitseminen kyseiseen verkkoon. Aina kun samaan verkkoon on useita polkuja, jokainen polku käyttää erilaista poistumisliitäntää reitittimessä päästäkseen kyseiseen verkkoon. Paras polku valitaan reititysprotokollalla sen arvon tai mittarin perusteella, jota se käyttää verkkoon pääsyn etäisyyden määrittämiseen. Mittari on määrällinen arvo, jota käytetään mittaamaan etäisyys tiettyyn verkkoon. Paras polku verkkoon on polku, jolla on alhaisin arvo. (Cisco Networking Academy 2014, 66–67.)

Reittitaulukoihin liittyvät konfiguraatiomuutokset voivat johtua tahattomista tai tahallisista muutoksista, jotka voivat johtaa hallitsemattomaan verkkoliikenteeseen. Yksi tahallista muutoksista on Routing Table Poisoning (RTP). Tämä tehdään muokkaamalla reitittimien ilmoittamia reititystietojen päivityspaketteja, jotka lähetetään naapurireitittimille. Se estää datapakettien lähettämistä

polun määränpäähän. RTP muuttaa hyppyjen määrän äärettömäksi eli tehden hyppyjen määrästä suuremman kuin sallittu hyppyjen enimmäismäärä. Kun reititysprotokolla havaitsee virheellisen reitin, kaikille verkon reitittimille ilmoitetaan, että tietyllä reitillä on ääretön etäisyys. Tämä saa kaikki reitin solmut näyttämään mahdottomilta saavuttaa, mikä estää reitittimiä lähettämästä paketteja virheellisen reitin kautta ja näin ollen jokainen pudottaa sen pois reititystaulustaan. Esimerkiksi reititysprotokolla RIP:n (Routing Information Protocol) tapauksessa hyppyjen enimmäismäärä on 15, joten sen hyppyjen määrä muutetaan arvoon 16. Tämä tekee polusta saavuttamattoman eikä sitä enää käytetä reitittämiseen. (Route Poisoning 2016.)

5.7.1 Toteutus

Juniperin laitteistolle on kehitetty erikseen Python-kirjasto Junos PyEZ, jolla voi hallita ja automatisoida Juniperin Junos käyttöjärjestelmää. Junos PyEZ mahdollistaa yhteyden muodostamisen suoraan laitteeseen käyttämällä konsoliyhteyttä, Telnetiä tai NETCONF-istuntoa SSH:n kautta.

Tässä työssä reititystaulujen valvontaan käytettiin Python -skriptiä, joka konfiguroitiin pyörimään Nagioksen palvelimelle. Koodin on alkuperäisenä tehnyt Matt Schmitz vuonna 2018, joka on sitten muokattu tähän työhön sopivaksi (Schmitz 2018). Skriptiin kirjataan halutun reitittimen IP-osoite, josta se hakee sen reitittimen reititystaulut ja parsii listalle halutut reitit. Halutut reitit kirjataan temp-fileen eli väliaikaiseen tiedostoon, joka säilyy siihen asti, kunnes uusi tiedosto luodaan sen päälle. Skripti tarkistaa reititystaulut viiden minuutin välein eli se on konfiguroitu toimimaan samalla tavalla kuin Nagioksen plugin, ajastukseen pohjautuvana tehtävänä (engl. Cron job). Vertailun logiikka on yksinkertainen. Jos reitittimestä vedetty nykyinen reittitaulu on sama kuin tilapäistiedostossa (viimeisestä suorituksesta), oletetaan, että muutoksia ei ole tapahtunut - ja komentosarja päättyy. Jos taas ne eivät täsmää, jokin on muuttunut ja skripti tallentaa uuden temp-filen. Python skripti löytyy kokonaisuudessaan liitteestä 2.

Nagioksen GUI:n puolella valvonta on suoritettu siten, että erillinen plugin valvoo reititystaulujen temp filen md5 tarkistussummaa (engl. checksum). Md5 on 128-bittinen tarkistussummatyyppi, joka muodostetaan tiedostolle. Tarkistussumman pitäisi säilyä samana, jos tiedostoon ei tule mitään muutoksia. Siinä tapauksessa, jos tiedostoon on tehty pieniäkin muutoksia, vaihtuu samalla tarkistussumma erilaiseksi. Nagioksen pluginiin lisättiin kynnysarvoksi md5 tarkistussumma, jolloin

