VERKONVALVONTATOTEUTUS

Janne Lehmonen

Opinnäytetyö
Toukokuu 2012

Tietoverkkotekniikka
Tekniikan ja liikenteen ala

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖN
KUVAILULEHTI

Tekijä(t) LEHMONEN, Janne	Julkaisun laji Opinnäytetyö	Päivämäärä 7.5.2012
	Sivumäärä 50	Julkaisun kieli SUOMI
	Luottamuksellisuus ( ) saakka	Verkkojulkaisulupa myönnetty (X)

Työn nimi
VERKONVALVONTATOTEUTUS

Koulutusohjelma
Tietoverkkotekniikka

Työn ohjaaja(t)
KOTIKOSKI, Sampo

Toimeksiantaja(t)
Ilmasotakoulu

Tiivistelmä

Opinnäytetyön tarkoituksena oli etsiä ja toteuttaa verkonvalvontaohjelmisto Ilmasotakoulussa sijaitsevaan suljettuun tietoverkkoon. Ilmasotakoulu on korkeakoulutasoinen sotilasopetuslaitos, ja se on yksi kolmesta puolustushaarakoulusta Puolustusvoimissa. Opetuslaitoksen tehtävänä on tuottaa osaajia rauhan-, kriisin- ja sodan ajan tehtäviin.

Verkonvalvonnan tarve on korostunut nykypäivänä verkkojen monipuolistumisen ja kasvamisen mukana. Verkonvalvonta on laaja käsite, joten työtä rajattiin liittyväksi vikojen ja suorituskyvyn valvontaan. Verkonvalvonta on toteutettu pääsääntöisesti käyttämällä SNMP:tä, joka on yleisesti käytetty protokolla verkonvalvonnassa.

Opinnäytetyö aloitettiin keräämällä tietopohjaa verkonvalvonnasta, minkä jälkeen määriteltiin verkonvalvontaohjelmiston vaatimukset. Verkonvalvontaohjelmistoja on olemassa nykyään todella paljon, joten kokemuspohjan kartuttamiseksi käytettiin avuksi myös muita samanaiheisia opinnäytetöitä. Vertailujen ja tutkimisen jälkeen verkonvalvontaohjelmistoksi päädyttiin valitsemaan Nagios Core.

Nagioksen avulla toteutettiin valvonnat tietoverkon kriittisimmille toiminnoille ja palveluille. Nagiokseen asennettiin myös kaksi lisäosaa Nconf ja PNP4Nagios, joiden avulla helpotettiin Nagioksen konfiguroimista ja saatiin piirrettyä valvottavista kohteista kuvioita. Verkonvalvontatoteutuksesta järjestettiin myös koulutustilaisuus, johon osallistuivat kaikki verkon järjestelmänvalvojat.

Lopputuloksena opinnäytetyöstä saatiin toimiva ja hyvin mukautuva verkonvalvontatoteutus, joka vastaa ja soveltuu hyvin toimeksiantajan määrittelemiin tarpeisiin.

Avainsanat (asiasanat) Verkonvalvonta, SNMP, MIB, Nagios, Nconf, PNP4Nagios

Muut tiedot

Author(s)
LEHMONEN, Janne

Type of publication
Bachelor's Thesis

Date
7.5.2012

Pages
50

Language
FINNISH

Confidential

( ) Until

Permission for web publication
(X)

Title
IMPLMENTATION OF NETWORK MONITORING

Degree Programme
Information Technology

Tutor(s)
KOTIKOSKI, Sampo

Assigned by
Air force Academy

Abstract

The purpose of the thesis was to look for and to carry out the network monitoring system to the closed data network located in air force academy. The air force academy is a university level military institution and it is one of the three defence branch schools in the Finnish Defence Forces. The task of the air force academy is to produce experts to the peace, crisis and wartime tasks.

The need for the network monitoring has become more marked during the present with the diversification and growth of networks. The network monitoring is a wide concept; therefore the work was marked off as to join the supervision of a faults ability and capacity. The network monitoring has been carried out by using mostly the SNMP a network monitoring protocol that has been generally used.

The thesis project started by collecting basic information of network monitoring and after that it was continued specifying the demands of the network monitoring system. Nowadays a great number of network monitoring systems exist therefore other theses with the same subject were also used as help to increase the experience and knowledge about the network monitoring systems. After the comparisons and examining, Nagios Core was chosen for the network monitoring system.

Nagios was carried out to help supervise the most critical functions and services of the data network. Nagios was installed with two additional parts, namely Nconf and PNP4Nagios to help to configure Nagios and to create ability to draw figures about the supervised targets. A training meeting about the network monitoring system was also arranged for the system supervisors of the network.

As a final result of this thesis, an adaptable and well operating network monitor system was implemented. The system meets the requirements defined by the assigner and applies well to what was obtained as the final result.

Keywords
Network monitoring, SNMP, MIB, Nagios, Nconf, PNP4Nagios

Miscellaneous

SISÄLTÖ

KUVIOT

TAULUKOT

LYHENTEET

CPU	Central Processing Unit
DNS	Domain Name System
HMAC	Hash Message Authentication Code
IETF	Internet Engineering Task Force
IP	Internet Protocol
ITU	International Telecommunication Union
JAMK	Jyväskylän Ammattikorkeakoulu Oy
MAC	Media Access Control
MIB	Management Information Base
NAGIOS	Verkonvalvontaohjelmisto
NMS	Network Management System
OID	Object Identifier
RAID	Redundant Array of Independent Disks
RFC	Requets for Comments
RRD	Round Robin Databases
SMI	Structure of Management Information

SMS	Short Message Service
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UPS	Uninterruptible Power Supply
US-CERT	United States Computer Emergency Readiness Team

1 VERKONVALVONNASTA TEHOA VERKONHALLINTAAN

Nykypäivänä kaikki koulutus- ja tietojärjestelmät ovat yhä enemmän ja enemmän tietokonepohjaisia ja toimivat tietoverkkojen päällä. Tästä johtuen tietoverkkojen määrä ja niiden palvelut kasvavat koko ajan. Tämä on tilanne myös Puolustusvoimissa. Verkkojen monimutkaistuessa ja laajentuessa tulee eteen väistämättä myös haasteita sekä ongelmia. Tämän takia verkonylläpitäjien täytyy jatkuvasti hakea uusia ratkaisuja verkonlaitteiden valvontaan ja palveluiden saatavuuden varmistamiseen. Verkonvalvontaohjelmistot ovat yksi vaihtoehto vastaamaan näihin haasteisiin.

Opinnäytetyön tavoitteena oli löytää sopiva verkonvalvontaohjelmisto ja toteuttaa sen avulla verkonvalvontatoteutus Ilmasotakoulussa sijaitsevaan suljettuun tietoverkkoon. Aikaisempaa verkonvalvontaa ei ollut olemassa. Verkonvalvonnan toteuttamiseen päädyttiin pääasiassa verkonpalveluiden ja verkon monipuolistumisen takia. Lisäksi perusteina olivat järjestelmänvalvojien työn helpottaminen, vikatilanteiden helpompi toteaminen ja ratkaiseminen sekä verkon ja palveluiden kapasiteetin riittävyyden reaaliaikainen varmistaminen. Myös henkilöresurssien jatkuva pienentyminen pakottaa kehittämään verkonhallintaan käytettäviä ratkaisuja. Tällä ei tarkoiteta henkilöiden vähentymistä, vaan työmäärän lisääntymistä ja monipuolistumista henkilöä kohden. Aiheena verkonvalvonta on laaja, joten tässä työssä keskityttiin verkon vikojen ja suorituskyvyn valvontaan. Käytännössä tämä tarkoittaa laitteiden saatavuuden, käyttöasteen ja kapasiteetin valvontaa.

Toimeksiantajana oli Ilmasotakoulu. Ilmasotakoulu on korkeakoulutasoinen sotilasopetuslaitos, ja se on yksi kolmesta puolustushaarakoulusta Puolustusvoimissa. Opetuslaitoksen tehtävänä on tuottaa osaajia rauhan, kriisin ja sodan ajan tehtäviin. Opetuslaitoksen vastuualueena on ilmavoimien ja ilmatorjunnan johtamisjärjestelmien, ilmapuolustushenkilöstön ja ilmavoimien eri kalustojärjestelmienkoulutus. Lisäksi se toimii Ilmavoimien johtamisjärjestelmän sekä ilmatorjunnan aselajikouluna.

2 VERKONVALVONTA

2.1 Yleistä

Verkonvalvonnalla tarkoitetaan yleisesti verkkolaitteiden valvontaa, ei niinkään verkossa liikkuvan liikenteen seuraamista tai valvomista. Sisäisen liikenteen valvontaan on olemassa eri ohjelmistot. Verkonvalvonta yleisesti liittyy osaksi verkonhallintaan, joka on isompi kokonaisuus. Luvussa 2 käydään läpi verkonvalvontaa yleisesti ja sitä, miten se liittyy osaksi verkonhallintaa. Lisäksi luvussa käydään läpi verkonvalvonnan eri osa-alueita perustuen ITU-T x.700 (International Telecommunication Union) suositukseen.

Verkonvalvonta koostuu yleisesti valvovasta ohjelmistosta, valvottavasta laitteesta ja protokollasta, joka hoitaa kyselyn. Verkonvalvonnalla ei suoranaisesti ole tarkoitus lisätä verkontietoturvaa, vaan parantaa verkon toimivuutta, nopeuttaa vikojen havaitsemista ja niistä toipumista.

Verkonvalvonta on nykypäivänä olennainen osa kaikkia tietoverkkoja. Periaatteessa jokaisessa vähänkään laajemmassa verkossa, jossa on tuotannollisia palveluita, elinehtona on, että verkkoa valvotaan kunnolla. Tämä takaa sen, että verkko pystyy tarjoamaan ne palvelut ja toiminnot, joita varten se on suunniteltu. Jos tässä epäonnistutaan, se johtaa pahimmassa tapauksessa koko yrityksen kaatumiseen tai tuotannollisiin vaikeuksiin, mistä voi aiheutua rahallista menetystä yritykselle.

Verkonvalvonnalla tarkoitetaan yleisesti järjestelmää, joka jatkuvasti valvoo tietoverkkoa ja sen palveluita. Havaitessaan tietoverkossa poikkeamia se ilmoittaa niistä järjestelmänvalvojalle halutulla tavalla. Ilmoitustapoja voi olla erilaisia. Esimerkiksi niitä voivat olla sähköposti, tekstiviesti tai vaikka älypuhelimeen kehitetty oma ohjelmisto.

Verkonvalvontaa voidaan toteuttaa monella eri tapaa, monilla erilaisilla ohjelmilla, yhdellä tai useammalla sovelluksella. Olipa ratkaisu mikä tahansa, kun verkkoa tai varsinkin sen aktiivilaitteita halutaan valvoa, näyttelee SNMP (Simple Network Management Protocol) verkonvalvonnassa erittäin suurta roolia. Kyseinen protokolla on kehitetty juuri tähän tarkoitukseen. Sen avulla pystytään tekemään monipuolisia verkonvalvontaratkaisuja ja sen soveltuvuus verkonvalvontaan ja hallintaan on hyvä.

2.2 Periaatteet

Verkonhallintaa katsottaessa on huomattava, miten laaja kokonaisuus se on. Verkonvalvonta voidaan käsittää monella eri tapaa. Lisäksi sen toteuttamiseen liittyy paljon erilaisia periaatteita ja suosituksia. Tästä johtuen eri organisaatiot ovat antaneet sen toteuttamiseen erilaisia suosituksia. ITU-T on jakanut sen verkonhallintastandardissaan viiteen eri osa-alueeseen seuraavalla tavalla (ITU-T x.700 1992, 3):

- vikojen valvonta
- käytön hallinta
- kokoonpanon hallinta
- suorituskyvyn valvonta
- turvallisuuden hallinta.

2.2.1 Vikojen valvonta

Nykypäivänä tietoverkkojen monimutkaistuessa ja laajetessa on otettava ehdottomasti huomioon myös se, mitä tehdään, jos jokin laite vikaantuu. Vikojen valvonnassa on olennaista se, että vika pystytään paikallistamaan nopeasti ja tiedottamaan sen mahdollisista vaikutuksista käyttäjille. Tämän toteamiseksi tietoverkon järjestelmänvalvojat tarvitsevat tiedon verkon toiminnasta reaaliaikaisena. (Hautaniemi 1994, luku 2.2.)

Seuraavassa ITU-T:n x.700 suosituksen vianhallintaan määrittelemät toiminnot (ITU-T X.700 1992, 4):

- vikalokien tutkiminen ja ylläpito
- vikojen yksilöiminen ja jäljitys
- havaittujen vikojen hyväksyntä ja toimenpiteet
- vikojen korjaaminen.

2.2.2 Käytön hallinta

Käytön hallinnassa keskitytään seuraamaan verkon resurssien käyttöä käyttäjä- tai ryhmätasolla. Tämä on tarpeellista, jotta voidaan varmistua, etteivät verkon käyttäjät tai ryhmät käytä oikeuksiaan väärin ja kuormita verkkoa muiden kustannuksella. Lisäksi on tarpeellista varmistaa, että he käyttävät verkkoa tehokkaasti eivätkä omalla tietämättömyydellään tai huolimattomuudellaan aiheuta ylimääräistä ja tarpeetonta verkkoresurssien käyttöä. Tällä tavalla verkon järjestelmänvalvojien on helpompi suunnitella tarpeita verkon laajennuksille ja resurssien saatavuuden parantamiselle. Lisäksi järjestelmänvalvojat voivat paremmin opastaa käyttäjiä verkon tehokkaampaan käyttöön. (Hautaniemi 1994, luku 2.3.)

