TAMPEREEN AMMATTIKORKEAKOULU

LIIKETALOUS

TUTKINTOTYÖRAPORTTI

Verkonhallinta ja verkonhallintajärjestelmä Nagios

Salla Salokannel

Tietojenkäsittelyn koulutusohjelma
toukokuu 2005
Työn ohjaaja: Harri Hakonen

TAMPERE 2005

TAMPEREEN
AMMATTIKORKEAKOULU

LIIKETALOUDEN YKSIKKÖ

Tekijä(t): **Salla Salokannel**

Koulutusohjelma(t): **Tietojenkäsittely**

Opinnäytetyön nimi: **Verkonhallinta ja verkonhallintajärjestelmä Nagios**
Network management and network management application Nagios

Työn valmistumis-
kuukausi ja -vuosi: **05/2005**

Työn ohjaaja: **Harri Hakonen** Sivumäärä: **41**

TIIVISTELMÄ

Yritysten perustoimintojen riippuvaisuus tietoverkoista luo yrityksen työntekijöille, verkonvastaaville sekä sidosryhmille helpon ja nopean tavan päivittäisten rutiinien hoitamiseen, yhteyden pitämiseen sekä yritystoiminnan ylläpitämiseen. Toisaalta tietoverkon haavoittuvaisuus, liian pienet verkon resurssit tai verkossa ilmenevät viat voivat saada koko toiminnan lamaantumaan. Hyvin suunnitellun ja toimivan verkonhallinnan avulla tietoverkkoa voidaan valvoa, seurata sen osien käyttöä sekä huomata mahdollisesti ilmenevät viat ja ongelmat jo kauan ennen kuin asia vaikuttaa loppukäyttäjän toimintaan.

Verkonhallintajärjestelmiä on tarjolla useita erilaisia. Suurin osa niistä on kaupallisia, ja niillä voidaan hallinnoida vain tietyn valmistajan laitteita. Yhtenä vaihtoehtona kaupalliselle ja laitekohtaiselle hallintajärjestelmälle on avoimeen lähdekoodiin pohjautuva verkonhallinnan työkalu Nagios. Se toimii Linux-alustalla, mutta sillä voidaan valvoa ja hallinnoida muista käyttöjärjestelmistä ja monen eri laitevalmistajan laitteista koostuvaa tietoverkkoa. Lisäksi Nagios tarjoaa rajattomat mahdollisuudet muokata sitä vapaasti yrityksen tarpeisiin sopiviksi.

Opinnäytetyön aihe on saatu toimeksiantona Pirkanmaan Hätäkeskukselta. Työn tavoitteena oli selvittää, minkälainen verkonhallintajärjestelmä Nagios on ja mitä kaikkea sillä voidaan valvoa Windows-koneista koostuvasta verkosta. Lisäksi työhön kuului rakentaa sellainen testiympäristö, jonka verkonhallintajärjestelmänä on Nagios ja johon on konfiguroituna erilaisia verkonhallinnan osia.

Työn viitekehyksenä on verkonhallinta käsitteenä, siihen kuuluvat osa-alueet sekä verkonhallinnassa huomioitavia asioita. Toisessa osassa selvitetään verkonhallintajärjestelmä Nagioksen käyttöönottoa sekä ohjelman tarjoamia vaihtoehtoja verkonhallinnan käyttöön.

Avainsanat: Verkonhallinta Verkonhallintajärjestelmät SNMP

Sisällysluettelo

1 Johdanto

Tietoverkot ovat olennainen osa nykypäivän yritystoimintaa. Usein tietoverkot ovat ratkaisevassa roolissa yrityksen päivittäisissä rutiineissa, ja pienikin häiriö tai ongelma saattaa johtaa suuriin taloudellisiin menetyksiin. Jotta ongelmilta voitaisiin välttyä, kannattaa uhrata hieman aikaa kunnollisen verkonhallintajärjestelmän luomiseen. Sen avulla viat ja ongelmat voidaan havaita ja niihin osataan puuttua jo ennenkuin ne muodostuvat todellisiksi riskitekijöiksi yrityksen perustoiminnalle. Lisäksi verkonhallinta antaa työkalut verkon kehittämiseen sekä resurssien tarkkailuun.

Verkonhallinta on pitkälti standardoitua, ja samat protokollat ovat käytössä eri valmistajien laitteilla. Kuitenkin valmistajilla on mahdollisuus kehittää täysin laitespesifisiä verkonhallintajärjestelmiä, jolloin suurien yritysten verkoissa voi muodostua ongelmaksi saada erilaisilla alustoilla toimivien eri valmistajien laitteet saman verkonhallintajärjestelmän alaisuuteen. Yhtenä vaihtoehtona voidaan käyttää Open Source-pohjaista verkonhallintatyökalua, Nagiosta, joka on vapaasti muunneltavissa täysin yrityksen tarpeita vastaaviksi.

Sain opinnäytetyölleni aiheen suorittaessani opintoihini kuuluvaa viiden kuukauden mittaista työharjoittelua Pirkanmaan Hätäkeskuksessa. Toimeksiannossa toivottiin selvitystä siitä, mitä kaikkea verkonhallintajärjestelmä Nagioksella voidaan hallinnoida ja valvoa lähinnä Windows-ympäristöstä, sekä mitä sillä kannattaa valvoa. Lisäksi toimeksiantoon kuului testiympäristön rakentaminen, missä käytössä tulisi olemaan Open Source-pohjainen verkonhallintajärjestelmä Nagios. Tavoitteena oli saada aikaan järjestelmä, jonka avulla Nagios voitaisiin ottaa käyttöön myös muissa hätäkeskuksissa yhtenä verkonhallintatyökaluna.

Opinnäytetyöraporttini tarkoituksena on kertoa mitä verkonhallinta on, mitä eri osa-alueita siihen kuuluu ja minkälaisia hyötyjä saavutetaan järjestelmällisellä verkonhallinnalla. Työn toinen osa sisältää selvityksen kuinka verkonhallintajärjestelmä Nagios otettiin käyttöön Pirkanmaan Hätäkeskukseen rakennetussa testiympäristössä, sekä perehtymisen Nagioksen toimintatapoihin.