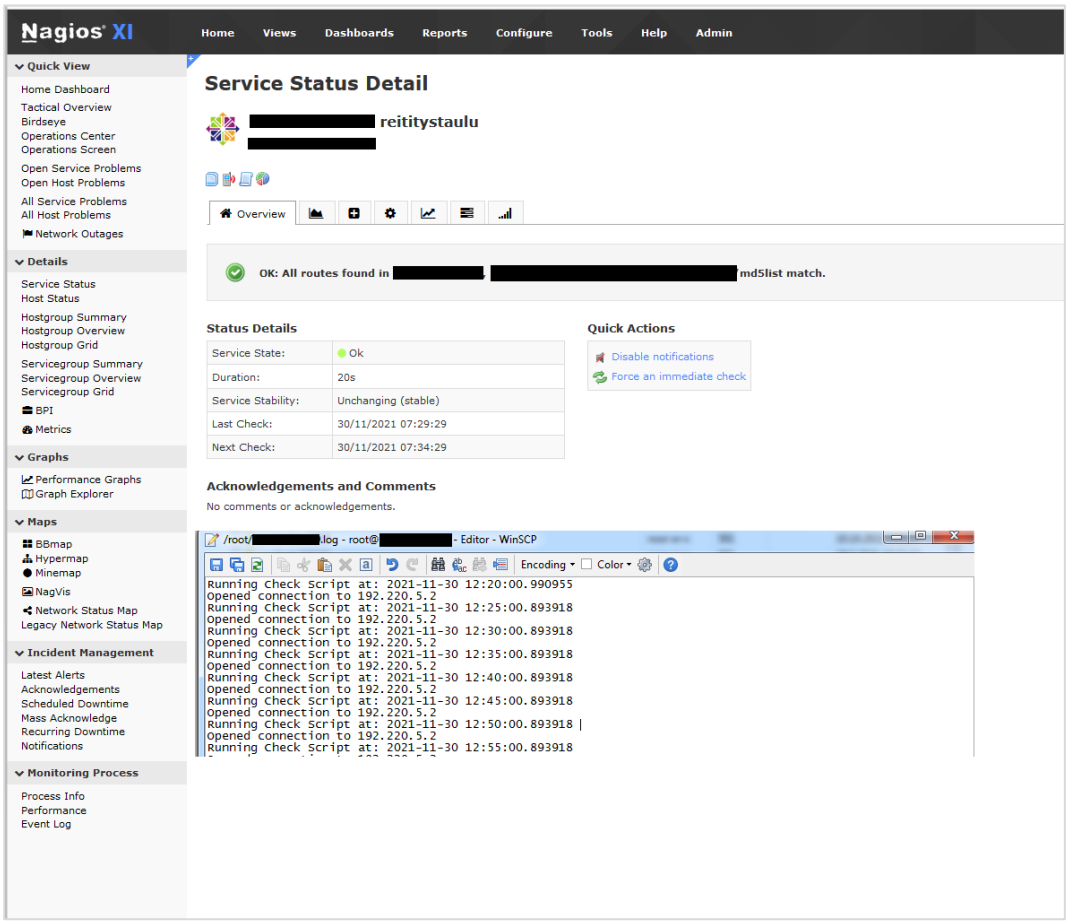
tilanne oli normaali eli kaikki halutut reitit löytyivät temp filestä. Kun muutoksia tapahtuu, vaihtuu myös temp filen tarkistussumma, joka laukaisee samalla hälytyksen Nagioksen GUI:ssa.

5.7.2 Testaus

Testi aloitettiin käynnistämällä python koodi pyörimään taustalla Nagios-palvelimella.