Edellytyksenä käytön hallinnalle on, että järjestelmänvalvojat pystyvät määrittelemään mitä tietoa kerätään, mistä sitä kerätään ja kuinka usein se ajetaan raportoitavaan muotoon. Lisäksi on pystyttävä määrittelemään käyttäjä- tai ryhmäkohtaiset kiintiöt verkkoresurssien käytölle sekä toimenpiteet kiintiöiden ylittyessä. (Hautaniemi 1994, luku 2.3.)

2.2.3 Kokoonpanon hallinta

Tämän päivän tietoverkot koostuvat monista erilaisista laitteista, palveluista ja tietojärjestelmistä. Kokoonpanon hallinnan tehtäviin kuuluu ylläpitää, päivittää ja lisätä

laitteiden välisiä riippuvuuksia sekä laitteiden tilaa koskevia tietoja normaalioloissa. Kokoonpanon hallinnan yhtenä tehtävänä on myös kyetä pysäyttämään ja käynnistämään laitteita, tarvittaessa myös automaattisesti sekä ennalta määrätyillä aikaväleillä.

Kokoonpanon hallinnassa järjestelmänvalvojan on pystyttävä määrittelemään eri laitteiden väliset riippuvuudet käyttäjien sekä palveluiden tarpeita vastaaviksi. Kun nämä asiat on kunnolla määritelty ja dokumentoitu, se helpottaa järjestelmänvalvojan työtä, kun verkkoon tulee muutoksia tai sitä tarvitsee uudelleen konfiguroida. Kun verkko ja sen laitteet on dokumentoitu hyvin, on paljon helpompaa lähteä kartoittamaan laitteiden päivitystarpeita ja inventoida yrityksen omaisuutta. Tällä saavutetaan etua erityisesti piilevien vikojen etsinnässä, mikä saattaa johtua laitteiden ohjelmistoversioiden yhteensopimattomuudesta. (Hautaniemi 1994, luku 2.4.)

Seuraavaksi on lueteltu ITU-T:n x.700 suosituksen kokoonpanon hallintaan määrittelemät toiminnot (ITU-T x.700 1992, 4):

- verkossa tapahtuvien merkittävien muutosten tiedottaminen
- konfiguraatioiden muuttaminen
- verkon tämän hetkisen tilatiedon kerääminen
- parametrien asettaminen verkon rutiinitoimintoihin
- hallittavien laitteiden ja laitekokonaisuuksien nimeäminen
- hallinnoitavien laitteiden alustaminen ja sammuttaminen.

2.2.4 Suorituskyvyn hallinta

Suorituskyvyn hallinnassa korostuu tietoverkon resurssien olemassaolon ja niiden käyttöasteen mittaaminen. Suorituskyvyn hallinnassa keskitytään nimenomaan ver-

konlaitteiden ja palveluiden mittaamiseen sekä valvontaan. Tietoverkon suorituskyvyn hallinta voidaan jakaa kahteen eri toimintoon, jotka ovat verkonvalvonta ja -hallinta.

Lisäksi suorituskyvyn hallinnassa keskitytään edellä mainitun tiedon tuomiseen järjestelmänvalvojien tietoon. Tästä on suurta apua esimerkiksi verkon tulevaisuuden suunnittelussa. Toisin sanoen se luo sille pohjan. Esimerkiksi pitkällä aikavälillä seuratusta suorituskyvystä saadaan tarpeellista tietoa, jotta voidaan perustella, suunnitella ja määritellä uusia ratkaisuja sekä laajennuksia verkkoon. Lisäksi verkon suorituskykyä voidaan arvioida käyttämällä seuraavia apukysymyksiä.(Hautaniemi 1994, kohta 2.5.)

- Mikä on verkon kapasiteetin käyttöaste?
- Jakaantuuko liikenne tasaisesti verkossa?
- Mitkä ovat verkon laitteiden vasteajat tai ovatko ne kasvamassa?
- Löytyykö verkon solmukohdista pullonkauloja?

Seuraavaksi on listattu ITU-T:n x.700 suosituksen suorituskyvyn hallintaan määrittelemät toiminnot (ITU-T x.700 1992, 4):

- tilastollisen tiedon keräämien
- ylläpitää ja tallentaa järjestelmän eri tilat
- kyky päätellä järjestelmän suorituskyky normaalissa ja poikkeavissa oloissa.

2.2.5 Turvallisuuden hallinta

Turvallisuuden hallinnalla pyritään estämään luvaton verkon, laitteiden ja tiedon käyttö. Lisäksi tarkoituksena on kerätä tietoa lokeihin verkosta. Tämä mahdollistaa

verkon tietoturvallisuuden kehittämisen jatkossa. Turvallisuuden hallinnalla on pystyttävä luomaan käytänteet verkon resurssien ja tiedon turvaamiseen. Tämän kaiken toteuttamiseksi turvallisuuden hallintaan on nimettävä henkilöt, jotka huolehtivat työntekemisestä.(Hautaniemi 1994, luku 2.6.)

Seuraavassa luetellaan ITU-T:n x.700 suosituksen turvallisuuden hallintaan määrittelemät toiminnot (ITU-T x.700 1992, 4):

- turvallisuusmekanismien ja palveluiden luominen, poistaminen ja hallinnointi
- turvallisuuteen liittyvän tiedon jakaminen
- turvallisuuteen liittyvien tapahtumien raportointi.

2.3 Ohjelmistot

Verkonvalvontaohjelmistot voidaan karkeasti jakaa kahteen eri ryhmään: kaupallisiin ohjelmistoihin ja ei-kaupallisiin ohjelmistoihin. Ei-kaupallisia ohjelmistoja kutsutaan yleensä myös ”vapaanlähdekoodin ohjelmistoiksi”. Näillä ohjelmistoilla on paljon muutakin eroa kuin se, että toinen maksaa ja toinen ei. Tästä esimerkkinä voisi sanoa, että kaupallisesta ohjelmistosta voi saada paremman tuen tuotteelleen, mutta ohjelmiston räätälöinnin tai puuttuvien ominaisuuksien osalta ollaan sidottuja ohjelmiston toimittajaan.

Erilaisia ohjelmistoja on olemassa todella paljon ja niiden toiminta periaatteet vaihtelevat. Monesti myös laitevalmistajat ovat tehneet laitteidensa hallintaan ja valvontaa omat sovelluksensa, mutta niiden toimivuus verkon kokonaisuuden valvontaan on yleensä rajallinen, koska ne on suunniteltu ainoastaan yhden laitevalmistajan lähtökohdista käytettäväksi laitevalmistajan omien laitteiden valvontaan ja hallintaan.

3 SNMP

3.1 Yleistä

Luvussa 3 käsitellään verkonvalvontaan oleellisesti liittyviä asioita ja toiminteita. Näitä ovat mm. SNMP (Simple Network Management Protocol), joka toimii verkonvalvonnassa käytettävänä protokollana, ja MIB (Management Information Base) mikä taas luo edellytykset SNMP-kyselyiden tietorakenteelle. Näiden kahden asian hallinta on hyvin oleellista verkonvalvonnan ja sen onnistumisen kannalta.

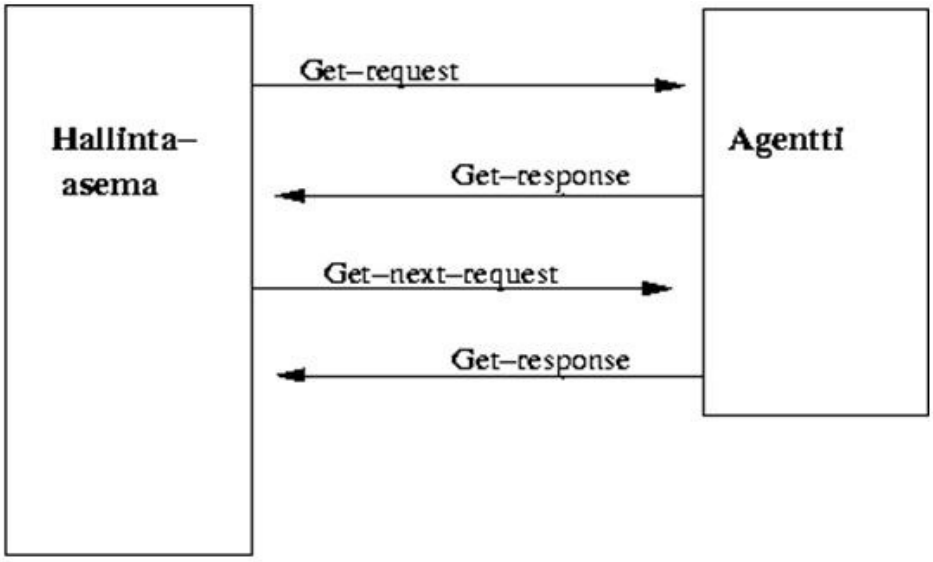
SNMP on yksi yleisemmistä verkonvalvontaan käytettävistä protokollista. Se on ollut yksi avaintekijöistä, joka on mm. mahdollistanut internetin ilmiömäisen kasvun. (SNMP Research International Inc. 2012.) Vaikka protokollan nimi viittaa yksinkertaiseen, SNMP kokonaisuutena on kaikkea muuta kuin yksinkertainen. Itse protokollan kehys ei kuitenkaan ole kovin monimutkainen. Protokolla on kehitetty eri valmistajien laitteiden, reitittimen, kytkimien, tietokoneiden, siltojen, UPS (Uninterruptible Power Supply) ja muiden tietoverkon aktiivilaitteiden valvontaan ja hallintaan.

SNMP on suunniteltu osaksi TCP/IP:n (Transmission Control Protocol / Internet Protocol) perhettä toimimaan UDP:n (User Datagram Protocol) päällä. Tämä tarkoittaa taas sitä, että SNMP on yhteydetön protokolla. Tästä johtuen hallinta-aseman ja agentin välille ei luoda erillistä yhteyttä, vaan jokainen sanoma on erillinen tapahtuma. Koska kyseessä on UDP, se aiheuttaa sen, että SNMP:n on itse huolehdittava viestien perille menosta. Tämän varmistamiseksi ne lähetetään yleensä ajastettuna tietyin väliajoin. (Kaario 2002, 269–270.)

3.2 SNMP:n toiminta ja viestityypit

SNMP:n toiminta perustuu neljään eri osaan jotka ovat: hallinta-asema, hallinta-agentti, hallintatietokanta (MIB) ja verkonhallinnan yhteyskäytäntö (Hautaniemi 1994). Hallinta-asemassa pyörii tai on asennettuna yleensä jonkin kaupallinen tai vapaanlähdekoodin verkonvalvontasovellus, joka käyttää SNMP-protokollaa tehdessään kyselyitä hallinta-agentilta. Hallinta-agentti on yleensä se valvottava laite, esimerkiksi palvelin, reititin, kytkin, UPS, tulostin tai jokin muu laite tai järjestelmä. Hallinta-agentin täytyy tukea SNMP:tä. SNMP-kyselyn saapuessa hallinta-agentille se vastaa kyselyyn lähettämällä pyydetyn tietueen takaisin. Lisäksi hallinta-agentti voi myös itse lähettää tietoja käyttämällä trap-viestiä.

Kuviossa 1 on esitetty esimerkki hallinta-aseman ja SNMP-agentin toiminnasta.



KUVIO 1. SNMP:n toiminta (Haikonen, Hlinovsky & Paju 2000).

SNMP sisältää seuraavat perusviestityypit:

Tiedonkyselyn (GET-request) viestityyppiä käytetään silloin, kun verkonvalvontasovellus haluaa lukea hallinta-agentilta tietoja. Agentti vastaanottaa viestin ja käsittelee sen mahdollisimman pian. Kovan kuormituksen alla olevilla laitteilla voi käydä niin, että laite joutuu hylkäämään viestin. (Mauro & Schmidt 2001, 35–36.)

Get-next-operaatiolla on mahdollista kysellä kokonainen MIB-ryhmän sisältö kerralla. Toisin sanoen kysyttävän MIB-ryhmän alla olevaa objektia varten generoidaan oma get-next pyyntö ja get-response. Koska MIB alipuut ovat hierarkkisia, jatketaan kyselyä niin kauan, että agentti palauttaa virheen ilmoittaakseen, että kyseisen alipuun pääty on saavutettu. Kuviossa 2 on esimerkki get-next operaatiosta. Linuxissa tämä voidaan suorittaa käyttäen komentoa SNMPWalk. (Mauro & Schmidt 2001, 37–38.)

```
$snmpwalk cisco.ora.com public system
system.sysDescr.0 = "Cisco Internetwork Operating System Software
..IOS (tm) 2500 Software (C2500-I-L), Version 11.2(5), RELEASE
SOFTWARE (fc1)..Copyright (c) 1986-1997 by cisco Systems, Inc...
Compiled Mon 31-Mar-97 19:53 by ckralik"
system.sysObjectID.0 = OID: enterprises.9.1.19
system.sysUpTime.0 = Timeticks: (27210723) 3 days, 3:35:07.23
system.sysContact.0 = ""
system.sysName.0 = "cisco.ora.com"
system.sysLocation.0 = ""
system.sysServices.0 = 6
```

KUVIO 2. SNMPWalk-komento (Mauro & Schmidt 2001, 37).

Kuten kuviosta käy ilmi, annettu komento palauttaa seitsemän eri MIB- muuttujaa. Jokainen näistä on osa system-ryhmää, jota alun perin kysyttiinkin. Tämä on erittäin hyödyllistä esimerkiksi silloin, kun halutaan tutkia, mitä kaikkia OID:ta laite sisältää.