2 Pirkanmaan Hätäkeskus

Suoritin opintoihini liittyvän työharjoittelun (30 opintopistettä) Pirkanmaan Hätäkeskuksessa Tampereella. Toimenkuvani oli järjestelmäasiantuntijan apuna työskentely, työtehtävät olivat monipuolisia ja hyvinkin vaihtelevia. Suurimassa roolissa työharjoitteluni aikana olivat uuden hätäkeskuksen käynnistys ja siihen liittyvät työt. 14.12.2004 Pirkanmaan Hätäkeskukseen siirtyivät poliisin hätäpuhelut numerosta 10022 ja 20.12.2004 pelastuksen ja sairaankuljetuksen hätänumerot numerosta 112. Jatkossa kaikkiin hätäpuheluihin vastataan numerossa 112, ja poliisin hätänumero 10022 jää vähitellen pois käytöstä.

Pirkanmaan Hätäkeskus on osa koko Suomea koskevaa hätäkeskusuudistusta, joka toteutetaan eduskunnan päätöksellä valtakunnallisesti Ahvenanmaata lukuun ottamatta. Aiemmin erillään toimineet pelastuksen kunnalliset hätäkeskukset sekä poliisin hälytyskeskukset yhdistetään valtion ylläpitämiksi hätäkeskuksiksi. Suomessa tulee olemaan yhteensä 15 hätäkeskusta. (112 – Hätäkeskuslaitos – Hätäkeskusuudistus.)

Henkilöstöä Pirkanmaan Hätäkeskuksessa on noin 60. Suurin osa on päivystäjiä, hallinnon henkilökuntaa on seitsemän ihmistä. Hätäkeskus sijaitsee Tampereen Käpylän kaupunginosassa. Pirkanmaan hätäkeskusalue koostuu 33 kunnasta. Alueella asuu 457 317 henkilöä (tilanne 31.12.2003). Hätäkeskus saa vuosittain arviolta 370 000 hätäpuhelua, joista aiheutuu 60 000 sairaankuljetustehtävää, 57 000 poliisitehtävää ja 8 000 palo- ja pelastustoimen tehtävää. (112 – Hätäkeskuslaitos – Perustietoa.)

Hätäkeskustoiminnan luonteesta johtuen tämä työ sisältää osioita, joita ei voida julkisesti esittää. Jotta työ ei kärsisi liikaa, salaiset tiedot on pyritty muuttamaan niin, että työn eheys ja ymmärrettävyys säilyy.

3 Verkonhallinta

Nykypäivän yritysten päivittäinen toiminta on oikein toimivien tietoverkkojen varassa. Oli kyseessä sitten pieni mainostoimisto, suuri matkapuhelinvalmistaja, päivittäistavarakauppa tai hätäkeskus, jokaista yhdistää riippuvuus tietoverkoista ja -järjestelmistä. Ongelmia verkon toiminnassa ilmenee varmasti, tärkeää olisi saada ne huomattua ja korjattua ennen kuin tilanteesta muodostuu este liiketoiminnalle. Verkonhallinta on osa kokonaisuutta, jossa pyritään takaamaan mahdollisimman vakaa toimintaympäristö. Muita kokonaisuuteen liittyviä osia ovat esimerkiksi palomuuri, NAT sekä virustorjunta ja roskapostin suodatus.

Perinteisessä verkonhallinnassa ongelmaan puututaan vasta kun se on jo ilmennyt ja aiheuttanut vahinkoa. Vian tai häiriön ilmetessä selvitetään häiriön laajuus, mitataan ja tutkitaan järjestelmää sekä korjataan epäilty vika. Toiminta seisoo kunnes ongelma on saatu korjattua, ja tähän saattaa kulua suhteellisen pitkäkin aika. Perinteinen tyyli ei tarjoa järjestelmän ylläpitäjille minkäänlaista historiatietoa eikä tietoa verkon liikenteestä. Koska verkot kasvavat jatkuvasti ja tietoliikenteen määrä lisääntyy, on perinteisen verkonhallinnan tilalle kehitettävä parempia ratkaisuja. (Tietoverkkolaboratorio – TKK luentomateriaali.)

Tietoverkkojen kasvu johtaa siihen, että yhä useampi kohta verkossa voi muodostua ongelmakohdaksi. Suurten verkkojen hallintaan ei enää riitä pelkät ihmistyövoimat, vaan avuksi pitää ottaa kunnollinen automatisoitu verkonhallintajärjestelmä. Standardisoidut järjestelmät takaavat, että verkonhallinta onnistuu erilaisistakin komponenteista koostuvissa verkoissa. (Hautaniemi 1994.)

3.1 Verkonhallinnalle asetettavat vaatimukset

Verkonhallinnalle asetettavat vaatimukset riippuvat siitä mistä näkökulmasta verkonhallintaa tarkastellaan ja mitä hyötyjä halutaan verkonhallinnalla saavuttaa. Liiketoiminnan kannalta verkonhallintaan kannattaa sijoittaa vain sen verran resursseja että sijoitukset ovat kannattavia. Järjestelmän ylläpitäjän kannalta verkonhallinnan tulee olla mahdollisimman helppoa ja yksinkertaista. Verkonhallinnan käytön tulee olla suunniteltua, ja siitä saadun tiedon hyödyntäminen mahdollisimman tehokasta. (Hautaniemi 1994.)

3.2 Verkonhallintaan liittyvät standardit ja protokollat

Verkonhallinnassa käytettävien työkalujen pitää olla standardisoituja, jotta ne toimivat eri laitevalmistajien laitteista rakennetuissa tietoverkoissa. Käytössä on kaksi standardisoitua protokollaa, SNMP

(Simple Network Management Protocol) ja CMIP (Common Management Information Protocol).

3.2.1 SNMP

SNMP on TCP/IP-verkonhallinnan standardiprotokolla. SNMP on tietoverkkojen hallinnassa tärkein käytetty protokolla. SNMP viittaa joukkoon verkonhallintastandardeja, joita ovat itse protokolla SNMP, tietokannan rakenteen kuvaus MIB (Management Information Base), sekä joukko tieto-olioita SMI (Structure of Management Information). (Jaakohuhta&Lahtinen 1997: 494.)