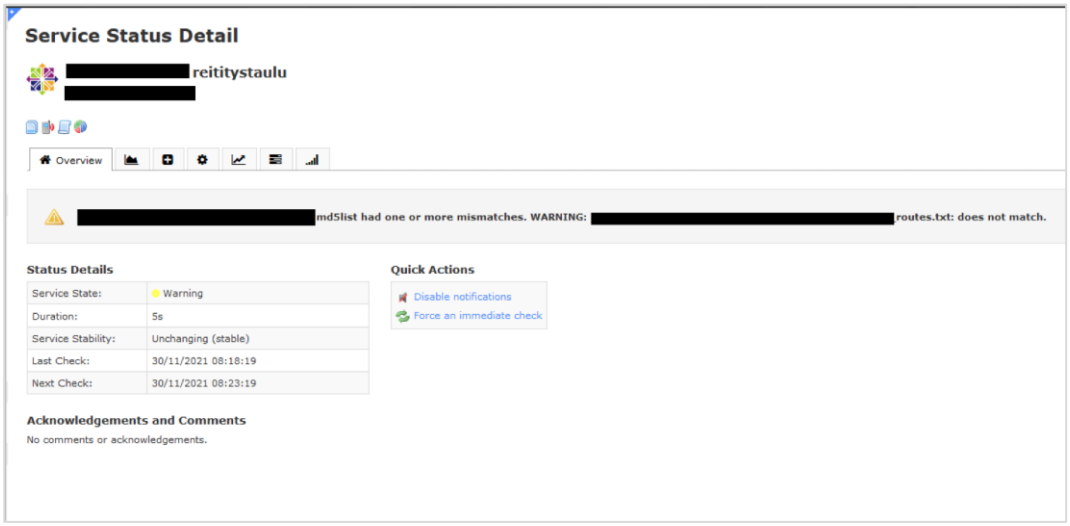
python3 *reititystaulukoodi.py* **&**

Tässä vaiheessa, kun reititystaulun tietoja ei vielä löydy, plugin luo uuden temp-filen, jota pidetään verkon normaalina tilanteena. Nykyiseltä temp-fileltä käydään hakemassa md5 checksum, joka lisätään tarkistussummaa valvovaan pluginiin. Tässä vaiheessa Nagios ilmoittaa, että kaikki halutut reitit löytyvät taulusta ja status on OK (ks. kuvio 21).



Kuvio 24 Nagios GUI ja Lokitiedosto

Tämän jälkeen verkkoon tehtiin pieni muutos todentaakseen, että Nagios osaa aiheuttaa siitä hälytyksen, kun reititystauluissa tapahtuu muutos. Nyt status on muuttunut tilaan ”warning” ja Nagios ilmoittaa, että md5 checksum on vaihtunut ja reititystaulu ei ole yhteensopiva (ks. kuvio 22).



Kuvio 25 Nagios GUI:n näkymä

6 Vaatimukset ja tulokset

6.1 Verkonvalvonnan hyödyt ja tärkeys

Aiheena oli tarkastella verkonvalvonnan ja -monitoroinnin merkitystä. Tutkimuksesta käy selkeästi ilmi, miten tärkeä rooli valvonnalla ja monitoroinnilla on nykyajan tietoverkoissa. Palveluiden siirtyessä yhä enemmän digitaaliseen muotoon, tulee verkon olla toimiva ja luotettava taatakseen yrityksen tuottavuuden. Pienetkin hallitsemattomat katkokset verkossa vaikuttavat negatiivisesti yrityksen imagoon, mikä aiheuttaa heti turhia tappioita. Nykyaikana vastuu kasvaa entisestään. Kun pandemia pakotti ihmiset työskentelemään enemmän etätöissä, joutuivat yritykset miettimään strategioita, joilla voitaisiin tarjota yhtä hyvät työskentelyolosuhteet sekä etä-, että toimistotyöntekijöille.

Parhaimmassa tapauksessa verkossa ongelmat ehditään tunnistamaan ennen kuin käyttökatkoksia kerkeää tapahtua. Tässä esiin nousee verkon suorituskyvyn monitoroinnin tärkeys. Kun työntekijöille tarjotaan työkalut, joita he tarvitsevat kommunikoidakseen keskenään ja tehdäkseen työnsä ilman häiriötekijöitä, tulee liiketoiminnasta tuottavampaa. Suorituskyvyn mittaus auttaa myös

ennalta ehkäisemään käyttökatkoksia, kun tiedetään, mikä on verkon ”normaalitila” ja näin ollen erikoisiin piikkeihin verkossa voidaan kiinnittää huomiota ja tehdä mahdollisia korjaustoimenpiteitä. Myöskin tietoturvan kannalta epätavallinen liikenne huomataan helpommin, mikä lisää verkon tietoturvallisuutta. Oikein optimoitu verkko ja sen kapasiteetin tunnistaminen mahdollistaa myös sen skaalautumisen suuremmaksi, joka lisää yrityksen liikevaihtoa.

Jos kuitenkin käy tilanne, joka aiheuttaa katkoksen verkkoon, tulee ongelman olla tunnistettavissa nopeasti. Valvonnan tulee olla toteutettu selkeästi, jotta se olisi tehokasta. Oikein priorisoidut hälytykset ja kohdistettu monitorointi lyhentää katkosaikoja merkittävällä tavalla.