Get-bulk (SNMPv2 ja SNMPv3) operaatio on tullut lisäksi SNMPv2:een. Tämä operaatio mahdollistaa hallintatyöaseman noutaa isoja osia taulukoista yhdellä kertaa. Normaalilla get-operaatiolla voidaan yrittää noutaa enemmän kuin yksi tietue kerralla, mutta viestien koot ovat yleensä rajoittuneet agentin määräämiksi. Jos agentti ei pysty palauttamaan kaikkea pyydettyä tietoa, se lähettää virheilmoituksen ilman minkäänlaista dataa. Get-bulk operaatio kertoo SNMP-agentille, että lähetä tietoa niin paljon kerralla kuin pystyt. Tästä johtuen myös ei-täydelliset vastaukset ovat mahdollisia. Get-bulk-operaatio vaattii kaksi erillistä kenttää, jotka pitää täyttää. Ne ovat "nonrepeaters" ja "max-repetitions". Näillä määritellään ensimmäiset noudettavat objektit ja mihin saakka kyselyssä mennään eli seuraavat kysyttävät objektit.

Kuviosta 3 näkyy kuinka get-bulk operaatiolla voidaan kysyä kerralla useampi objekti. (Mauro & Schmidt 2001, 38–39.)

```
$ snmpbulkget -v2c -B 1 3 linux.ora.com public sysDescr ifInOctets ifOutOctets
system.sysDescr.0 = "Linux linux 2.2.5-15 #3 Thu May 27 19:33:18 EDT 1999 i686"
interfaces.ifTable.ifEntry.ifInOctets.1 = 70840
interfaces.ifTable.ifEntry.ifOutOctets.1 = 70840
interfaces.ifTable.ifEntry.ifInOctets.2 = 143548020
interfaces.ifTable.ifEntry.ifOutOctets.2 = 111725152
interfaces.ifTable.ifEntry.ifInOctets.3 = 0
interfaces.ifTable.ifEntry.ifOutOctets.3 = 0
```

KUVIO 3. SNMP Get-bulk esimerkki (Mauro & Schmidt 2001, 39).

Tiedonkirjoittamista (Set) käytetään silloin, kun halutaan asettaa hallinta-agentille tiettyjä arvoja. Set-operaation toiminta on hyvin samankaltainen kuin muillakin operaatiolla, mutta sillä pystytään oikeasti muuttamaan esimerkiksi reitittimen asetuksia. Tämä tekee siitä hyvin vaarallisen komennon käytettäväksi. Tästä johtuen set-operaatiota käytettäessä on aina huolehdittava riittävästä tietoturvasta. Kuviosta 4 nähdään, miten esimerkiksi ciscon-reitittimille voidaan asettaa sen paikkatieto. (Mauro & Schmidt 2001, 38–39.)

```
$ snmpget cisco.ora.com public system.sysLocation.0
system.sysLocation.0 = ""
$ snmpset cisco.ora.com private system.sysLocation.0 s "Atlanta, GA"
system.sysLocation.0 = "Atlanta, GA"
$ snmpget cisco.ora.com public system.sysLocation.0
system.sysLocation.0 = "Atlanta, GA"
```

KUVIO 4. Esimerkki SET-operaatiosta (Mauro & Schmidt 2001, 40).

Get-response-viestillä SNMP-agentti lähettää hallinta-asemalle sen pyytämän tiedon. Viesti sisältää MIB-kohteiden arvot, jotka hallinta-asema on agentilta pyytänyt, sekä mahdolliset vikakoodit liittyen kyselyyn. (Mauro & Schmidt 2001, 42–45.)

SNMP-Trap mahdollistaa SNMP-agentin lähettää omatoimisesti tietoja hallinta-asemalle. Agentti lähettää Trap-viestinsä hallinta-aseman osoitteeseen, mikä sille on konfiguroitu. Hallinta-asema ei vastaa näihin viesteihin ollenkaan, joten agentin on mahdotonta päätellä, onko kyseinen viesti mennyt perille. Alla on muutama esimerkki tilanteista, joissa agentti voisi käyttää trap-viestiä. (Mauro & Schmidt 2001, 42–45.)

- joku Agentin (esim. reititin) interfacesta menee alas
- joku Agentin (esim. reititin) interfacesta nousee ylös
- reitittimen tai kytkimen tuuletin rikkoutuu.

Inform (SNMPv2 ja SNMPv3) -operaatiolla tarkoitetaan mekanismia, millä hallinta-asemat pystyvät keskustelemaan keskenään ja vaihtamaan tietoja. Inform-viestin yksi uusista ominaisuuksista on se, että vastaanottaja vastaa siihen ilmoittaakseen viestin vastaanotetuksi. Inform-viestiä voidaan myös käyttää lähettämään SNMPv2 trap-viestejä agentilta hallinta-asemalle, jolloin hallinta-asema vastaa viestiin ja informoi agenttia, että on vastaan ottanut viestin. (Mauro & Schmidt 2001, 46.)

Report (SNMPv2 ja SNMPv3) - viesti määriteltiin varalle SNMPv2: n, mutta sitä ei koskaan otettu siinä käyttöön. Se otettiin mukaan SNMPv3 määrittelyihin ja sen tarkoituksena on mahdollistaa SNMP-tilakoneiden kommunikoida keskenään. (Mauro & Scmidt 2001, 46.)

SNMP ja sen toiminta määritellään monissa eri dokumenteissa. IETF (Internet Engineering Task Force) on määritellyt oman RFC:n (Requets For Comments) kullekin SNMP:n versiolle erikseen. Näitä on aikamoinen lista, joten ei ole tarkoituksen mukaista alkaa luetella niitä erikseen. (SNMP 2012.) Lisäksi SNMP:n kuuluu MIB (Management Information Base) ja SMI (Structure of Management Information) hallintatietokannat mitkä ovat hyvin olennainen osa SNMP:tä. (Kaario 2002, 270.)

3.3 Tietoturva

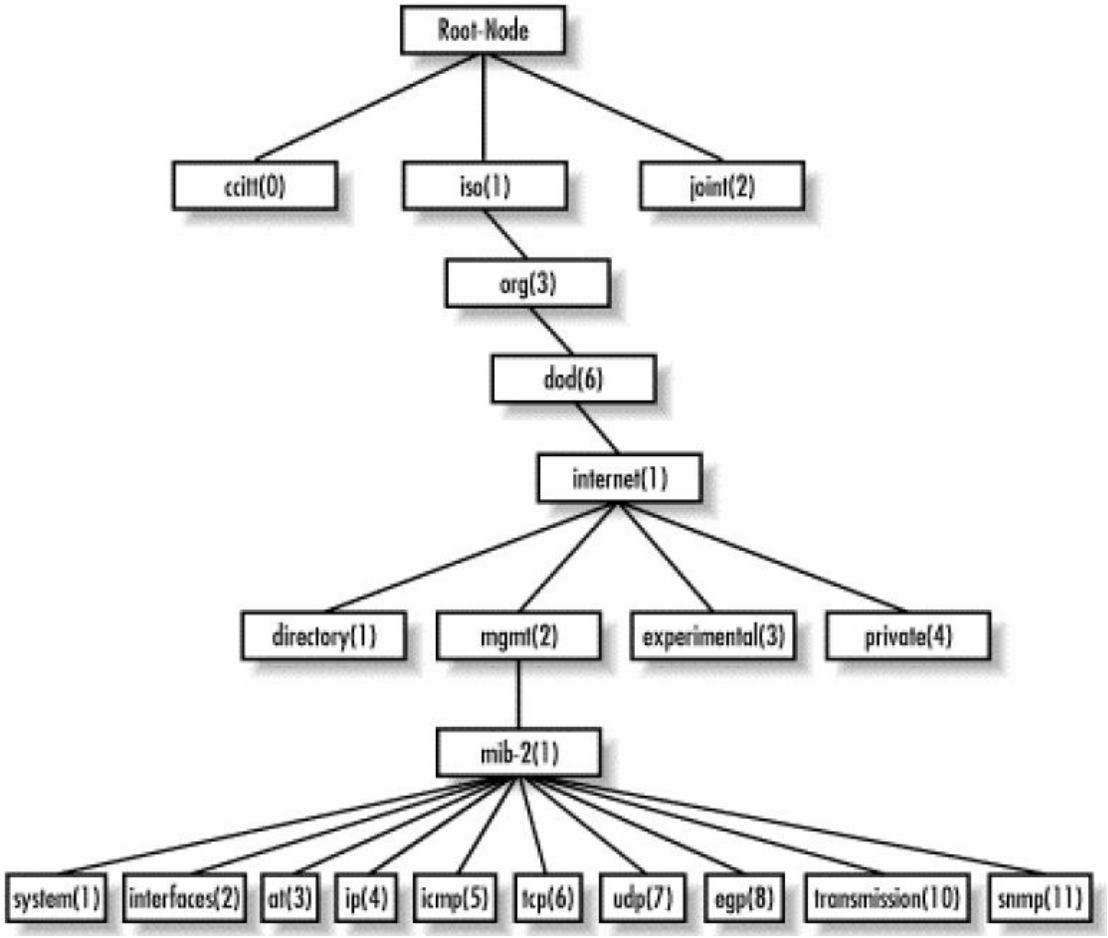
Yleisesti ottaen SNMP on tietoturvan kannalta haastava ja vaarallinen protokolla, jos sitä käytettään huolimattomasti ja ei-asianmukaisesti toteutettuna. Ensimmäisessä SNMP:n versiossa tietoturva ominaisuudet on pyritty hoitamaan community string:llä. Käytännössä tämä tarkoittaa sitä, että SNMP-kehyksessä on oma paikka ”salasanalle”. Tämä salasana kulkee protokollankehyksessä kuitenkin selväkielisenä, joten kuka tahansa, joka pääsee verkkoon fyysisesti käsiksi, pystyy sen näkemään. (Mauro & Schmidt 2001, 279.) Tämän takia SNMPv1:stä käytettäessä ei ole suositeltavaa käyttää SET-viestejä ollenkaan, vaan enemmänkin keskittymään verkonvalvontaan tekemällä GET-Request viestejä laitteille.

Uusien versioiden myötä tietoturva on parantunut SNMP:n osalta huomattavasti, mutta kuitenkin ainoastaan SNMPv3:sta voidaan nykypäivänä suositella käytettäväksi verkonvalvonnassa. Kolmas versio SNMP:stä on tuonut ainoastaan turvallisuuteen liittyviä parannuksia ja sen lisäksi se sisältää kaikki versioiden yksi ja kaksi ominaisuudet. (Mauro & Schmidt 2001, 279.)

Kuitenkin SNMPv3:ssa on olemassa myös haavoittuvuuksia, joten sitäkään ei voida sanoa täysin turvalliseksi. Tästä johtuen verkonvalvonnassa on syytä käyttää myös muita toimenpiteitä tietoturvan parantamiseksi. Tästä esimerkkinä US-CERT:n (United States Computer Emergency Readiness Team) artikkeli, jossa kerrotaan SNMPv3:n haavoittuvuudesta liittyen protokollan käyttämään autentikointiin HMAC (Hash Message Authentication Code). Jotkut SNMPv3:n toteutukset voivat sallia käyttää autentikoinnissa lyhennettyä HMAC koodia, joka on vain yhden tavun pituinen. Sallittavan koodin lyhyydestä johtuen se on altis (brute-force) hyökkäykselle. (US-CERT 2008.)

3.4 MIB

MIB-II (Management Information Base) on erittäin tärkeä hallintaryhmä, koska kaikkien laitteiden, jotka tukevat SNMP:tä on myös tuettava MIB-II (Mauro & Schmidt 2001, 33). Verkonvalvonnassa käytettävät OID (Object IDentifier) löytyvät juurikin MIB-II:n alta, joten niitä on syytä käydä hieman tarkemmin läpi. Kuviossa 5 on MIB-II alipuu kuvattuna.



KUVIO 5. MIB II Puu

Kuviossa 5 huomion arvoisena on OID 1.3.6.1.4 (private(4)) ja sen alla olevat tietueet. Tämän kohdan alle on määritelty kaikki valmistajakohtaiset tietueet ja niitä joudutaan käyttämään verkonvalvonnassa hyvin usein, kuten esimerkiksi tässä työssä reitittimen ja UPS:n valvonnassa. (ks. luvut 4.3.1 ja 4.3.4.)

Taulukossa 1 käydään läpi MIB-II:n alla olevien ryhmien kuvaukset ja OID-osoitteet.

TAULUKKO 1. MIB II

Alipuu	OID	Kuvaus
system	1.3.6.1.2.1.1	Sisältää listan kohteista, jotka liittyvät järjestelmään kuten esim. Ylhäällä oloaika, nimi ja järjestelmän yhteystieto.
Interfaces	1.3.6.1.2.1.2	Tämä ryhmä pitää kirjaa jokaisen interfacen tilasta ja monitoroi mitkä ovat ylhäällä tai alhaalla. Sen lisäksi pitää kirjaa lähetetyn ja vastaanotetun tiedon määrästä sekä mahdollisista pakettien virheistä ja hylkäämisistä.
At	1.3.6.1.2.1.3	AT (address tranlation) ryhmä on mukana ainoastaan alaspäin sopivuuden takia ja se todennäköisesti tulee poistumaan MIB-II:sta
ip	1.3.6.1.2.1.4	Pitää kirjaa kaikesta IP:hen liittyvistä asioista mukaan lukien reititys.
icmp	1.3.6.1.2.1.5	Pitää kirjaa ICMP() virheistä ja hylkäyksistä jne..
tcp	1.3.6.1.2.1.6	Pitää kirjaa mm. TCP-yhteyksistä.
udp	1.3.6.1.2.1.7	Pitää kirjaa sisääntulevien ja ulosmenevien UDP:n yhteyksien tilastoista ja datagrammeista.
egp	1.3.6.1.2.1.8	PItää kirjaa monista EGP:n tilastoista ja pitää yllä taulukkoa EGP:n naapurista.
transmission	1.3.6.1.2.1.10	Tällä hetkellä tässä ryhmässä ei ole määritettynä kohteita, mutta muita mediaan liittyviä MIB:jä on määritelty käyttämällä tätä alipuuta.
snmp	1.3.6.1.2.1.11	Mittaa SNMP:n suorituskykyä ja pitää kirjaa vastaanotetuista ja lähetetyistä SNMP-paketeista.