SNMP:n historia ja kehitys

Internetin kasvun alettua 1980-luvun puolivälissä tuli ajankohtaiseksi miettiä jonkinlaisen verkonhallintaprotokollan kehittämistä. Useista ehdokkaista eniten nousivat esille kolme vaihtoehtoa: HEMS (High-level Entity-Management System), mikä oli yleistys HMP:sta (Host Management Protocol), SNMP sekä CMOT (CMIP over TCP/IP), minkä yrityksenä oli tuoda ISO:n verkonhallintaprotokolla CMIP TCP/IP-verkkojen hallintatyökaluksi. Vuonna 1988 SNMP valittiin väliaikaisratkaisuksi, ja CMOT pitkän tähtäimen ratkaisuksi, perusteluna usko että OSI-protokolla syrjäyttää TCP/IP-protokollan, jolloin TCP/IP:hen perustuvia ratkaisuja ei katsottu kannattavaksi kehittää suurella volyymilla. Kuitenkin SNMP:n nopea kehitys syrjäytti hyvin pian muut verkonhallintaprotokollat, ja vuonna 1989 siitä tuli jo de facto-standardi, jonka päälle monet laitevalmistajat olivat kehittäneet järjestelmiään. Keväällä 1990 SNMP:sta tuli Internet-standardi. (Hautaniemi 1994.)

Johtuen SNMP:n puutteellisista autentikointimenetelmistä, sitä alettiin vuonna 1992 kehittämään kahteen suuntaan. Toinen suuntaus perehtyi vain ja ainoastaan turvallisuusmäärittelyihin, ja toinen kaikkeen muuhun paitsi turvallisuusseikkoihin. Vuonna 1993 nämä kaksi suuntausta yhdistettiin, jolloin tuloksena saatiin uusi standardoitu versio SNMP:sta, SNMPv2. (Hautaniemi 1994.)

SNMP:n toimintaperiaate

SNMP käyttää toiminnassaan UDP (User Datagram Protocol) -protokollaa ja sen porttia 161. SNMP on siis yhteydetön ja epäluotettava protokolla, toisin kun TCP (Transmission Control Protocol). SNMP:n jokainen kysely aiheuttaa jonkin vastauksen, joten luotettavuutta ei tarvita. Mikäli viesti ei mene perille, lähetetään kysely uudelleen. Pakettien saapumisjärjestyksellä ei ole merkitystä, sillä jokainen kysely ja vastaus käsittää vain yhden tietosähkeen. (Hunt 1998:357.)

SNMP:n toimintaan kuuluu neljä perusosaa; hallinta-asema, hallinta-agentti, hallintatietokanta sekä verkonhallinnan yhteyskäytäntö. Hallinta-asema on se laite, mikä kerää tietoja verkon tilasta hallinta-agenteilta. Hallinta-asemassa on jonkinlainen käyttöliittymä, sekä verkonhallintasovellus. Hallinta-agentti sijaitsee jokaisessa hallittavaan verkkoon kuuluvassa laitteessa. Hallintatietokanta sijaitsee jokaisessa hallittavassa laitteessa, missä toimii hallinta-agentti. Verkonhallinnan yhteyskäytännöllä määritellään kuinka hallinta-asema ja hallinta-agentti keskustelevat keskenään. SNMP:n yhteyskäytännön perusviestit ovat get, set ja trap. Get eli lukeminen on viesti, minkä hallinta-asema lähettää hallinta-agentille kysyäkseen siltä tietoja. Set eli asettaminen on viesti, millä hallinta-asema voi muuttaa hallinta-agentin olioiden arvoja. Trap:n eli ilmoituksen avulla hallinta-agentti lähettää hallinta-asemalle tiedon verkossa tapahtuneista muutoksista. (Hautaniemi 1994.)

SNMP:n PDU (Protocol Data Unit) on kyselyiden ja vastausten lähettämiseen käytetty tietosähke. Se voi olla viittä eri tyyppiä, GetRequest, GetNextRequest, SetRequest, GetResponse, tai Trap. GetRequest PDU-sanomalla hallinta-asema pyytää päivitettyjä tietoja hallinta-agentilta. GetNextRequest-sanomalla hallinta-asema kysyy taulukossa seuraavana olevat tiedot. SetRequest-viestillä hallinta-asema muuttaa hallinta-agentin laitteessa olevia tietoja. GetResponse-sanomalla hallinta-agentti vastaa hallinta-aseman GetRequest-, GetNextRequest- ja Set-Request-kyselyihin. Trap-sanomalla hallinta-agentti varoittaa epätavallisesta tilanteesta hallinta-asemaa. Hallinta-asema ei vastaa agentin lähettämiin viesteihin, joten on mahdollista että virhetilanteesta ilmoittava Trap-sanoma ei koskaan mene perille hallinta-asemalle eikä agentti tiedä asiasta mitään. (Hunt 1998:357.) SNMP aiheuttaa kyselyillään ja vastauksillaan suhteellisen paljon liikennettä verkkoon, jolloin on pohdittava sen käyttöä verkon kriittisissä osissa.

SMI

SMI määrittelee kuinka tiedot esitetään SNMP-ympäristössä. SMI sisältää tiedot kuinka määritellään ja rakennetaan MIB:t, eli olioiden hierarkisen järjestelmätietokannan. SMI:ssa kerrotaan mitkä tietotyypit ovat MIB:ssa käytettävissä, miten resurssit esitetään ja miten resurssi nimetään yksikäsitteisesti. (Tietoverkkolaboratorio – TKK luentomateriaali.)

SMI:n käyttämä objektien nimeämiskäytäntö perustuu ISO:n hallinnoimaan hierarkiseen nimiavaruuteen. Jokainen valvottava objekti saa oman yksilöllisen nimensä, objektitunnuksen (object identifier) mikä on samalla sen hierarkiatasoa esittävä numero. Objektit muodostavat yhdessä puumaisen rakenteen, mikä laajenee alaspäin eri haaroiksi ja useiksi tasoiksi. Objektien nimeämiseen käytetään ISO:n standardia ASN.1 (Abstract Syntax Notation One). Tämän formaatin

ansiosta tietoja voidaan määritellä riippumatta käytössä olevasta tietokone- tai merkkijärjestelmästä. (Hunt 1998:359.)

MIB

MIB on hallintatietokanta, mikä määrittelee mitkä hallintaoliot löytyvät mistäkin laitteesta. Se rakentuu SMI:n määrittelemän nimeämiskäytännön mukaisesti, mutta voidaan koota halutuista objekteista hyvinkin räätälöidysti tietyn laitteen valvonta- ja hallintavaatimuksien mukaisesti. MIB kerää tietoja laitteesta tietokantaan, ja agentin kautta tiedot lähetetään hallinta-asemalle verkonhallintajärjestelmään.
(Leinwald & Conroy 1996:153.)

MIB:sta on olemassa kaksi versiota, MIB1 sekä MIB2. MIB1 on MIB2:sen alijoukko, ja TCP/IP-protokollapinon standardi MIB-tietokanta. MIB2 sisältää enemmän objekteja ja on kehittyneempi versio MIB1:sta. Molemmat MIB:t ovat RFC-dokumenteissa määriteltyjä standardeja. (Leinwald & Conroy 1996:153.)