6.2 Nagios XI monitorointiohjelmana

Työssä tarkasteltiin Nagios XI:n soveltuvuutta verkon monitoroinnissa. Nagios XI on tehty yksinkertaiseksi, mutta tehokkaaksi monitorointisovellukseksi. Sen etuna on juuri sen käytettävyys, verkonvalvojalla ei tarvitse olla niin paljon teknistä osaamista, kuinka järjestelmä toimii taustalla saadakseen kuitenkin rakennettua toimivan monitorointiratkaisun. Configuration wizardit ovat hyvin ohjattuja step-by-step-tyyppisiä asennuksia, jotka tekevät konfiguroinnista helpompaa. Nagios tarjoaa myös skaalautuvuutta suurella määrällä avoimen lähdekoodin lisäosia, joilla pystyy mukauttamaan palvelunsa juuri sellaiseksi kuin haluaa. Samalla avoimen lähdekoodin käyttämissä lisäosissa piilee myös huonotkin puolet. Jotkut lisäosat saattavat olla toiminnaltaan monimutkaisia, joka vaatii käyttäjältä lisää asiantuntemusta. Toisena ongelmana on tietoturva, sillä kun kuka tahansa saa lisätä ohjelmistoja ilman sen tarkempaa valvontaa, voivat jotkut käyttää tätä hyödykseen. Lisäosiin voidaan naamioida asioita, jotka nuuskivat verkosta tietoturva-aukkoja. Kolmantena on tuen puute, joka on yksi pahimmista haittapuolista. Nagioksella on suuri joukko käyttäjiä, jotka jakavat henkilökohtaisen kokemuksensa tuotteen käytöstä, mutta se ei voi korvata asiakastukea, joka on erityisesti koulutettu kyseisen palvelun käyttämiseen.

Työssä huomattiin, että Nagios tuntui painottuvan enemmän palvelimien ja päätelaitteiden monitorointiin. Tämä osoittautui haasteeksi, sillä kohdeverkko sisälsi enemmän reitittimiä, kytkimiä sekä palomuureja. Niiden valvonnassa haasteen toi se, että piti tarkkaan rajata mitä portteja valvottaisiin, sillä Nagios esimerkiksi antoi käyttämättömistä porteista turhia hälytyksiä. Työn aikana

huomattiin, että jo olemassa oleva valvontasovellus verkon reitittimille, kytkimille ja palomuureille olisi selkeäkäyttöisempi, joten Nagiosta voisi enemmän käyttää juuri palvelinpuolen valvonnassa.

6.3 Reititystaulujen valvonta

Yhtenä tutkimusaiheena oli tutkia, kuinka hyvin Nagioksella pystyisi valvomaan reititystaulun muutoksia. Reititystaulujen valvonnan tuottamisessa haasteena oli aineiston puute. Aiheesta löytyi hyvin heikosti aikaisempia toteutuksia, joten palvelu jouduttiin tuottamaan eri palasista, mikä ei ole paras mahdollinen lähtökohta pyrkiessä tekemään siitä luotettavan. Tavoitteeseen päästiin osittain, sillä palvelu todettiin toimivaksi ja Nagios osasi ilmoittaa tapahtuvista muutoksista. Ajanpuutteen vuoksi toteutus jäi hieman suunniteltua yksinkertaisemmaksi ja ei olisi käytettävyyden ja tietoturvan kannalta paras mahdollinen ratkaisu.

Palvelua pystyisi lisäkehittämään automatisoimalla molempia lisäosia enemmän. Python koodiin pystyisi lisäämään kohdan, joka kertoisi mitkä reitit löytyvät edelleen taulusta ja mitkä eivät. Tämä helpottaisi vianetsintää erittäin paljon. Myöskin md5 tarkistussumman lisääminen käsin on epäkäytännöllistä, joten tarkistussumman päivittyminen automaattisesti olisi järkevämpää.

6.4 Pohdinta

Työssä eniten yllätyksenä tuli se, miten aikaa vievää uuden monitorointipalvelun pystyttäminen on ja miten tärkeää on suunnitella kerralla toimiva ratkaisu. Ilman aikaisempaa kokemusta verkonvalvonnasta ja sen monitoroinnista, kului aikaa paljon teorian lukemiseen ja käytännön toteutuksessa yrityksen ja erehdyksen kautta oppimiseen.