4 RATKAISUT JA TOTEUTUS

4.1 Yleistä

Luvussa 4 käydään läpi käytännön työn tekeminen sekä tuodaan esille perusteluita ja kriteerejä, mitkä ovat vaikuttaneet opinnäytetyön tekemiseen. Kuitenkin sillä tavalla, että kaikkiin yksityiskohtiin ei paneuduta äärimmäisen tarkasti, vaan kerrotaan ne työn ja raportoinnin kannalta oleellisella tasolla. Tästä esimerkkinä ohjelmistojen asennukset, joita ei ole mielestäni tarkoituksen mukaista käydä läpi kohta kohdalta (asennusohjeet saatavissa internetistä), vaan käydään läpi yleiset periaatteet. Tämä sen takia, koska ohjelmistot ja käyttöjärjestelmät päivittyvät hyvin nopealla tahdilla, ja samalla asennusohjeet muuttuvat.

Opinnäytetyössä ongelman asetteluna oli sopivan avoimeen lähdekoodiin perustuvan verkonvalvontaohjelmiston löytäminen ja konfiguroiminen Ilmasotakoulun käyttöön. Tätä ongelmaa lähdettiin ratkaisemaan ensiksi määrittelemällä kriteerit ja mittarit, eli käytännössä, mitä ominaisuuksia ohjelman piti täyttää (kriteerit) ja mitä sillä haluttiin valvoa (mittarit). Tämän avuksi luotiin taulukko, johon kerättiin vaadittavat ominaisuudet ja niille annettiin kullekin oma painoarvokerroin. Käytännön työn ratkaisemiksi kerättiin kokemuksia myös muista samasta aiheesta tehdyistä opinnäytetöistä. Lopuksi järjestelmästä pidettiin koulutustilaisuus muulle henkilöstölle, missä mm. kerättiin kehitysideoita järjestelmää varten.

4.2 Vaatimukset ja ohjelmiston valinta

Kun ohjelmistoa lähdetään valitsemaan jotain tiettyä tarkoitusta varten, tulisi silloin hyvin tarkasti miettiä ja määritellä, mitä ohjelmistolla tullaan tekemään, mihin sitä tarvitaan ja mitä siltä sen tekemiseen vaaditaan. Ainoastaan tällä tavoin voidaan varmistua siitä, että valitulla ohjelmistolla on edellytykset täyttää siltä vaadittavat toiminnot ja tarpeet. Tässä luvussa käydään lävitse niitä vaatimuksia ja ominaisuuk-

sia, jotka olivat lähtökohtana verkonvalvontaohjelmiston valinnalle, sekä itse ohjelmiston valinta.

4.2.1 Vaatimukset

Vaatimukset jaettiin aluksi kahteen eri osa-alueeseen, jotka olivat ohjelmistolta vaadittavat ominaisuudet ja mittarit. Ominaisuuksilla tarkoitetaan ohjelmiston perusominaisuuksia ja mittareilla taas ohjelmiston kykyä valvoa erilaisia laitteita, palveluita ja arvoja.

Ominaisuuksien määrittelyssä lähdettiin kulkemaan niin sanotusti alhaalta ylöspäin. Apuna käytettiin soveltuvasti OSI (Open Systems Interconnection) -mallia, joka käytiin läpi kerroskerrokselta. Tämä helpotti ohjelmiston ominaisuuksien ja mittareiden määrittelyä ja pystyttiin varmistumaan, että kaikki verkkokerrokset on otettu jollakin tapaa huomioon. Lopputuloksena saatiin lista ominaisuuksista ja mittareista, jotka ohjelman piti täyttää. Alla on lueteltu ohjelmiston valintaan vaikuttavia ominaisuuksia:

- linux yhteensopiva
- ilmainen (Open source)
- hyvä dokumentointi
- SNMP
- hyvä muokattavuus
- yhteisö
- käytettävyys.

Mittarien valitsemiseksi lähdettiin kartoittamaan vanhaa kokemuspohjaa tietoverkosta ja sen ongelmista. Tällä tapaa saatiin tuotua esille ne solmukohdat, jotka ainakin on saatava valvonnan alle. Tässä keskityttiin ITU-T:n X.700 määrittelyjen pohjalta suorituskyvyn – ja vikojenhallintaan.

Taulukossa 2 on lueteltu valvottavat laitteet ja niiden mittarit.

TAULUKKO 2. Laitteet ja mittarit

Valvottava laite	Mittarit
Reititin	CPU:n käyttöaste
	Saatavuus
	Linkkien käyttöaste
Palvelin	Saatavuus
	CPU:n käyttöaste
	DNS
	Muistinmäärä
	Levytila
	palvelut (esim. SQL-server)
Tulostin	Saatavuus
	tarvikkeet (paperi, muste..)
Työasema	Palvelut
	Saatavuus
UPS	lämpötila
	tila
	akun varaus

4.2.2 Ohjelmiston valinta

Ohjelmiston valinta muodostui yhdeksi ydinkysymykseksi tässä opinnäytetyössä. Ehdottomana lähtökohtana toimeksiantajalla oli, että ohjelmiston täytyi olla ns. open source- ohjelmisto eli käytännössä ilmainen.

Erilaisten verkonvalvontaohjelmistojen suuren määrän vuoksi opinnäytetyössä jouduttiin tekemään kompromisseja ohjelmistonvalintaan vaikuttavien kokemusten keräämisessä. Tästä johtuen päädyttiin keräämään kokemuksia myös muista verkonval-

vontaohjelmistoista samanaiheisista jo tehdyistä opinnäytetöistä. Tällä pyrittiin laajentamaan kokemuspohjaa lopullisen valinnan tekemistä varten ja välttämään jokaisen ohjelmiston asentamista ja testaamista erikseen. Kokemuspohjan keräämisessä apuna käytettiin seuraavia opinnäytetöitä: Verkon monitorointi ja tikettijärjestelmä, Miika Sillanpää 2010, Labranet-verkon monitorointi, Jussi Sunnari 2010, Verkonvalvontasovellusten vertailu, Kimmo Vesa 2007, sekä SNMP-Verkonvalvonta vapaan lähdekoodin ohjelmilla, Tuomas Nikka 2007. Seuraavassa on muutamia esimerkkejä aikaisemmin luetelluista opinnäytetoistä, joita on käytetty kokemuspohjan keräämiseen.

Sillanpää (2010, 20) toteaa, että Nagios ei itsessään sisällä mitään monitorointi ohjelmistoa, vaan monitorointi toteutetaan plugineilla, ja että niitä voidaan kehittää mitä tahansa tarkoitusta varten. Tämän takia Nagios on varteen otettava vaihtoehto sellaisiin ympäristöihin, joissa tarvitaan suurta muokattavuutta.

Sunnarin (2010, 38–41) mukaan, Pandora FMS ohjelmisto kuului yhteen varteen otettavista vaihtoehdoista, mutta ohjelmistosta löydettiin asennuksen jälkeen vakavia puutteita liittyen ohjelmiston ominaisuuksiin ja valvottavien kohteiden lisääminen oli tehty hankalaksi. Tämän lisäksi SNMP:tä ei edes ollut saatu toimimaan ollenkaan. Vertailtavista ohjelmistoista Zabbix oli taas paljon parempi asennettavuudeltaan ja ulkoasultaan, mutta hälytysten konfiguroinnissa oli havaittavissa pahoja virheitä.

Kimmo Vesa (2007, 47–48) toteaa yhteenvedossaan Zappix:n olevan Nagiosta parempi käyttöliittymältään ja yksinkertaisuudeltaan mutta ei muokattavuudeltaan.

Nilkan (2007, 34) mukaan, Cacti on monipuolinen ja laaja ohjelma, mutta se ei ole varsinainen NMS (Network Management System) eli siinä on rajoittuneet valvonta mahdollisuudet. Tästä esimerkkinä se, että Cacti ei tue ollenkaan SNMP:n Trap-viestejä.

Potentiaalisista ohjelmistoista tehtiin myös vertailutaulukko, johon kerättiin numeeriset arvot luvussa 4.1.1 asetettujen omaisuuksien mukaan sekä määriteltiin kullekin ominaisuudelle painoarvo sen mukaan, kuinka tärkeäksi ominaisuus nähtiin lopputuloksen kannalta. Vertailutaulukossa (ks.taulukko 3) ominaisuuksien arvostelu toteutettiin asteikolla 1-3 puolen pisteen välein. Painoarvoiksi määriteltiin numerot 1-5:een yhden pisteen välein. Painoarvojen määrittelyssä eniten painoarvoa annettiin muokattavuudelle (5) ja vähiten yhteisölle (3). Muokattavuus oli yksi toimeksiantajan tärkeimmistä kriteereistä.

TAULUKKO 3. Ohjelmistojen vertailutaulukko

Ohjelmisto	kokemukset	muokattavuus	dokumentointi	yhteisö	käytettävyys	yht.
Cacti	2	2	2	2	2	40
Nagios core	2	3	2,5	3	2	50
Opennms	2	2	2	2	2	40
Pandora FMS	1	2	1	1,5	2	32,5
Zappix	2	2	2,5	2,5	3	49,5
Painoarvot	4	5	4	3	4	

Ohjelmistoista tehdyn vertailutaulukon ja kerätyn tietoperustan perusteella verkonvalvontaohjelmistoksi päädyttiin valitsemaan Nagios core.

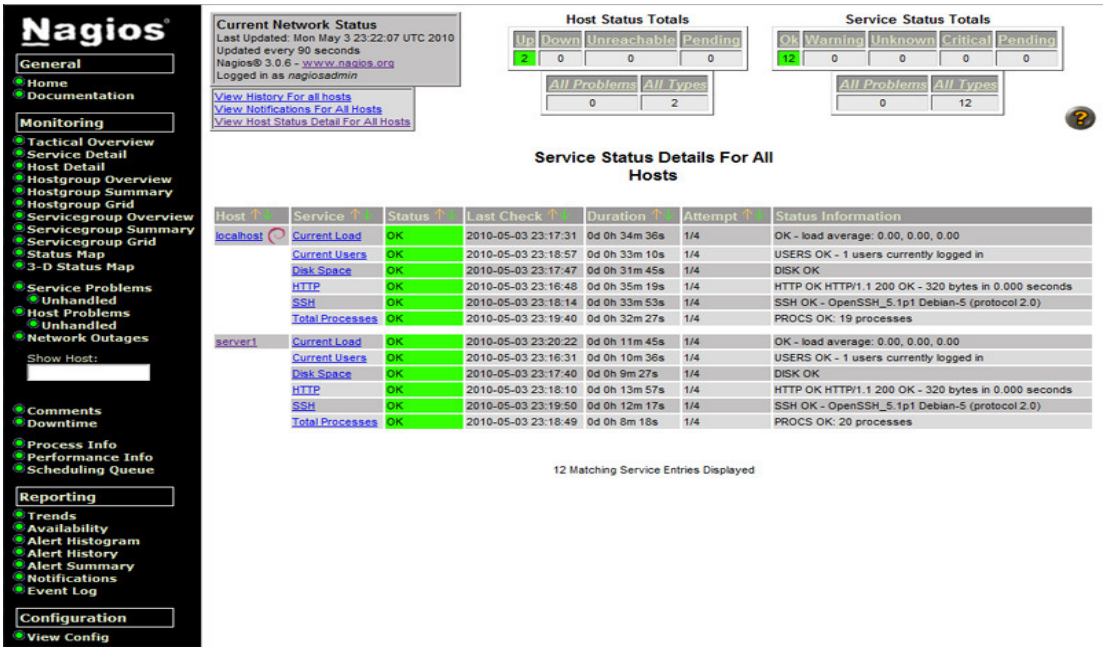
4.3 Nagios verkonvalvontaohjelmistona

4.3.1 Yleistä

Nagios on vapaaseen lähdekoodiin (GNU General Public License) perustuva tehokas verkonvalvontaohjelmisto, jonka avulla yritykset pystyvät tunnistamaan ja ratkaisemaan IT- infrastruktuurisia ongelmia ennen kuin ne vaikuttavat yrityksen kriittisiin

prosesseihin ja aiheuttavat taloudellisia menetyksiä. Nagios on suunniteltu erittäin hyvin skaalautuvaksi sekä joustavaksi verkonvalvontaohjelmistoksi. (Nagios core 2012.) Tämä tekee sen erittäin hyvin soveltuvaksi sellaisiin ympäristöihin, missä järjestelmät ovat kokonaisuutena erityisiä ja vahvasti räätälöityjä. Tämä oli yksi niistä vahvoista ominaisuuksista, joka johti Nagioksen valintaan. Nagioksesta löytyy myös kaupallinen versio, joka tunnetaan nimellä Nagios XI.

Kuviossa 6 on esimerkki Nagioksen käyttöliittymästä Services välilehdeltä.