Verkonvalvontajärjestelmän on tuettava laitteilla käytössä olevia MIB:a, jotta niiden lähettämä tieto hallinta-agenttien kautta hallinta-asemalle saadaan ymmärrettävään muotoon verkon ylläpitäjälle. Joillakin laitevalmistajilla saattaa olla käytössään sellaisia MIB:a, ettei niiden antaman tiedon esittäminen onnistu kuin saman valmistajan tekemällä ohjelmistolla. (Hunt 1998:360.)

RMON

RMON (Remote Monitoring) kerää tietoja verkkoliikenteestä ja auttaa analysoimaan verkossa tapahtuneita muutoksia. Se toimii RMON-MIB:n päällä. RMON:lla on yhdeksän objektiryhmää, joiden avulla kerätyn tiedon mukaan voidaan tehdä päätelmiä verkon tilasta.

Tilastoryhmä (statistics) kerää perusinformaatiota kaikista määritellyistä aliverkoista, joiden liikenne kulkee Ethernet-liitynnän läpi. Ryhmä tilastoi pakettien pituuden, tavujen ja pakettien määrän, sekä havaitut pakettien virheet ja törmäykset. Historiaryhmä (history) voidaan asettaa keräämään tietoa joltain tietyltä aikaväliltä. Hälytysryhmässä (alarm) asetetaan laskureille rajoja, joiden mukaan lähetetään hallinta-asemalle varoitusviestejä (trap). Host-ryhmän taulukkoon kerätään tietoa laitteiden vastaanottamista ja lähettämistä paketeista, lähetetyistä virhepaketeista sekä multicast- ja broadcast-paketeista. HostTopN-ryhmään voidaan määritellä kriteerejä, joiden perusteella saadaan tietoa esimerkiksi eniten liikennettä aiheuttavasta laitteesta. Matrix-ryhmässä seurataan tiettyjen laiteparien välistä liikennettä. Filter-, capture- ja event-ryhmät kuuluvat yhteen. Filter- ja capture-ryhmiin määritellään erilaisin suodattimin mitä verkosta seurataan. Event-ryhmään määritellään minkälainen ilmoitusviesti lähetetään

hallinta-asemalle, jos filter- tai capture-ryhmän suodattimien ehdot täyttyvät. (Hautaniemi 1994.)

SNMP:n ongelmia

SNMP:lla on kolme heikkoa kohtaa; turvallisuus, yhteensopivuus vain tietyn verkkoprotokollan kanssa sekä suurista tietokannoista tapahtuvan tiedon etsiminen. SNMP lähettää viestit verkossa selväkielisenä tekstinä ilman salausta. Niinpä jokainen jolla on pääsy verkkoon tutkimaan verkon liikennettä, voi lukea SNMP:n viestien sisällön ja käyttää hyväkseen niistä saatuja tietoja. (Leinwald & Conroy 1996:165.)

SNMP on suunniteltu ja standardisoitu vain IP-verkkojen valvontaan ja hallintaan. Vaikka suurin osa tietoverkoista onkin IP-perustaisia, on muitakin protokollia olemassa sekä yleisesti käytössä. Laajoissa tietokantahauissa SNMP aiheuttaa verkolle suurta liikennettä, sillä sen viestintäperiaate tukee vain yksinkertaisia hakuja, jolloin palautetaan vain yksi tieto kerrallaan yhdessä kyselyssä. Mikäli tietokannasta halutaan saada esimerkiksi 2000 riviä sisältävä taulu, missä jokainen rivi aiheuttaa 4 erilaista kyselyä, saadaan Get-Next-Request-viestin aiheuttamilla Get-Request sekä Get-Response-viesteillä aikaan 2*4*2000 eli 16000 pakettia. (Leinwald & Conroy 1996:166.)

3.2.2 CMIP

CMIP on osa ISO:n OSI-järjestelmänhallintaa, ja se tukee tiedon vaihtoa verkon hallintajärjestelmän sekä hallinta-agenttien välillä. CMIP:n suurimpina kehittäjinä ovat olleet hallitukset sekä yritykset, ja sen tärkeimpänä päämääränä on ollut korjata ja parantaa SNMP:ssa esiintyneet puutteet. CMIP jaetaan kahteen osaan; TCP/IP-verkkojen päällä toimiva CMOT sekä IEEE 802-standardille kehitetty CMOL (CMIP over LLC). (Common Management Information Protocol.)

CMIP käyttää ISO:n luotettavaa ja yhteydellistä siirtomekanismia, sekä turvallisuusmenetelmää joka tukee pääsylistoja, valtuutusta sekä turvallisuuslokien keräämistä. SNMP:n tapaan CMIP:lla on käytössään viestien välittämiseen verkossa PDU:t. CMIP:lla näitä tietosähkeitä on käytössään yksitoista erilaista SNMP:n viiden sijaan. Muita etuja SNMP:hen nähden turvallisuusseikkojen sekä monipuolisempien muuttujien lisäksi ovat parempi verkonhallintajärjestelmän hyödyntäminen että laadukkaampien raporttien laadinta myös epätavallisista verkossa ilmenneistä tiloista. (Common Management Information Protocol.)

CMIP:tä käytetään enemmän telekommunikaatiopuolella, ja se onkin ITU:n (The International Telecommunication Union) hyväksymä protokolla TMN (Telecommunication Management Network)-standardia

tukevien laitteiden hallinnointiin. Lähiverkot pohjautuvat kuitenkin suurimmaksi osaksi TCP/IP-protokollapinoon, ja lähiverkkojen laitteet tukevat enimmäkseen SNMP-protokollaa. Tämä on yksi suuri tekijä, minkä takia CMIP:sta ei ole tullut kovinkaan suosittua protokollaa lähiverkkojen hallinnointiin. (Common Management Information Protocol.) Muita syitä CMIP:n harvinaisuuteen ovat CMIP:n suuri resurssien käyttö, sekä protokollan monimutkaisuus, mikä vaatii verkon käyttäjältä paljon asiantuntemusta sekä koulutusta (Leinwald & Conroy 1996:190).