Työn ensimmäisessä osassa hyödynnettiin paljon eri lähteitä aiheen tarkastelussa ja onnistuttiin vastaamaan hyvin tutkimuskysymyksiin ja näin ollen teoriaosuudessa tavoitteeseen päästiin. Toisessa osassa aikataulutuksen kanssa tuli hieman kiireitä ja tulos ei ollut aivan halutunlainen, mutta kuitenkin toimiva. Suurimpina haasteina oli rajata mitä palveluita olisi järkevin monitoroida, jotta valvonta pysyisi tehokkaana. Haasteita aiheutti myös se, että käytännön osuus suoritettiin ennen kunnollista syventymistä teoriaosuuteen, joka näkyy myös käytännön osuuden valinnoissa.

Kirjoitusvaiheen haasteina oli rajata, kuinka tarkasti tulisi selittää mistäkin aiheesta. Työ sisälsi laajoja kokonaisuuksia, joiden tiivistäminen lyhyeen, mutta ymmärrettävään muotoon oli haastavaa. Esimerkiksi Nagios sisältää todella paljon erilaisia ominaisuuksia, joten niistä täytyi yrittää karsia oleellisimpia asioita työhön liittyen.

Työn kautta ymmärrys tietoverkoista syventyi huomattavasti, josta on varmasti hyötyä tulevaisuudessa. Työ lisäsi myös käsitystä, kuinka hyvin hoidettuna verkonvalvonta vaikuttaa koko yrityksen toimintaan. Kokemusta karttui myös linux-käyttöjärjestelmästä, sekä hieman eri ohjelmointikielistä kuten pythonista ja perlistä.

Lähteet

All About Network Alerts + Best tools. 2020. Solarwindsin verkkojulkaisu. Viitattu 20.2.2022. https://logicalread.com/network-alerts/#.YkW8DyhBwuW.

Andini, M. 2022. 5 negative ways it disruptions can affect your business. Artikkeli ProjectCubiclen sivustolla. Viitattu 2.2.2022. https://www.projectcubicle.com/5-negative-ways-it-disruptions-can-affect-your-business/.

Belding, G. 2020. ICMP protocol with Wireshark. Blogikirjoitus infosec sivustolla. Viitattu 4.2.2022. https://resources.infosecinstitute.com/topic/icmp-protocol-with-wireshark/.

Charbonneau, P. 2020. Network Fault Monitoring vs. Network Performance Monitoring. Blogikirjoitus Obkion sivustolla. Viitattu 2.2.2022. https://obkio.com/blog/fault-monitoring-vs-performance-monitoring/.

Cisco Networking Academy. 2014. Routing Protocols Companion Guide. Indianapolis, Ind. : Cisco Press.

Definitive guide to network monitoring. n.d. Artikkeli Advanced Cyber Solution -organisaation sivuilla. Viitattu 2.2.2022. https://www.advancedcyber.co.uk/the-definitive-guide-to-network-monitoring#what_is_a_network_monitoring_solution.

Emmit, J. 2020. SNMP: Understanding Simple Network Management Protocol. Blogiteksti hallinta- ja tietoturvaohjelmistoyritys Kaseyan sivuilla. Viitattu 15.2.2022. https://www.kaseya.com/blog/2020/09/14/snmp-simple-network-management-protocol/.

Galstad, E. 2017. NRPE Dovcumentation. Virallinen ohje NRPE:n käytöstä. Viitattu 5.3.2022. https://assets.nagios.com/downloads/nagioscore/docs/nrpe/NRPE.pdf.

Gerhards, R. 2009. The Syslog Protocol. IETF:n julkaisema RFC-dokumentti numero 5424. Viitattu 20.2.2022. https://datatracker.ietf.org/doc/html/rfc5424.

Harrington, D., Presuhn, R. & Wijnen, B. 2002. An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks. IETF:n julkaisema RFC-dokumentti numero 3411. Viitattu 15.2.2022. https://datatracker.ietf.org/doc/html/rfc3411.

History of Nagios. n.d. Nagioksen virallinen sivusto. Viitattu 1.3.2022. https://www.nagios.org/about/history/.

ISO/IEC 7498-4:1989. Management framework for open systems interconnection (OSI) for CCIT applications. Viitattu 2.2.2022. https://janet.finna.fi/, ITU-T Recommendations.