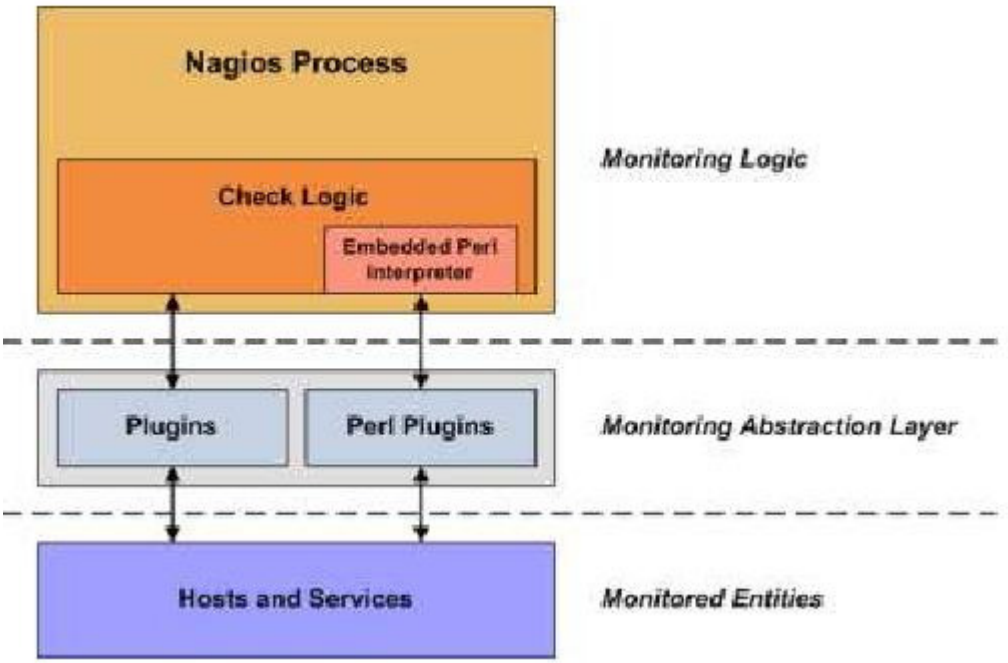


KUVIO 6. Nagios hosts/services

4.3.2 Nagios plugins

Nagios eroaa muista verkonvalvontajärjestelmistä siinä, että se ei sisällä itsessään ollenkaan mekanismeja, millä se voisi valvoa ulkoisia palveluita tai laitteita. Nagios core on pelkästään ydin jonka ympärille verkonvalvonta ja siihen tarvittavat mekanismit rakentuu. Nagiokseen on asennettava plugineja, jotka hoitavat laitteidenvalvonnan ja ilmoittavat havaituista ongelmista tarvittaessa eteenpäin. Pluginit ovat käytännössä pieniä ohjelmia, mitkä hoitavat valvontatiedon keräämisen laitteilta ja palveluilta. Ne ovat yleensä käännetty suoritettaviksi ohjelmiksi tai skripteiksi ja niitä voidaan myös tarvittaessa ajaa suoraan komentokehotteessa.

Kuviossa 7 on esitetty periaatekuva pluginien toiminnasta. Siitä näkyy, että pluginsien molemmanpuolin on oma rajapintansa minkä kautta ne keskustelevat sekä Nagioksen ja valottavien laitteiden kanssa. Alemmassa rajapinnassa on käytössä hyvin usein SNMP, kun taas ylempi rajapinta on Nagioksen sisäinen rajapinta ja se on määritelty ohjelmiston tekijän toimesta.



KUVIO 7. Nagios plugins (Nagios Enterprises 2012, 141.)

Se, mikä tekee Nagioksesta erittäin muokattavan verkonvalvontaohjelmiston, on juuri tämäntyyppinen plugin arkkitehtuuri, koska se mahdollistaa melkein minkä tahansa asian valvonnan. Valmiita pluginejä löytyy todella paljon esimerkiksi osoitteista: http://nagiosplugins.org/ ja http://exchange.nagios.org/directory/Plugins. Mikäli näistä ei löydy sopivaa pluginiä, on niitä mahdollista kirjoittaa myös itse.

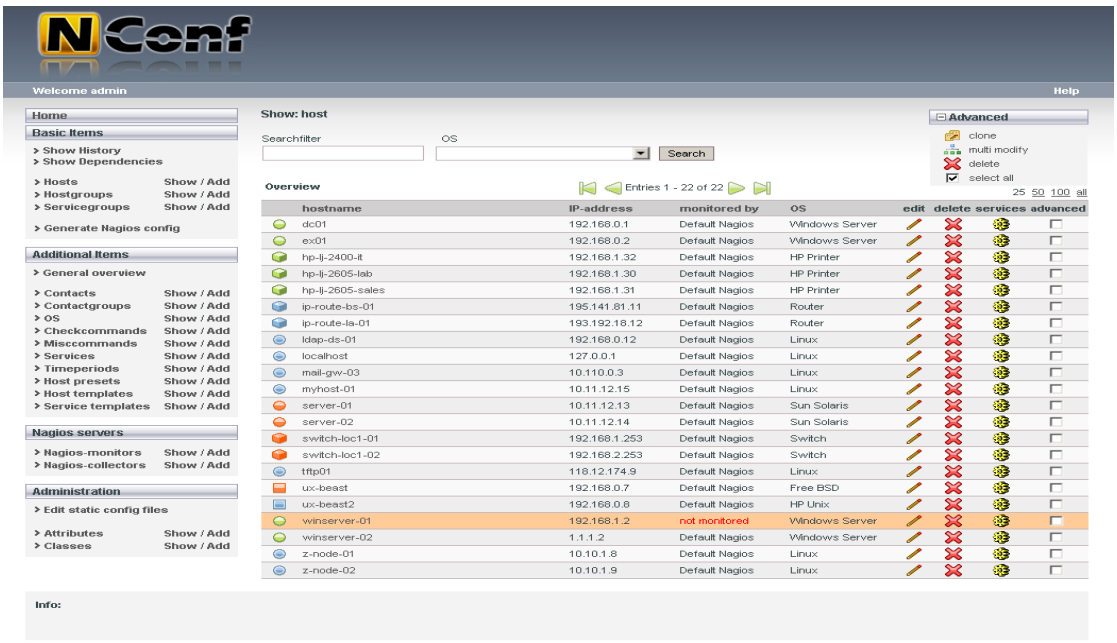
4.3.3 Nconf

Nconf on PHP pohjautuva web-työkalu Nagioksen hallintaan. Kyseisellä ohjelmistolla pystytään käytännössä hallitsemaan kaikkia Nagioksen asetuksia ja se on erityisesti

suunnattu isoihin ympäristöihin, mutta toimii luonnollisesti myös pienemmissä ympäristöissä. Se poikkeaa muista samanlaisista työkaluista tarjoamalla laajempitasoisia toimintoja kuten malleja, riippuvuuksia ja kyvyn konfiguroida jaettua Nagios-topologiaa. Jaetulla Nagios-topologialla tarkoitetaan sitä, että verkossa on useampia Nagios palvelimia, jotka suorittavat valvontaa. Nconf on pääasiassa suunnattu järjestelmänvalvojille, joilla on jo aikaisempaa kokemusta Nagioksesta, mutta etsivät käytännöllisempää tapaa hallita sitä. (Nconf 2012.)

Nconf:n kehitys alkoi vuonna 2006. Projektin tavoitteena oli luoda graafinen työkalu järjestelmänvalvojille, jolla pystyisi automaattisesti hallitsemaan jaettua Nagios palvelinten topologiaa. Nconf perustuu PHP, Perl ja MySQL kieliin. Web-käyttöliittymä on toteutettu PHP:lla. Ohjelman kaikki näkymät ja muodot ovat luotu mahdollisimman dynaamisiksi, jotta muutoksien tarve PHP-koodiin olisi mahdollisimman vähäinen. Nconf:n suunnittelussa on yritetty keskittyä helppoon -ja mahdollisimman hyvään käytettävyyteen. (Nconf 2012.)

Kuviosta 8 näkyy esimerkki Nconf:n web-käyttöliittymästä, mistä näkyy hallittavat laitteet, kytkimet, palvelimet, reitittimet ja tulostimet (Nconf 2012).



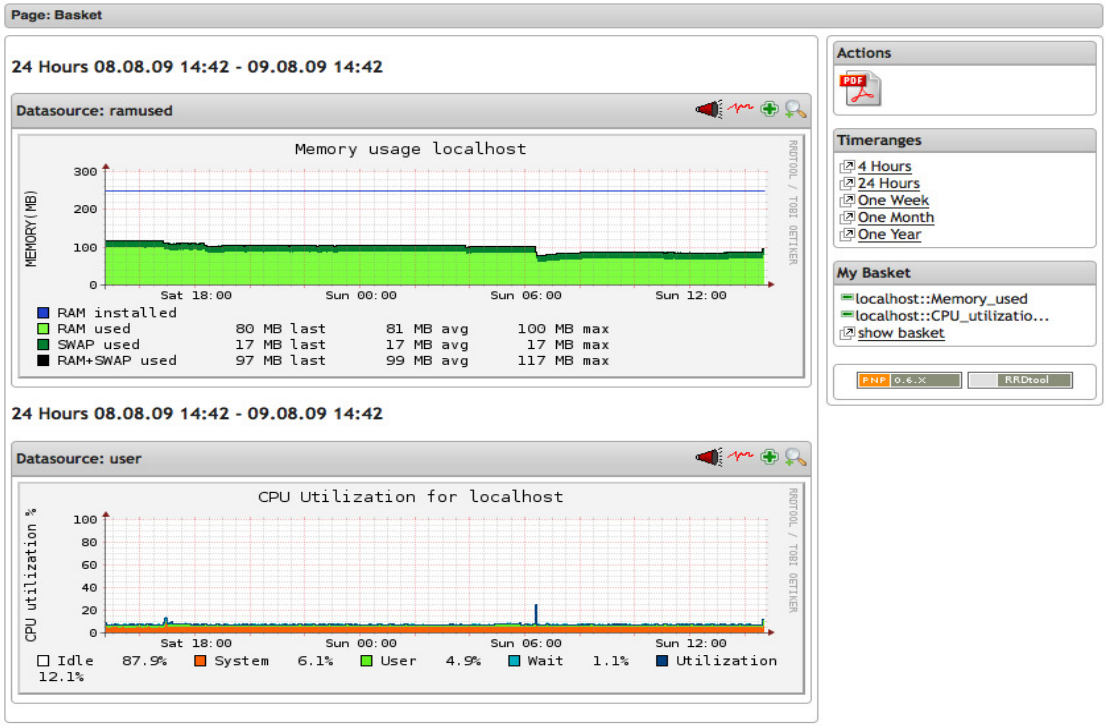
KUVIO 8. Nconf

Tämä sovellus (NConf) helpottaa Nagioksen hallinnointia erittäin paljon. Sillä on huomattavan nopeaa lisätä uusia valvottavia laitteita tai palveluita. Kaikki hallinta voidaan suorittaa selainpohjaisesti etänä, eikä Nagioksen tekstitiedostoihin tarvitse koskea.

Tehtäessä muutoksia Nconf:n avulla Nagiokseen Nconf tarkistaa lisättyjen parametrien syntaksin ja ilmoittaa, jos se löytää virheitä. Tämän lisäksi se ottaa jokaisesta konfiguraatiosta ns. "snap shotin" ja tallentaa sen. Tämä on hyvä ominaisuus silloin, kun jotain menee pieleen, niin aina voidaan palata askel taaksepäin.

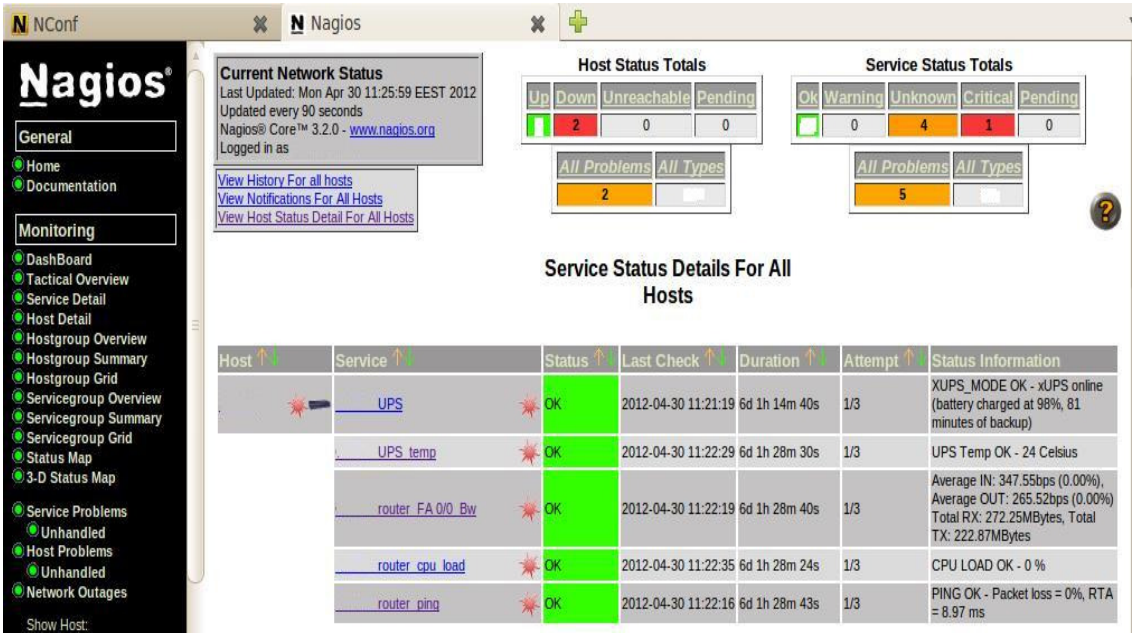
4.3.4 PNP4Nagios

PNP4Nagios on lisäosa Nagiokseen, millä pystytään analysoimaan pluginien tarjoamaa dataa ja tallentamaan se RRD-tietokantaan (Round Robin Databases). Se tarjoaa myös käyttöliittymän millä päästään katselemaan tallennettuja tietoja kuvaajista. Kuviossa 9 on esitetty esimerkki PNP4Nagioksen kuvaajasta. (PNP4nagios 2012.)



KUVIO 9. Esimerkki PNP4Nagioksen kuvaajasta

PNP4Nagioksen asentamisen jälkeen valvottaville palveluille on tehtävä joko oma ”template” tai lisättävä jo olemassa olevaan ”templateen” ”action URL”, jotta kuvaajat saadaan näkyviin. Tämä osoittaa palvelimen sisäisen osoitteen palveluiden ja laitteiden kuvaajille. Kuviossa 10 on Nagioksen näkymä service- välilehdeltä kun PNP4Nagios on konfiguroitu palveluille.