3.3 Verkonhallinnan osa-alueet

Jotta verkonhallintaa voidaan tarkastella kokonaisuutena, on tunnistettava siihen kuuluvat osa-alueet. ISO on määritellyt osana OSI-järjestelmähallintaa verkonhallintaan kuuluvat toiminnalliset osiot. Tämä jako on yleisesti käytössä myös muiden verkonhallintajärjestelmien vaatimusten kuvauksessa. ISOn määrittelemät avainalueet verkon hallinnassa ovat vikojen hallinta, käytön hallinta, kokoonpanon hallinta, suorituskyvyn hallinta sekä turvallisuuden hallinta. (Hautaniemi 1994.)

3.3.1 Vikojen hallinta (fault management)

Vikojen hallinnassa keskitytään paikallistamaan, eristämään ja mahdollisesti korjaamaan tietoverkossa ilmennyt vika (Leinwald & Conroy 1996:9). Ensimmäinen askel vikojen hallintaan on määritellä jonkinlainen varoitusjärjestelmä mikä ilmoittaa viasta sen ilmetessä. Yksinkertaisimmillaan varoitus voi olla laitteen kyljessä palava valo, mutta kehittyneemmässä verkonhallinnassa verkon ylläpitäjä saa tiedon viasta ja sen sijainnista suoraan työpöydälleen. (Duck & Read 2003:337-338.)

Verkossa ilmenevä vika vaikuttaa yleensä välittömästi verkon suorituskykyyn ja käyttäjien saatavilla oleviin resursseihin. Vikojen hallinta on yksi suurin tekijä verkonhallinnassa, mutta hyvin toteutettuna se voi itsessään kasvattaa verkon vikasietoisuutta. Siinä saadaan nopeasti havaittua missä vika on ja mitä vaikutuksia vialla on verkon toimintaan. Erilaisilla mekanismeilla voidaan varmistua verkon toimimisesta oikein, esimerkiksi tutkimalla lokitiedostoja, hälytyksiä tai raportteja vioista. Lisäksi vikojen hallinnan tulisi toimia tehokkaasti, mutta aiheuttamatta ylimääräistä liikennettä verkossa. (Jaakohuhta & Lahtinen 1997:496 – 497.)

Vikojen hallinta kasvattaa tietoverkon luotettavuutta antamalla ylläpitäjille välineet nopeasti paikallistamaan verkossa esiintynyt vika ja ryhtyä korjaustoimenpiteisiin. Mikään verkko ei kuitenkaan voi toimia ilman koskaan ilmeneviä ongelmia tai katkoksia, vaikka verkon käyttäjät niin olettavatkin. Kun tietoverkko menettää laitteen tai linkin

toiminnan, on tärkeää kuitenkin minimoida sen vaikutukset käyttäjille, ja parhaassa tapauksessa tilanne saadaan pelastettua ilman, että käyttäjät edes huomaavat asiaa. Vikojen hallinnan tehokkaalla käytöllä saadaan muutettua tilanne verkonvalvojan kannalta edullisempaan suuntaan: keskitytään vikojen korjaamisen sijasta niiden ennaltaehkäisyyn. (Leinwald & Conroy 1996:38.)

3.3.2 Käytön hallinta (accounting management)

Käytön hallinnassa kerätään tietoa verkon resurssien käytöstä, ja sen avulla voidaan esimerkiksi laskuttaa organisaation yksiköitä ja määritellä resurssien saatavuutta. Jos jollain osa-alueella huomataan huomattavaa verkon resurssien kulutusta, voidaan asiaan puuttua lisäämällä oikealle osalle lisäresursseja. (Leinwald & Conroy 1996:12-13.)

Verkon ylläpitäjä määrittelee käytön hallinnalle asetettavat vaatimukset. Mikäli organisaatiossa halutaan käyttää käytön hallinnasta saatuja tietoja laskutuksen perustana, on ylläpitäjän mietittävä ja määriteltävä minkälaista tietoa verkosta kerätään, mistä verkon osista sitä kerätään ja mikä on se aikaväli milloin tiedot kootaan yhteen. Käyttäjillä ei saa olla pääsyä kerättyyn tietoon. Käytön hallinnasta saatavia lukuja voidaan käyttää myös määriteltäessä lisäresursseja tarpeen vaatiessa. (Jaakohuhta & Lahtinen 1997:498-499.)

Käytön hallinnasta suurin saatava hyöty on sen tarjoama mahdollisuus seurata ja tilastoida tarkasti verkon resurssien kuormitusta. Jos suunnitellaan verkon laajentamista, on helppoa päättää mitkä osat tarvitsevat lisäyhteyksiä tai -palveluita. Tulosvastuullisissa organisaatioissa voidaan käytön hallinnan perusteella saaduista luvuista laskuttaa toisia yksiköitä ja organisaatioita. (Jaakohuhta & Lahtinen 1997:499.)

3.3.3 Kokoonpanon hallinta (configuration management)

Kokoonpanon hallinta tarkoittaa jokaisen verkkoon kuuluvan komponentin dokumentointia, esimerkiksi mikä versio jostain ohjelmasta on käytössä milläkin laitteella. Päivityksiä tehtäessä voidaan dokumentista tutkimalla selvittää mitkä laitteet tarvitsevat päivitystä uudempaan versioon. Ilman käytön hallintaa jokainen laite pitäisi jokaisen muutoksen yhteydessä tutkia erikseen.(Leinwald & Conroy 1996:10.)

Kokoonpanon hallinnan suurimpana vaatimuksena on mahdollisuus sen avulla käynnistää ja pysäyttää verkon laitteita, erityisesti vikatilanteissa. Silloin jokainen verkon laite ja osa pitää olla tarkasti dokumentoituna, jotta pystytään tekemään hallittuja muutoksia verkon rakenteeseen toiminnan siitä kärsimättä. (Jaakohuhta & Lahtinen 1997:500.)

Kun verkon pääkäyttäjä on kerännyt kaikki tiedot kokoonpanon hallintaa varten, on tietojen päivittäminen tietysti myös tärkeää. Näin saadaan aina ajantasalla olevaa tietoa verkon komponenteista ja laitteiden ohjelmistoversioista. (Jaakohuhta & Lahtinen 1997:501.)
Joskus verkossa piilevä vika saattaa johtua niinkin yksinkertaisesta asiasta kuin kytkimien porttien erilaisista nopeusmäärittelyistä ja siitä seuranneesta liikenteen hidastumisesta.

3.3.4 Suorituskyvyn hallinta (performance management)

Suorituskyvyn hallinnassa seurataan verkon resurssien käyttöä. Esimerkkejä mitattavista ominaisuuksista ovat verkon kuormitus, prosenttimääräinen resurssien käyttö, virhesummat ja vasteaika. Oikeanlaisella verkon suorituskyvyn hallinnalla pystytään huomaamaan verkon pullonkaulat jo ennen kuin käyttäjältä ehtii tulemaan palautetta asiasta. (Leinwald & Conroy 1996:12.)