Jaakohuhta, H. 2005. Lähiverkot: Ethernet. Helsinki: IT Press.

Järjestelmäkeskus. n.d. Puolustusvoimien logistiikkalaitoksen infosivu järjestelmäkeskuksesta. Viitattu 10.3.2022. https://logistiikkalaitos.fi/jarjestelmakeskus.

Lerner, A. 2014. The cost of downtime. Artikkeli Gartnerin sivustolla. Viitattu 2.2.2022. https://blogs.gartner.com/andrew-lerner/2014/07/16/the-cost-of-downtime/.

Mackin, J. 2017. How Nagios XI Works. Artikkeli Nagioksen virallisella sivustolla. Viitattu 2.3.2022. https://www.nagios.com/videos/2017/07/nagios-xi-get-monitoring-xi-works/.

Macpherson, J. 2021. 2 Common types of Network Management Protocols. Artikkeli Park Place -Technologies -yrityksen sivuilla. Viitattu 15.2.2022. https://www.parkplacetechnologies.com/blog/types-of-network-management-protocols/.

Medhi, D. & Ramasamy, K. 2018. Network routing: algorithms, protocols and architectures. Cambridge: Morgan Kaufmann Publishers.

Nagios XI. n.d. Nagios XI:n virallinen sivusto. Viitattu 6.3.2022. https://www.nagios.com/products/nagios-xi/.

NCPA. n.d. Virallinen ohje NCPA:n käytöstä. Viitattu 5.3.2022. https://www.nagios.org/ncpa/help.php.

Network monitor success in 8 easy steps. n.d. Artikkeli Spiceworks -yrityksen sivustolla. Viitattu 2.2.2022. https://www.spiceworks.com/it-articles/network-monitor-success/.

Network Monitoring: Protocols, Best Practices and Tools. 2020. Solarwindsin tytäryhtiö Tek-Toolsin verkkojulkaisu. Viitattu 2.2.2022. https://www.tek-tools.com/network/network-monitoring-guide-and-tools.

Postel, J. 1981. Internet Control Message Protocol. IETF:n julkaisema RFC-dokumentti numero 792. Viitattu 20.2.2022. https://datatracker.ietf.org/doc/rfc792/.

Puolustusvoimien logistiikkalaitos. n.d. Logistiikkalaitoksen kuvaus organisaatiosta. Viitattu 10.3.2022. https://logistiikkalaitos.fi/tietoa-meista.

Route Poisoning. 2016. Technopedian verkkojulkaisu. Viitattu 20.3.2022. https://www.techopedia.com/definition/16205/route-poisoning.

Scarpati, J. 2021. Simple Network Management Protocol (SNMP). Artikkeli Techtargetin verkkosivuilla. Viitattu 15.2.2022. https://www.techtarget.com/searchnetworking/definition/SNMP.

Schmitz, M. 2018. Juniper SRX – Automated Route Monitoring. Tieteellinen artikkeli 0x2142 sivustolla. Viitattu 15.3.2022. https://0x2142.com/juniper-srx-automated-route-monitoring/.

Serie, M. 2021. Nagios Core vs. Nagios XI: 4 Key Differences. Verkkojulkaisu Nagioksen virallisilla sivuilla. Viitattu 1.3.2022. https://www.nagios.com/news/2021/07/nagios-core-vs-nagios-xi/.

Tolliver, W. 2020. SNMP traps in PRTG. blogiteksti Passlerin sivustolla. Viitattu 5.2.2022. https://blog.paessler.com/snmp-traps-in-prtg.

Tutkimus- ja kehittämistoiminta. n.d. Tilastokeskuksen verkkojulkaisu. Viitattu 10.3.2022. https://www.stat.fi/meta/kas/t_ktoiminta.html.

Twin, A. 2021. Total cost of ownership (TCO). Artikkeli Investopedian sivuilla. Viitattu 2.2.2022. https://www.investopedia.com/terms/t/totalcostofownership.asp.

Ultimate guide to network monitoring. 2019. Solarwindsin tekemä ohje verkonvalvontaan. Viitattu 2.2.2022. https://www.dnsstuff.com/network-monitoring#basics-network-monitoring.

Understanding And Using Configuration Wizards In Nagios XI. 2017. Nagioksen virallinen käyttöohje. Viitattu 2.3.2022. https://assets.nagios.com/downloads/nagiosxi/docs/Understanding-And-Using-Configuration-Wizards-In-Nagios-XI.pdf.