KUVIO 10. Nagios ja PNP4Nagios

Kuviossa 10 on palveluiden oikealle puolelle ilmestynyt punainentähti, tätä klikkaamalla aukeaa kyseisen palvelunkuvaaja selaimen välilehdelle.

4.4 Valvonnan konfigurointi

Luvussa 4.4 käsitellään palveluiden ja laitteiden konfiguroimista Nagioksen valvottavaksi, keskittyen lähinnä Nagioksen komentojen luomiseen Nconf:n avulla ja päätelaitteiden konfiguroimiseen.

Aktiivilaitteilta tehtävissä kyselyissä käytettiin avuksi SNMP-protokollaa ja sen versiota 3. Tässä vaiheessa kuitenkin huomattiin, että kaikki vanhemmat reitittimet eivät tukeneet täydellisesti SNMP versiota 3, vaan ainoastaan sen autentikointi toimintoa.

Tämä johtui pääsääntöisesti cisco-tuotteiden IOS- versiosta. Tästä johtuen niiden osalta jouduttiin tyytymään käyttämään SNMP-protokolaa ilman salausta. Myös jotkut pluginit eivät myöskään tukeneet täysin SNMPv3:sta vielä. Windows koneiden osalta apuna käytettiin NSClientiä, jolla mahdollistettiin pääsy palvelimien tarvittaviin tietoihin kuten prosessorin käyttöaste, muisti ja levytila.

4.4.1 Reititin

Reitittimien valvontaan käytettiin kolmea eri mittaria, saatavuus, CPU:n käyttöaste ja linkkien liikennemäärän valvonta. Näillä kolmella mittarilla voidaan valvoa reitittimen käyttöasteen riittävyyttä, ja nähdään, jos se rupeaa muodostumaan pullonkaulaksi muulle verkolle.

Reitittimen saatavuus hoidettiin Nagioksen omalla siihen tarkoitetulla plugin:llä (check_ping). Tämä komento suorittaa kaikessa yksinkertaisuudessaan ainoastaan ping- kyselyn valitulle reitittimelle. Tämän pluginin tiedoista saadaan reitittimen vasteaika ja data-pakettien hukkaantumisen määrä (paket-loss), sekä niille voidaan asettaa halutut hälytysarvot (warning ja critical).

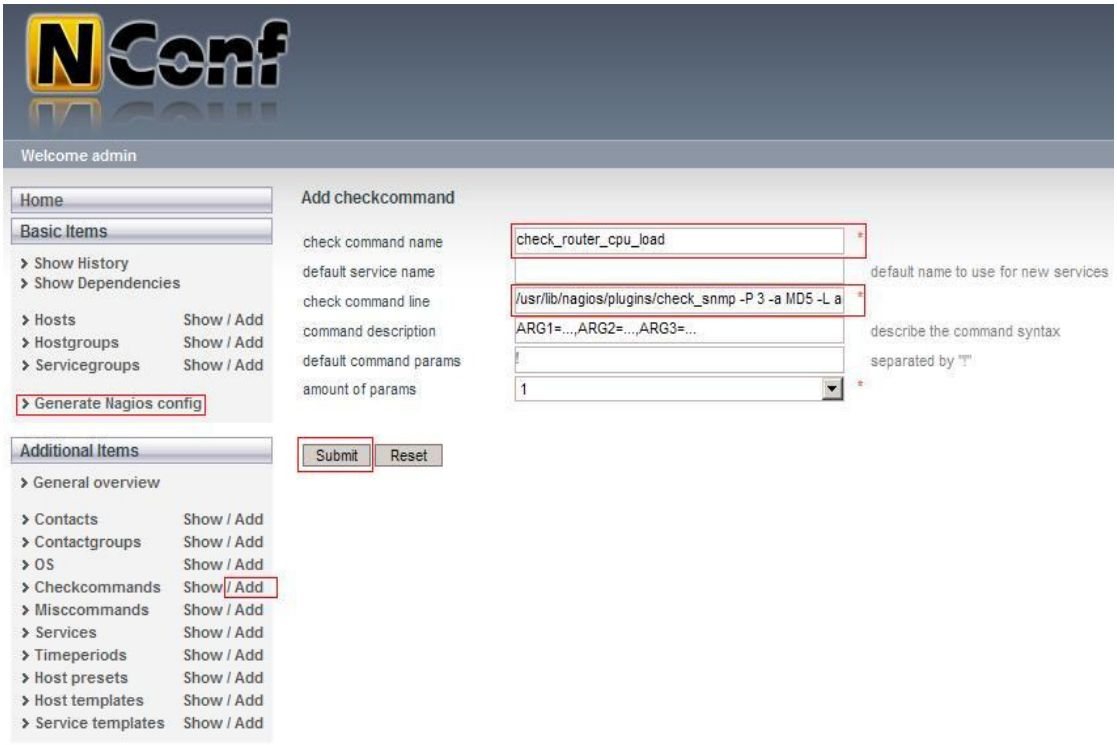
Reitittimille pitää myös konfiguroida SNMP, jotta Nagioksen SNMP-kyselyt onnistuisivat. Konfigurointi eroaa jonkin verran riippuen siitä käytetäänkö SNMP:n versiota 1 vai versiota 3. Käytettäessä versiota yksi riittää ”community stringin” määrittäminen ja lisäksi sille annetaan myös luku- tai kirjoitusoikeus. Kirjoitusoikeuden antamista ei suositella ollenkaan käytettäessä SNMP-versiota 1.

Käytettäessä SNMP-versio 3:sta joudutaan reitittimelle määrittämään myös ryhmä ja käyttäjä. CISCO:n reitittimen konfigurointi tapahtui komentorivin kautta. Apuna siinä käytettiin CISCO:n tekemää ohjetta. Käytännössä reitittimelle täytyy luoda SNMPv3-käyttäjä ja ryhmä, sekä niille on määritettävä niiden ominaisuudet. Käyttäjälle konfiguroidaan SNMP-versio, käyttäjännimi, autentikointi-protokolla, salausprotokolla, salausprotokollan salasana, ryhmä ja pääsylista. Ryhmälle taas konfiguroidaan SNMP-versio, ryhmännimi, käytetäänkö salausta/autentikointia vai molempia ja annetaan

joko luku/kirjoitusoikeudet tai molemmat sekä asetetaan tarvittaessa pääsylista. (Cisco Systems 2012.)

CPU:n käyttöasteen konfiguroinnissa lähdettiin ensimmäiseksi luomaan Check- komentoa kyseistä kohdetta varten. Ensimmäiseksi piti selvittää, mikä OID vastaa kyseistä tietoa (CPU:n käyttöastetta). OID:n löytämiseksi CISCO:lta löytyy tähän tarkoitukseen tehty työkalu (SNMP Object Navigator). Tämän työkalu avulla voidaan etsiä ja selata ciscon laitteiden OID:ta ja se myös kertoo kuvaukset löydetyille OID:lle (CISCO Support 2012).

Kuviosta 11 nähdään miten Nconf:lla luodaan uusi tarkistuskomento.



KUVIO 11. Uusi tarkistuskomento

Nagios käyttää CPU:n käyttöasteen selvittämiseen Check_snmp Pluginiä ja kyseinen komento syötetään ”check command line” riville. SNMP:stä käytettiin versiota 3. Komento oli kokonaisuutena seuraavanlainen: ” */usr/lib/nagios/plugins/check_snmp -P 3 -a MD5 -L authNoPriv -H '$HOSTADDRESS$' -U admin -A '$ARG1$' -o .1.3.6.1.4.1.9.9.109.1.1.1.1.8.1 -w 80 -c 90 -u '%' -l CPU LOAD* ”. Tämän komennon

parametrit on käyty tarkemmin läpi liitteessä 1. Kun tarvittavat tiedot oli syötetty, niin sen jälkeen painetaan vain kohdista ”Submit” ja ”Generate Nagios config”. Tämän jälkeen Nconf automaattisesti syöttää tiedot Nagioksen konfiguraatio tiedostoihin ja tarkastaa, että niissä ei ole syntax-virheitä.

Linkkien kaistanvalvonta ei ollutkaan ihan niin yksinkertaisesti toteutettavissa kuin olisi luullut. Ongelmaksi muodostui se, että reitittimistä ei löytynyt suoraan kaistan käyttöä vastaavaa OID:tä, vaan ainoastaan laskuri, mikä laskee läpimenevän datan määrän oktetteina. Kuitenkaan reititin ei laske datan määrää aika yksikköä kohden, vaan datan kokonaismääränä. Toisin sanoen laskuri jatkaa kasvamistaan jatkuvasti, kunnes se tulee täyteen ja nollaantuu. Näitä laskureita on kaksi yhtä linkkiä kohti: yksi sisään menevälle datalle (IfInOctets 1.3.6.1.2.1.31.1.1.1.6) ja yksi ulosmenevälle datalle (IfOutOctets 1.3.6.1.2.1.31.1.1.1.10). Tästä johtuen sitä varten jouduttiin etsimään plugin, joka hoitaa laskentatyön, jotta mahdollinen läpimenevä data saadaan mitattua ja yksiköksi saataisiin datanmäärä aikayksikköä kohden esimerkiksi (Mb/s).

Osoitteesta http://exchange.nagios.org/directory/Plugins/ löytyi plugin (check_iftraffic42.pl) tätä tarkoitusta varten. Kyseinen plugin on kirjoitettu perl:llä ja se käyttää SNMP:tä linkin data määrän selvittämiseksi. Ulostulona se antaa sisään ja ulosmenevän datanmäärän muodossa mb/s, linkin käyttökapasiteetin prosentteina ja siirretyn kokonaisdatan määrän. Check_iftraffic_42.pl plugin mahdollistaa myös edellä mainitun tiedon näyttämisen kuvaajana PNP4Nagioksen kautta.

Itse plugin asentaminen onnistui pelkästään kopioimalla se kansioon missä Nagioksen pluginit sijaitsivat, mutta se ei suostunut käynnistymään puuttuvien CPAN-moduulien (Comprehensive Perl Archive Network) vuoksi. Nagiosta asennettaessa tätä ei pystytty ottamaan huomioon ja kun tässä vaiheessa palvelin, jonka päällä Nagios pyöri, oli jo eristetty Internetistä, niin puuttuvien moduulien asentaminen piti tehdä käsin. Tämä oli huomattavan työlästä ja aikaa vievää, verrattuna siihen, että sen olisi voinut tehdä automaattisesti suoraan internetistä.

CPAN:n ideana on helpottaa perl:llä tapahtuvaa ohjelmointi Linux-ympäristöissä eli kaikkea ei tarvitse kirjoittaa itse, vaan voidaan käyttää valmiita moduuleja erilaisten toimintojen suorittamiseen (CPAN 2012).

Kuviossa 12 on esimerkki kuvaajasta, joka näyttää sisään -ja ulosmenevän liikenteen määrän.

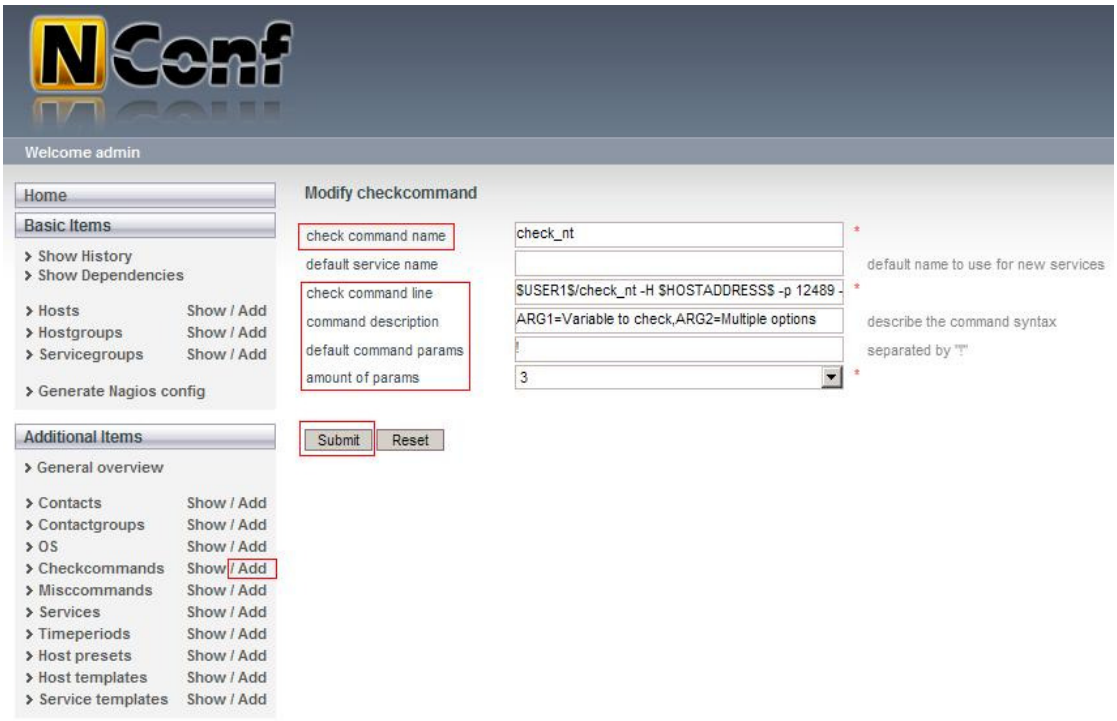
KUVIO 12. Check_iftraffic42.pl

Kuviosta 12 voi nähdä, kuinka sisään menevä liikenne on ollut huomattavasti suurempaa kuin ulostulevan. PNP4nagioksessa on myös ominaisuus, mikä suhteuttaa kuvaajan pystyakselia siinä olevaan datan määrän mukaan, tästä johtuen vasemman reunan nopeusyksiköt ovat erilaiset.