Suorituskyvyn hallinta koostuu kahdesta osasta: tietoverkon valvonta ja tietoverkon hallinta. Valvonnassa kerätään tietoja verkon resurssien käytöstä ja seurataan mitkä verkon osat aiheuttavat eniten liikennettä, missä tapahtuu eniten pakettien törmäyksiä ja kuinka pitkiä ovat vasteajat erilaisten sovellusten käytössä. Hallinnassa keskitytään säätämään verkon osia valvonnasta saatujen mittaustulosten perusteella. Suorituskyvyn hallinta menee osittain päällekkäin käytön hallinnan kanssa, mutta suorituskyvyn hallinnassa keskitytään kuitenkin enemmän verkon laitteiden hallintaan, kun taas käytön hallinnassa seurataan verkon resurssien käyttöä. Suorituskyvyn hallinnasta saatujen tulosten perusteella voidaan miettiä verkon mahdollista laajentamista tai pohtia ratkaisuja verkossa esiintyneisiin pullonkauloihin esimerkiksi reititystietoja muuttamalla tai liikenteen hajauttamisella. (Jaakohuhta & Lahtinen 1997:501-503.)

3.3.5 Turvallisuuden hallinta (security management)

Turvallisuuden hallinta tarkoittaa tietoverkon käyttäjille asetettuja rajoituksia. Kaikki tietoverkossa oleva tieto ei saa olla kaikkien saatavilla, ja salattujen tietojen käyttöä seurataan. (Leinwald & Conroy 1996:11.)

Turvallisuuden hallinta verkonhallinnassa ei tarkoita eri käyttäjäryhmille tai käyttäjille annettavia oikeuksia ja tunnuksia verkon resurssien käyttöä varten. Turvallisuuden hallinta on määrittelyä ketkä pääsevät tutkimaan verkonhallintaan kuuluvia tietoja, sekä kenellä on pääsy verkkoon liitettyihin laitteisiin. Turvallisuuden hallintaan kuuluu olennaisena osana erilaisten lokitietojen kerääminen, tallentaminen ja analysoiminen. (Jaakohuhta & Lahtinen 1997: 503.)

Vaatimuksena turvallisuuden hallinnalle on kyky tarjota välineitä verkonhallinnasta saatavan tiedon turvaamiseen. Turvallisuuden hallinta lisää koko tietojärjestelmän turvallisuutta kun kriittisiin tietoihin ja laitteisiin pääsy on mahdollisimman rajoitettua. (Jaakohuhta & Lahtinen 1997:503-504.)

Verkonhallinta tarkoittaa itseasiassa kahta asiaa; verkon valvontaa ja verkon hallintaa. Verkon valvonnan tärkeimpiä tehtäviä on verkon tilan havannointi. Siihen kuuluvat suorituskyvyn valvonta, vikojen valvonta sekä käytön valvonta. Verkon hallintaa ovat kokoonpanon sekä turvallisuuden hallinta, ja niiden avulla ylläpidetään verkkoon kuuluvien eri laitteiden ja komponenttien asetuksia. Verkon valvonta ja verkon hallinta muodostavat yhdessä kokonaisuuden, jota kutsutaan verkonhallinnaksi. Verkonhallintajärjestelmä on ohjelmisto, jonka avulla verkonhallinta toteutetaan. (Jaakohuhta & Lahtinen 1997:504-505.)

3.4 Verkonhallintajärjestelmän arkkitehtuuri

Verkonhallintajärjestelmän ulkoinen arkkitehtuuri voidaan jakaa kolmeen erilaiseen malliin; keskitetty, hierarkinen tai hajautettu. Jokaisessa mallissa on haittansa ja hyötynsä, ihannetila tietysti olisi että ulkoinen arkkitehtuuri vastaisi organisaation verkon todellista rakennetta.

Keskitetyssä arkkitehtuurissa verkonhallinta suoritetaan yhdellä koneella, mikä sisältää kaiken tiedon verkosta ja siihen kuuluvista laitteista ja palveluista. Hyötynä saavutetaan järjestelmän yksinkertaisuus sekä turvallisuuden takaaminen. Kun kaikki tieto sijaitsee yhdessä paikassa, on järjestelmän tietojen tutkiminen ja päivittäminen yksinkertaista. Lisäksi voidaan helposti varmistaa, ettei järjestelmään pääse ulkopuoliset tunkeutujat käsiksi sallimalla pääsy järjestelmään vain tietyille henkilöille. Huonona puolena keskitetyssä järjestelmässä on sen heikko vikasietoisuus. Varmuuskopionti on tärkeää, sijoitettuna jollekin toiselle koneelle. Laitteiden lisääminen hallintajärjestelmään saattaa muodostua vaikeaksi, mikäli verkko kasvaa kovinkin suureksi. Keskitetyn järjestelmän huomattavin haitta on kuitenkin sen aiheuttama suuri liikennemäärä. Kun yksi kone ottaa yhteyttä kaikkiin hallittaviin koneisiin, ja hallittavat koneet vastaavat järjestelmälle, voi verkon liikenne kasvaa liian suureksi resursseihin nähden ja muu verkkoliikenne kärsii. (Leinwald & Conroy 1996:22-23.)

Hierarkinen verkonhallinnan arkkitehtuuri käyttää palvelin-asiakasmallia. Järjestelmässä tietyt toiminnot suoritetaan järjestelmän keskuskoneella, ja tietyt toiminnot asiakaskoneissa. Järjestelmässä asiakaskoneet käyttävät palvelimella toimivaa tietokantaa verkon yli, mutta tekevät itsenäisiä kyselyitä niiden alaisuuteen määritellyille laitteille. Järjestelmä ei ole kokonaan riippuvainen yhdestä laitteesta,

verkonhallintatehtävät ja verkon valvonta on jaettu kaikille laitteille, mutta käytössä on keskitetty tietokanta. Esimerkkinä hierarkisesta arkkitehtuurista on RMON, eli etähallintajärjestelmä, missä määritelty laite kerää tietoja verkosta ja liikenteestä ja lähettää yhteenvetoja järjestelmän keskuskoneelle. Verkonhallinnan konfiguroiminen hierarkiseen järjestelmään vaatii etukäteissuunnittelua, jotta vältyttäisiin päällekkäisiltä kyselyiltä samalle laitteelle. (Leinwald & Conroy 1996:23-25.)