Using The Core Config Manager For Host Management. 2018. Nagioksen virallinen käyttöohje. Viitattu 2.3.2022. https://assets.nagios.com/downloads/nagiosxi/docs/Using-The-Nagios-XI-Core-Config-Manager-For-Host-Management.pdf.

Valtierra, M. 2017. Cloud security best practices: part 4 Application Security. Blogiteksti Medium sivustolla. Viitattu 6.2.2022. https://medium.com/@cohesivenet/cloud-security-best-practices-part-4-application-security-1985e2316355.

Velimirovic, A. 2021. Nagios Tutorial: Continuous Monitoring with Nagios Core and XI. phoenixNAP yrityksen verkkojulkaisu. Viitattu 2.3.2022. https://phoenixnap.com/blog/nagios-monitoring-tutorial.

What Is LDAP & How Does It Work?. n.d. Oktan artikkeli aiheesta LDAP. Viitattu 5.3.2022. https://www.okta.com/identity-101/what-is-ldap/.

Liitteet

Liite 1. Nagioksen hostname ja verkkoasetukset

 # hostnamectl set-hostname XXXXXX

 # vi /etc/sysconfig/network-scripts/ifcfg-ens33

TYPE="Ethernet"

PROXY_METHOD="none"

BROWSER_ONLY="no"

BOOTPROTO="none"

DEFROUTE="yes"

IPV4_FAILURE_FATAL="no"

IPV6INIT="yes"

IPV6_AUTOCONF="yes"

IPV6_DEFROUTE="yes"

IPV6_FAILURE_FATAL="no"

IPV6_ADDR_GEN_MODE="stable-privacy"

NAME="ens33"

DEVICE="ens33"

ONBOOT="yes"

ZONE=public

IPADDR0= xxx.xxx.xxx.xxx

PREFIX0=nn

GATEWAY0= xxx.xxx.xxx.xxx

DNS1= xxx.xxx.xxx.xxx

DOMAIN=

IPV6_PEERROUTES=no

IPV6_PEERDNS=no

Liite 2. Reititystaulujen valvonnan python koodi

```
from jnpr.junos import Device
from jnpr.junos.op.routes import RouteTable
import sys
import datetime
import os
import smtplib
import socket
import time



#Pakolliset tiedot

deviceName = 'TÄHÄN-REITITTIMEN-NIMI'
deviceIP = 'TÄHÄN-REITITTIMEN-IP-OSOITE'
apiuser = 'REITITTIMEN-KÄYTTÄJÄTUNNUS'
apipassword = 'KÄYTTAJÄN-SALASANA'




hostname = socket.gethostname()
logfile = "./routetable_REITITIN.log"
tempfile = "/usr/local/nagios/etc/check_md5_sums/REITITIN_routes.txt"
timestamp = datetime.datetime.now()



# Aseta reitittimen ip-osoite/käyttäjätiedot
dev = Device(deviceIP, user=apiuser, password=apipassword)


x = True
while(True):

    def checkTable():
        time.sleep(300)
        with open(logfile, 'ab') as a:
            a.write("Running Check Script at: %s \n" % timestamp )
        try:
            # Avaa istunto reitittimelle
            dev.open()
            with open(logfile, 'ab') as a:
                a.write("Opened connection to %s \n" % deviceIP)
        except:
            with open(logfile, 'ab') as a:
                a.write("ERR: FAILED TO OPEN CONNECTION TO %s \n"  % deviceIP)
            sys.exit()

        # Vedä reitittimen reititystaulu
        allroutes = RouteTable(dev)
        table = allroutes.get().keys()

        # Sulje istunto reitittimelle
        dev.close()

        # Tarkasta löytyykö temp-fileä entuudestaan
        # jos ei, tee uusi temp-file
        if not os.path.isfile(tempfile):
            with open(tempfile, 'ab') as a:
                a.write(str(table))
            sys.exit()

        # Paikallinen tiedosto, jota käytetään pitämään kirjaa opituista reiteistä
        with open(tempfile, 'r+b') as a:
            lastroutes = a.readlines()
            # Vertaile onko tullut muutoksia
            if str(table) == str(lastroutes):
                sys.exit()
            if str(table) != str(lastroutes):
                pass
        # Poista vanha file ja tee uusi
        os.remove(tempfile)
        with open(tempfile, 'w+b') as a:
            a.write(str(table)


    if __name__ == '__main__':
        checkTable()
```