4.4.2 Palvelin

Valvottaessa windows palvelimia Nagioksella, valvontaan on käytettävä apuohjelmaa, jotta tarvittavat tiedot saadaan sieltä ulos eli Nagioksella ei voida suoraan valvoa windows koneita. Tämän suorittamiseen on olemassa useitakin vaihtoehtoja, mutta tässä tapauksessa käytettiin ohjelmaa nimeltä NSClient.

NSClient on yksinkertainen ja tehokas työkalu, millä voidaan valvoa Windows palvelimia. Se on rakennettu Nagiosta varten, mutta sitä voidaan myös käyttää muissa verkonvalvonta sovelluksissa (NSClient 2012). Palvelimien perustietojen saamiseksi NSClient:lla käytetään Nagioksessa Check_nt pluginiä. Lisääminen toimii lähes samalla tavalla kuin kaikissa muissakin tarkistuskomentojen lisäyksissä.(Ks. Kuvio 13.)



KUVIO 13. Check_nt komennon lisääminen

Käytännössä Check_nt plugin keskustelee Windows koneelle asennetun NSClientin kanssa. Tällä komennolla voidaan valvoa kaikkia NSClientin tarjoamia tietoja esim. levytila. Riippuen valvottavasta kohteesta komennon parametritietoihin syötetään kohteenmukaiset arvot. Kohteen tietojen syöttäminen kannattaa tehdä käyttämällä argumentteja. Tällä tavalla voidaan käyttää vain yhtä check_nt komentoa kaikkiin windows koneisiin tehtäviin valvontoihin. Käytettävien argumenttien määrä voidaan valita vapaasti. Nämä syötetään lisättäessä tai muutettaessa komentoa. Itse argumentit syötetään taas vastaavasti silloin, kun ollaan lisäämässä laitteelle valvottavaa palvelua.

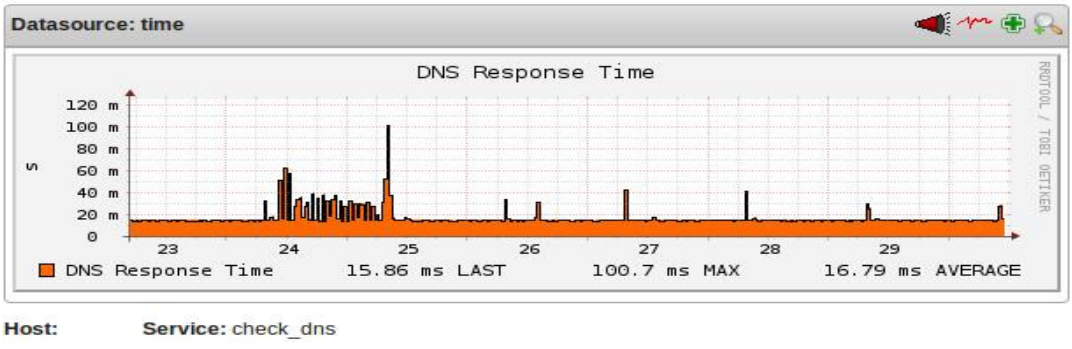
Kuviossa 14 on esitetty ne argumentit, jotka vaaditaan SQL- server palvelun valvomiseksi.

params for check command	
ARG1:	SERVICESTATE
ARG2:	-d SHOWALL
ARG3:	-l "SQL Server (MSSQLSERVER)"

KUVIO 14. Check_nt komennon parametrit

Kaikkien muiden kohteiden kuten levytila, muisti, CPU:n käyttöaste valvominen hoidetaan vain muuttamalla parametrien arvoja. Näiden perusteella NSClient tietää mitä valvottavaa kohdetta kulloinkin tarkoitetaan. Kyseinen komento ja sen toiminnot on esitelty tarkemmin liitteessä 2.

DNS:n (Domain Name System) valvomiseksi käytettiin check_dns nimistä pluginia, joka taas käyttää nslookup-ominaisuutta toiminnon suorittamiseksi. Kyseiselle komennolle syötetään yksinkertaisuudessaan vain kaksi parametria: valvottavan toimialueen nimi ja sen IP-osoite. Nagios mahdollistaa myös DNS-clustereiden valvomisen. Tällä tarkoitetaan esimerkiksi sitä, että valvottavia DNS-palvelimia olisi useita. Jos näitä kaikkia palveluita valvottaisiin erikseen, niin yksittäisen palvelun kaatuminen aiheuttaisi aina hälytyksen. Käytettäessä cluster- toimintoa voidaan esimerkiksi määritellä, että vasta kahden samanlaisen palvelun kaatuminen aiheuttaa hälytyksen. Tässä työssä ei kuitenkaan ollut tarpeellista toteuttaa tämänlaista valvontaa, joten sitä ei käsitellä tarkemmin. Kuviossa 15 on esimerkki DNS:n vasteajoista.



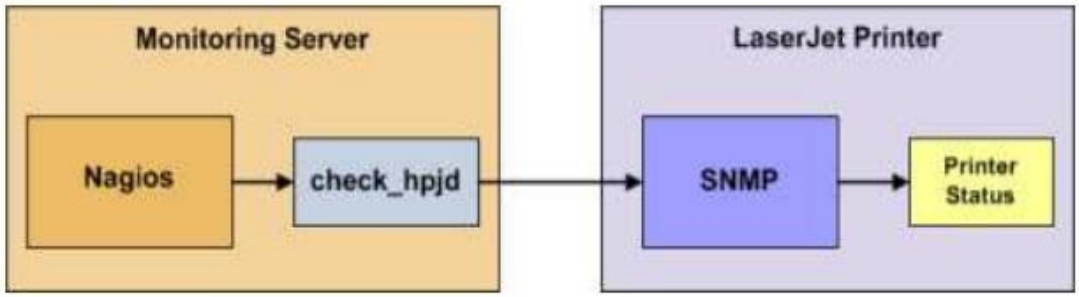
KUVIO 15. Esimerkki DNS:n vasteajoista

Työasemien valvonnassa keskityttiin suurimmalta osalta samoihin asioihin kuin palvelimienvalvonnassa. Suurimmalta osalta niissä valvottiin verkon kannalta tärkeiden palveluiden päälläoloa.

4.4.3 Tulostin

Tulostimien kohdalla asiaa helpottamaan Nagioksesta löytyy check_hpjd plugin, joka keskustelee SNMP:n välityksellä tulostimien kanssa, jossa on JetDirect-kortti asennettuna. Kyseisellä plugin:llä voidaan tulostimista saada mm. seuraavia tietoja: paperi jumissa, paperi loppu, tulostin ei tavoitettavissa (offline), värijauhe vähissä, muisti vähissä, jokin luukuista auki tai tulostimen ulostulo täynnä. Tällä toiminnolla saavutettiin käytännön hyötyjä tulostimien värikasettien saatavuudessa, sillä värijauhe vähissä ilmoitukset ohjattiin suoraan niiden hankinnasta vastaavan sähköpostiin. Tällä pystytään osaltaan varmistamaan, että väriainekasetit eivät pääse loppumaan, vaan niitä voidaan hankkia todellisen kulutuksen mukaan.

Kuviossa 16 on esitetty check_hpjd plugin toimintaperiaate.



KUVIO 16. Hpjd-plugin toimintaperiaate (Nagios Enterprises 2012, 29).

4.4.4 UPS

UPS:n valvonnassa käytettiin check_xups_mode.pl nimistä pluginiä, jolla saatiin selville UPS:n tila, akunvaraus prosentteina ja jäljellä oleva akunvaraus minuutteina.

Lämpötilavalvontaa kyseisestä plugin:stä ei löytynyt, vaan se toteutettiin SNMP-kyselyllä. Osoitteesta http://www.oidview.com/mibs/534/XUPS-MIB.html löytyy XUPS-MIB-moduuli, joka sisältää kaikki OID:t mitä EATON:in valmistamat UPS tukevat. OIDview on sivusto, mistä voi ladata eri valmistajien MIB-kantoja ilmaiseksi, myös tietoverkkojen hallintayhteisöt käyttävät sitä.(OIDview 2012.) Kyseinen OID, mistä lämpötilan arvo löytyi, oli 1.3.6.1.4.1.534.1.6.1.

UPS:n valvonnalla pyrittiin parantamaan järjestelmien valvontaa. Yhtenä puutteena järjestelmissä oli se, että niissä ei tällä hetkellä ollut erillisiä lämpötila antureita, joten UPS:n lämpötila antureista saatiin suuntaa antavaa tietoa laiteräkkien jäähdytyksen toimivuudesta.

4.4.5 Hälytykset

Toimeksiantajan vaatimuksena oli, että hälytykset hoidetaan sähköpostin välityksellä. Ulkoisia hälytyksiä ei ollut mahdollista käyttää. Tällä tarkoitettaan esimerkki SMS-viestin (Short Message Service) lähettämistä.

Nagios jakaa laitteiden ja palveluiden tilat kahteen eri luokkaan (HARD ja SOFT). SOFT- tila ilmenee silloin, kun laite on joko (DOWN) tilassa tai (non-OK) tilassa. Tässä vaiheessa siitä ei vielä lähde hälytystä eteenpäin, vaan se taltioidaan lokeihin. Tämän jälkeen Nagios tarkastaa palvelun tai laitteen tilan uudelleen ennalta määrätyllä aikavälillä. Kun Nagios on tarkastanut palvelun tai laitteen uudelleen n- kertaa, eikä se ole noussut normaalitilaan, vaihtuu palvelu tai laite silloin tilaan (HARD). (Nagios Enterprises 2012, 183–184.)

Nagios noudattaa tiettyä logiikkaa siihen, milloin ilmoitus laitteesta tai palvelusta lähtee eteenpäin. Ilmoitus lähtee silloin eteenpäin, kun valvottavalaite tai palvelu menee (HARD) tilaan tai se pysyy siinä tilassa yli määritetyn ajan, jolloin siitä lähetetään uusi ilmoitus. Jokaisella laitteelle tai palvelulle asetetaan yhteysryhmä, mille se lähettää ilmoituksen. Yhteysryhmä voi sisältää useita eri henkilöitä, myös laite tai palvelu voi sisältää vastaavasti useita yhteysryhmiä.(Nagios Enterprises 2012, 194.)

Ennen kuin ilmoitus lähtee, on tiettyjen ehtojen täytyttävä. Ensinnäkin Nagios pitää konfiguroida niin, että ilmoitukset sallitaan. Lisäksi on olemassa erilaisia suodattimia, jotka tarkistetaan, ennen kuin ilmoitus lähtee eteenpäin. Niitä ovat mm. palvelu - ja laitesuodatin sekä yhteyssuodatin. Nämä molemmat sisältävät useampia toimintoja, jotka tarkastetaan ennen kuin ilmoitus lähtee eteenpäin. Lisäksi jokaiselle käyttäjälle voidaan määrittää millä aikaväillä ja minkälaisesta tapahtumasta lähetetään ilmoitus.

Nagios lähettää sähköpostin järjestelmänvalvojille käyttämällä siihen luotua valmista komentoa. Sama komento voidaan myös ajaa komentoriviltä. Nagios itsessään kerää ainoastaan viestinsisällön ohjelman sisällä olevista macroista. Tästä johtuen on hälytyksen yhteyteen myös mahdollista määritellä muita tarvittavia tai haluttavia toimintoja, kuten esimerkiksi SNMP-komento reitittimen tai kytkimenportin sulkemiseksi.

4.5 Järjestelmän koulutus

Opinnäytetyöhön sisältyi myös verkonvalvontaohjelmiston kouluttaminen järjestelmänvalvojatiimille. Henkilöstöresursseista johtuen kaikkien henkilöiden, jotka kuuluivat edellä mainittuun ryhmään, on pystyttävä käyttämään/konfiguroimaan Nagiosta. Koulutusta lähdettiin suunnittelemaan yhteistyössä järjestelmänvalvojatiimin esimiehen kanssa. Häneltä saatiin resurssit koulutuksen järjestämiseksi mm. koulutusajankohta, osaamistavoitteet, kesto, jne.

Koulutus koostui teoriaosuudesta ja käytännönharjoittelusta. Teoriaosuudessa käytiin läpi verkonvalvonnan perusteita ja tässä opinnäytetyössä luotu valvontajärjestelmä, sekä sen ominaisuudet. Käytännönharjoitteluosuudessa käytiin syvällisemmin läpi luotua valvontajärjestelmää ja tehtiin ns. "hands on labroja" mm. laitteiden lisäämisistä, poistamisista, valvottavien palveluiden lisäämisistä ja vian hakua.

Verkonvalvontajärjestelmän koulutustilaisuudessa tuli esille muutamia kehitysideoita joille olisi tarvetta ja niistä olisi hyötyä verkonvalvonnassa. Yhtenä esimerkkinä näistä oli, että voitaisiinko Nagioksella toteuttaa MAC (Media Access Control)-osoitteiden

valvontaa reitittimiltä ja kytkimiltä. Alustavan selvittelyn mukaan osoitteiden kysely onnistuu käyttämällä SNMP:tä. Pääasiallisena ongelmana tässä on osoitteiden jatkokäsittely, eli ne olisi saatava kyselyn jälkeen järkevästi luettavaan muotoon esim. (exel-taulukko). Tästä olisi hyötyä mm. verkon MAC-osoitteiden keräämisessä ja valvonnassa. Myös palvelimien RAID (Redundant Array of Independent Disks)-pakkojen valvonta tuli myös esille. RAID-pakkojen valvonnalla voitaisiin varmistua siitä, että rikkoutuneet levyt voidaan vaihtaa riittävän ajoissa. Tällä voidaan estää se, että palvelimen tiedostojärjestelmä ei pääse tuhoutumaan jos levyrikkoja tulee lyhyellä aikavälillä paljon.

Lisäksi pohdittiin mahdollisuutta räätälöidä Nagiokseen sellainen toiminto joka ajaa palvelimet ja työasemat alas kun UPS:n varaustila laskee liian alas. Kyseinen ominaisuus löytyy nykypäivänä hyvinkin useasta UPS:sta valmiina, mutta tällaisella ratkaisulla pystyttäisiin tekemään se keskitetysti ja järjestelmäkohtaisesti. Kaikki nämä kehitysideat jäävät tehtäväksi tämän opinnäytetyön ulkopuolelle.