Hajautettu arkkitehtuuri on keskitetyn ja hierarkisen verkonhallinnan arkkitehtuurin yhdistelmä. Siinä verkonhallintajärjestelmä on jaettu osiin, joista jokainen hallinnoi omaa osuuttaan verkosta. Jakoperusteena voi olla vaikka laitteiden konkreettinen sijainti tai voidaan yhdistää samanlaiset laitteet (kytkimet, reitittimet, palvelimet) omiin osioihinsa. Järjestelmässä toimii useita tietokantoja, jotka kannattaa replikoida keskenään. Hajautettu arkkitehtuuri sisältää niin keskitetyn kuin hierarkisen arkkitehtuurin edut. Yhdessä paikassa säilytetään yhden hallittavan verkonosion tiedot, yhdestä paikasta päästään kaikkiin verkonhallintasovelluksiin, arkkitehtuuri ei ole riippuvainen yksittäisestä koneesta sekä verkonhallintatehtävät ja verkon monitorointi on jaettu verkon eri osille. (Leinwald & Conroy 1996:25-26.)

3.5 Verkonhallintajärjestelmälle asetettavia vaatimuksia

Hyvän verkonhallintajärjestelmän tarkoituksena on hallita ja valvoa useista erilaisista komponenteista koostuvaa verkkoa. Verkonhallintajärjestelmän pitäisi sisältää graafisen käyttöliittymän, karttakuvan verkosta, jonkinlaisen tietokannan, standardoidun menetelmän kyselyjä varten, muokattavissa olevia valikoita sekä mahdollisuuden seurata tapahtumalokeja (Leinwald & Conroy 1996:18).

Graafinen käyttöliittymä antaa käyttäjälle mahdollisuuden seurata helposti järjestelmän tapahtumia sekä syöttää järjestelmään tietoja. Karttakuvan avulla voidaan tutkia verkon rakennetta graafisesti, ja huomioidaan nopeasti missä verkon osassa vika ilmenee. Jos karttakuvaan lisätään värit ja erilaiset kuvat ilmentämään verkon osien tilaa, pystytään yhdellä silmäyksellä tarkistamaan verkon komponenttien tila. Standardoiduilla kyselyillä voidaan kerätä tietoja eri valmistajien ja erilaisilla alustoilla toimivien verkon osien tilasta. Ihannetilassa kyselyt voidaan suorittaa yhdellä verkonhallintaprotokollalla. (Hautaniemi 1994.)

Tietokanta on tarpeellinen tietojen tallentamista ja hakua varten. Useat tietokantahallintajärjestelmät mahdollistavat käyttäjien laativan eriteltyjä raportteja ja suorittaa automaattisia varmuuskopiointeja. Tapahtumalokien avulla voidaan seurata verkossa tapahtuneita hälytyksiä ja ongelmia. (Leinwald & Conroy 1996:18-20.)

Verkonhallintajärjestelmän on annettava käyttäjälle mahdollisuus laajentaa ja muokata järjestelmää käyttäjien tarpeita vastaaviksi. (Hautaniemi 1994.)

Mikäli järjestelmä on suunniteltu vain tietyn valmistajien tiettyjen koneiden hallintaan, on se tarpeeton, mikäli verkon komponenteissa tapahtuu muutoksia (Leinwald & Conroy 1996:20). Koska verkonhallintajärjestelmä sisältää yrityksen toiminnan kannalta oleellisia tietoja, ovat turvallisuusasiat pidettävä myös mielessä.

Useat suuret laitetoimittajat tarjoavat standardeihin perustuvaa verkonhallintajärjestelmää, jolla ei kuitenkaan kyetä hallinnoimaan kun valmistajan omia laitteita. Tarjolla on kuitenkin alustariippumattomia järjestelmiä, joihin saadaan konfiguroitua koko organisaation verkko laitevalmistajista riippumatta. Nämä ohjelmat eivät välttämättä perustu SNMP-protokollaan, vaan toimivat niin sanotulla pollausmenetelmällä. Järjestelmälle määritellään koneiden IP-osoitteet ja portit, joita halutaan tarkkailla. Määrätyin väliajoin ohjelmat käyvät tarkastamassa halutut palvelut käyttäen sopivaa protokollaa, ja jos on aihetta antaa varoitus, niin se voidaan lähettää sähköpostiin tai matkapuhelimeen. (Feldman1999:366–368.)

4 Nagios

Nagios on avoimeen lähdekoodiin perustuva verkonvalvonta ja -hallintaohjelmisto. Se valvoo käyttäjän määrittelemiä verkon komponentteja, ja ilmoittaa jos jossain verkon osassa ilmenee ongelma ja kun asiat ovat taas kunnossa. Alun perin Nagios on suunniteltu toimimaan Linux-käyttöjärjestelmän päällä, mutta se toimii hyvin myös muiden yleisimpien *NIX-variaatoiden kanssa. (Nagios : About Nagios.)

Nagioksella voidaan valvoa verkosta useita eri asioita. Sillä voidaan monitoroida erilaisia verkkopalveluita, kuten SMTP (Simple Mail Transfer Protocol), POP3 (Post Office Protocol 3), HTTP (Hypertext Transfer Protocol), NNTP (Network News Transfer Protocol), PING (verkossa olevan laitteelle lähetettävä viesti, jolla tutkitaan onko laite tavoitettavissa), FTP (File Transfer Protocol) ja DNS (Domain Name Service). Nagioksella voidaan seurata verkossa olevien työasemien ja palvelimien resursseja, esimerkiksi prosessorin kuormaa, kovalevyn ja muistin käyttöä, käynnissä olevia sovelluksia, lokitiedostoja sekä käyttäjien määrää. Nagioksella pystytään helposti seuraamaan verkon tilaa. Nagios voidaan määritellä lähettämään varoitusviesti vastuuhenkilön matkapuhelimeen tai sähköpostiin. Nagios sisältää hyvät raportointi- ja analysointityökalut päätöksentekoa varten. Sillä voidaan hallinnoida ja valvoa erilaisista käyttöjärjestelmistä koostuvaa verkkoa, ainoastaan itse ydinohjelma täytyy asentaa Linux-koneelle. (Nagios : About Nagios.)