5 POHDINTA

Tämän opinnäytetyön tarkoituksena oli löytää verkonvalvontaohjelmisto, joka vastaisi mahdollisimman hyvin toimeksiantajan määrittelyjä. Tämä nähtiin tarpeelliseksi verkon kasvamisen ja monipuolistumisen takia. Ongelmaa lähdettiin ratkaisemaan ensiksi keräämällä tietoperustaa verkonvalvonnasta, siihen liittyvistä toiminnoista ja periaatteista. Tiedonkeruuta vaikeutti aiheesta tehdyn kirjallisuuden vähyys ja saatavuus. Kuitenkin erilaisia suosituksia ja opinnäytetöitä on aiheesta tehty runsaasti.

Tietoperustan keräämisen jälkeen alettiin miettiä ratkaisua itse alkuperäiseen ongelmaan, joka oli soveltuvan ohjelmiston löytäminen. Tästä esimerkkinä, työssä käytettiin ITU-T:n määritelmiä suorituskyvyn- ja vikojenvalvonnasta (Ks. luvut 2.2.4, 2.2.1.) ja Hautaniemen määrittelemiä apukysymyksiä suorituskyvynvalvontaan (Ks. luku 2.2.4.) Myös SNMP-protokollan ominaispiirteet ja toiminnot täytyi ottaa huomioon. Tärkeimpänä näistä oli SNMP:n tietoturvallisuus. Tästä johtuen verkkoon täytyi

tehdä erinäisiä ratkaisuja tietoturvan parantamiseksi, mutta tietoturva syistä niitä ei voida tässä opinnäytetyössä käsitellä.

Ohjelmiston valintaan kerättiin tietoa myös muista opinnäytetöistä jotka liittyivät verkonvalvontaan. Tällä pyrittiin parantamaan tietoperustaa verkonvalvontaohjelmista ja niiden hyvistä sekä huonoista puolista. Toimeksiantajan määrittelyissä oli ohjelmiston muokattavuus asetettu painoarvoltaan korkeimmaksi, joka osaltaan johti Nagioksen valintaan verkonvalvontaohjelmistoksi, vaikka se ei välttämättä kaikilta muilta ominaisuuksiltaan ollutkaan paras. (Ks. Taulukko 3.)

Nagioksen huonoina puolina oli aikaisempien opinnäytetöiden perusteella mm. käytettävyys, joka pääasiassa johtui siitä, että asetusten tekeminen pitää normaalisti tehdä tekstitiedostoihin. Tämä ongelma pystyttiin ratkaisemaan asentamalla Nconf-ohjelma, jolla on mahdollista tehdä Nagioksen konfigurointi graafisesti selaimen välityksellä. Tällä tavalla Nagioksen käytettävyyttä pystyttiin parantamaan huomattavasti.

Lisäksi Nagiokseen asennettiin PNP4Nagios, jotta pystyttiin luomaan kuvaajia valvottavista kohteista ja niiden mittareista. Näitä kuvaajia voidaan käyttää yhdenlaisena raportointi keinona ja ne mahdollistavat osaltaan ITU-T:n X.700 suosituksen suorituskyvyn hallintaa liittyvät vaatimukset. Näiden kahden lisäosan asentamisella on ollut suuri vaikutus Nagioksen käytettävyyteen ja sen hallinnointiin. Varsinkin ilman NConf:ia olisi Nagioksen konfiguroiminen huomattavasti työläämpää. Lopputuloksena ohjelmiston valintaan tähän asteisten käyttökokemusten perusteella on oltava todella tyytyväinen.

Itse Nagioksen asentaminen ei ollut hirveän monimutkainen prosessi, mutta oma kokemus ja osaaminen linux- käyttöjärjestelmistä olivat työn alussa sen verran vähäistä, että asennus prosessin aikana tulleiden ongelmien ratkaisuun jouduttiin etsimään apuja internetistä. Toisaalta myös asennusohjeet on tehty aina jollekin linux-jakelulle, joten jakelun vaihtuessa yleensä asennuksessa joudutaan vähän sovelta-

maan eikä voida orjallisesti luottaa ohjeisiin. Näin tässäkin tapauksessa kävi. Tämä oli omiaan lisäämään tietämystä linux pohjaisista käyttöjärjestelmistä, joten työn loppua kohden asiat alkoivat tuntua helpommilta.

Vaikka verkonvalvontajärjestelmä on mielestäni onnistunut sekä toimiva, jäi siihen vielä parannettavaa ja kehitettävää. Verkonvalvonnassa ei ole kyse projektista, vaan prosessista, jota täytyy ylläpitää ja kehittää jatkuvasti. Ainoastaan tällä tavalla voidaan nykypäivänä taata ja varmistaa tietoverkkojen toimivuus ja niiden palvelujen saatavuus.

LÄHTEET

CISCO Support. 2012. SNMP Object Navigator. Viitattu 10.3.2012 http://tools.cisco.com/Support/SNMP/do/BrowseOID.do?local=en.

Cisco Systems. 2012. SNMPv3 IOS -commands Guide. Verkkodokumentti. Viitattu 23.3.2012. http://www.cisco.com/en/US/docs/ios/12_0t/12_0t3/feature/guide/Snmp3.html.

CPAN. 2012. Comprehensive Perl Archive Network. Viitattu 20.3.2012 http://www.perl.org/cpan.html.

Haikonen, J., Hlinovsky, J. & Paju, A. 2000. Teletekniikan perusteet. Kurssin harjoitus-työ. Viitattu 21.2.2012. http://www.netlab.tkk.fi/opetus/s38118/s00/tyot/47/snmp.shtml

Hautaniemi, M. 1994. TKK/Atk-keskuksen TCP/IP-verkon valvonta ja hallinta. Diplomi-työ. Aalto-yliopiston teknillinen korkeakoulu, Tietotekniikan osasto. Viitattu 16.2.2012. http://www.netlab.tkk.fi/julkaisut/tyot/diplomityot/611/thesis.html.

ITU-T X.700. 1992. Management framework for open systems interconnection (OSI) for CCITT applications. Viitattu 19.2.2012. http://www.itu.int/rec/T-REC-X.700-199209-I.

Kaario, K. 2002. TCP/IP- verkot 2.painos. Jyväskylä: Docendo Finland Oy.

Mauro, D. & Schmidt, K. 2001. Essential SNMP. Sebastopol:O'Reilly

Nagios core. 2012. Overview about Nagios. Viitattu 6.3.2012. http://www.nagios.org/about

Nagios Enterprises. 2012. Nagios Core Version 3.x Documentation. http://www.nagios.org, Documentation, Nagios Core Documentation, Nagios Core 3.x, PDF-Manual.

Nagios host/services. Viitattu 16.3.2012. http://www.debiantutorials.com/installing-nagios-core-monitoring-system-client-and-server/.

Nagios Plugins. 2012. Manpages of Nagios plugins. Viitattu 21.3.2012. http://nagiosplugins.org/man.

Nconf. 2012. What is Nconf? Viitattu 10.3.2012. http://www.nconf.org/dokuwiki/doku.php?id=nconf:about:introduction:nconf.

Nikka, T. 2007. SNMP-Verkonvalvonta vapaan lähdekoodin ohjelmilla. Opinnäytetyö. Tampereen Ammattikorkeakoulu, Tietojenkäsittelyn koulutusohjelma. Viitattu 3.3.2012. https://publications.theseus.fi/bitstream/handle/10024/10044/Nikka.Tuomas.pdf?sequence=2.

NSClient. 2012. About NSClient. Viitattu 10.3.2012. http://www.nsclient.org/nscp/

OIDview 2012. Free XUPS-MIB MIB Download.Viitattu 28.3.2012. http://www.oidview.com/mibs/534/XUPS-MIB.html

PNP4nagios 2012 PNP4Nagios Documentation. Viitattu 9.3.2012 http://docs.pnp4nagios.org/pnp-0.6/start

Puska, M. 2000. Lähiverkkojen tekniikka. 2.painos. Jyväskylä: Gummerus Kirjapaino Oy.

Sillanpää, M. 2010. Verkon monitorointi ja tikettijärjestelmä. Opinnäytetyö. Jyväskylän ammattikorkeakoulu, Tekniikan ja liikenteen ala. Viitattu 10.3.2012. https://publications.theseus.fi/bitstream/handle/10024/26545/Sillanpaa_Miika.pdf?sequence=1.

Sunnari, J. 2010. Labranet-verkon monitorointi. Opinnäytetyö. Jyväskylän ammattikorkeakoulu, Tekniikan ja liikenteen ala, Tietoverkkotekniikka. Viitattu 13.3.2012. https://publications.theseus.fi/bitstream/handle/10024/21624/20101110_jussi_sunnari_sensuroitu.pdf?sequence=1.

SNMP Research International, Inc. The SNMP Protocol. Viitattu 20.2.2012. http://www.snmp.com/protocol/

US-CERT. 2008. SNMPv3 improper HMAC validation allows authentication bypass. Viitattu 21.3.2012. http://www.kb.cert.org/vuls/id/878044

Vesa, K. 2007. Verkonvalvontasovellusten vertailu. Opinnäytetyö. Lahden Ammattikorkeakoulu, Tietotekniikan koulutusohjelma. Viitattu 3.3.2012. https://publications.theseus.fi/bitstream/handle/10024/11913/2007-11-02-03.pdf?sequence=1.

LIITTEET

Liite 1. Check_snmp plugin. (Nagios Plugins 2012.)

check_snmp v1.4.15-46-gefa2 (nagios-plugins 1.4.15)Copyright (c) 1999-2007 Nagios Plugin Development Team<nagiosplug-devel@lists.sourceforge.net>

Check status of remote machines and obtain system information via SNMP

Usage:

check_snmp -H <ip_address> -o <OID> [-w warn_range] [-c crit_range] [-C community] [-s string] [-r regex] [-R regexi] [-t timeout] [-e retries] [-l label] [-u units] [-p port-number] [-d delimiter] [-D output-delimiter] [-m miblist] [-P snmp version] [-L seclevel] [-U secname] [-a authproto] [-A authpasswd] [-x privproto] [-X privpasswd]

Options:

-h, --help Print detailed help screen

-V, --version Print version information

-H, --hostname=ADDRESS Host name, IP Address, or unix socket (must be an absolute path)

-p, --port=INTEGER Port number (default: 161)

-n, --next Use SNMP GETNEXT instead of SNMP GET

-P, --protocol=[1|2c|3] SNMP protocol version

-L, --seclevel=[noAuthNoPriv|authNoPriv|authPriv] SNMPv3 securityLevel

-a, --authproto=[MD5|SHA] SNMPv3 auth proto

-x, --privproto=[DES|AES] SNMPv3 priv proto (default DES)

-C, --community=STRINGOptional community string for SNMP communication (default is "public")

-U, --secname=USERNAME SNMPv3 username

-A, --authpassword=PASSWORD SNMPv3 authentication password

-X, --privpasswd=PASSWORD SNMPv3 privacy password

-o, --oid=OID(s) Object identifier(s) or SNMP variables whose value you wish to query

-m, --miblist=STRING List of MIBS to be loaded (default = none if using numeric OIDs or 'ALL' for symbolic OIDs.)

-d, --delimiter=STRING Delimiter to use when parsing returned data. Default is "="Any data on the right hand side of the delimiter is considered to be the data that should be used in the evaluation.

-w, --warning=THRESHOLD(s) Warning threshold range(s)

-c, --critical=THRESHOLD(s) Critical threshold range(s) –rate Enable rate calculation. See 'Rate Calculation' below --rate-multiplier Converts rate per second. For example, set to 60 to convert to per minute

-s, --string=STRING Return OK state (for that OID) if STRING is an exact match

-r, --ereg=REGEX Return OK state (for that OID) if extended regular expression REGEX matches

-R, --eregi=REGEX Return OK state (for that OID) if case-insensitive extended REGEX matches --invert-search Invert search result (CRITICAL if found)

-l, --label=STRING Prefix label for output from plugin

-u, --units=STRING Units label(s) for output data (e.g., 'sec.').

-D, --output-delimiter=STRING Separates output on multiple OID requests

-t, --timeout=INTEGER Seconds before connection times out (default: 10)

-e, --retries=INTEGER Number of retries to be used in the requests

-v, --verbose Show details for command-line debugging (Nagios may truncate output) This plugin uses the 'snmpget' command included with the NET-SNMP package.

Liite 2. Check_nt plugin (Nagios Plugins 2012.)

check_nt v1.4.15-46-gefa2 (nagios-plugins 1.4.15) Copyright (c) 2000 Yves Rubin (rubiyz@yahoo.com) Copyright (c) 2000-2007 Nagios Plugin Development Team<nagiosplug-devel@lists.sourceforge.net>

This plugin collects data from the NSClient service running on a Windows NT/2000/XP/2003 server.

Usage:

check_nt -H host -v variable [-p port] [-w warning] [-c critical]

[-l params] [-d SHOWALL] [-u] [-t timeout]

Options:

-h, --help Print detailed help screen

-V, --versionPrint version information

Options:

-H, --hostname=HOSTName of the host to check

-p, --port=INTEGER Optional port number (default: 1248)

-s, --secret=<password> Password needed for the request

-w, --warning=INTEGER Threshold which will result in a warning status

-c, --critical=INTEGER Threshold which will result in a critical status

-t, --timeout=INTEGER Seconds before connection attempt times out (default: -l, --params=<parameters> Parameters passed to specified check (see below) -d, --display={SHOWALL} Display options (currently only SHOWALL works) -u, --unknown-timeout Return UNKNOWN on timeouts10)

-h, --help Print this help screen

-V, --version Print version information

-v, --variable=STRING Variable to check