4.1 Laitteisto- ja ohjelmistovaatimukset

Ainoa vaatimus Nagioksen toiminnalle on PC, mihin on asennettuna Linux-käyttöjärjestelmä (tai jokin Unix-variantti) sekä c-tulkki. TCP/IP pitää olla konfiguroituna jotta järjestelmän tarkistukset liikkuvat verkon yli. Jos käytetään CGI:tä, tarvitaan myös käynnissä oleva sekä oikein konfiguroitu www-palvelin (suositeltuna Apache). Verkkoasetusten täytyy olla luonnollisesti oikein (IP-osoite, verkkomaski, oletusyhdyskäytävä sekä DNS:n osoite.)

Mikäli ei olla varmoja mitä osia Linuxin asennuksessa tulee ottaa mukaan, kannattaa valita täydellinen Linuxin asennus. Nagioksen toimimattomuus saattaa johtua jonkin asennuksessa poisjätetyn osan puuttumisesta. Graafinen käyttöliittymä kannattaa asentaa, sillä se helpottaa huomattavasti Nagioksen käyttöä.

4.2 Lisenssi

Nagios on lisensoitu GNU-projektin General Public Lisence-ehtojen mukaan. Se tarkoittaa, että Nagiosta saa laillisesti kopioida, jakaa ja

muunnella yksinkertaisten ehtojen mukaan. Nagios on täysin ilmainen, se ei sisällä minkäänlaisia lisenssi- tai päivitysmaksuja. Myös plug-in:a saa vapaasti muokata omiin tarkoituksiin paremmin sopiviksi. (Nagios : About Nagios.)

4.3 Nagioksen asentaminen ja konfigurointi

Seuraavassa kerron kuinka Nagios on asennettu ja konfiguroitu nimenomaan tässä työssä toimivaksi. Nagioksen kotisivuilta löytyy täydellinen dokumentaatio palvelun käynnistämistä varten. Nagioksen saa ladattua osoitteesta www.nagios.org, tällä hetkellä uusin saatavana oleva versio on 2.0. Tässä työssä on käytetty versiota 1.2, koska se on viimeisin tällä hetkellä saatavana oleva vakaa versio Nagioksesta. Nagioksen käyttöönotto ei käy käden käänteessä vain lataamalla ohjelma Internetistä ja asentamalla se. Palvelun aloittaminen vaatii paljon konfigurointia ja kärsivällisyyttä ennen kuin mitään voidaan monitoroida.

4.3.1 Ohjelman lataus ja asentaminen

Nagios koostuu monesta eri osasta. Ydinohjelmaa ajetaan Nagios-koneella demonina, se aikatauluttaa ja suorittaa koneiden ja palveluiden tarkastukset. Cgi-tiedostot esittävät koneiden ja palveluiden tilan, ja niiden kautta määritellään mitä valvotaan ja miten. Pluginien avulla suoritetaan koneille määritellyt palvelut.

Ensimmäisenä kirjaudutaan Linuxiin pääkäyttäjän (root) tunnuksilla. Luodaan väliaikaishakemisto Nagioksen lataamista ja asentamista varten hakemistoon /tmp/nagios mkdir-komennolla. Ladataan Nagioksen kotisivuilta osoitteesta www.nagios.org/download Nagioksen ydinohjelman versio 1.2, sekä plug-in:sta versio 1.3.1. Avataan komentotulkki, ja siirrytään oikeaan hakemistoon komennolla cd /tmp/nagios/ . Puretaan ladatut tiedostot ja luodaan hakemisto purettavia tiedostoja varten kirjoittamalla ensin rivi gunzip nagios-1.2.tar.gz ja seuraavaksi rivi tar xf nagios-1.2.tar.

Luodaan Nagiokselle oma hakemisto komennolla mkdir /usr/local/nagios. Hakemistorakenne on vapaasti valittavissa, mutta edellä mainittu hakemistorakenne on suositeltava, sillä sitä käytetään dokumentaatioissa sekä esimerkkikonfiguraatioissa. Luodaan käyttäjä nagios komennolla adduser nagios. Komento lisää käyttäjän automaattisesti luomaansa ryhmään nagios.

Lisätään http-palvelun hakemistoon hakemisto, josta ajetaan Nagioksen cgi-tiedostoja (/var/www/html/nagios/cgi-bin), sekä html-tiedostoja (/var/www/html/nagios/). Hakemistoihin ei laiteta mitään tiedostoja, vaan niihin viitataan Apachen konfigurointitiedostossa skriptialiaksilla sekä aliaksilla. Näihin palataan Apachen konfigurointikohdassa.

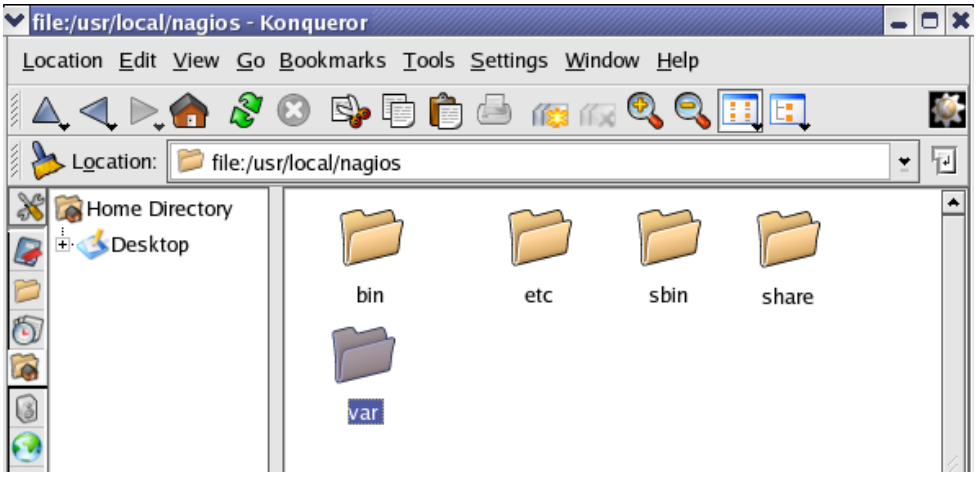
Ajetaan konfigurointiskripti, joka on mallia ./configure –prefix=prefix --with-cgiurl=cgiurl --with-htmurl=htmurl --with-nagios-user=someuser --with-nagios-grp=somegroup, missä

prefix = Nagioksen kotihakemisto, tässä /usr/local/nagios
cgiurl = hakemisto cgi:en ajamista varten,
tässä /var/www/html/nagios/cgi-bin
htmurl = html-tiedostojen sijaintihakemisto,
tässä /var/www/html/nagios/
someuser = nagios
somegroup = nagios.

Tässä tapauksessa konfigurointiskripti on siis kokonaisuudessaan ./configure --prefix=/usr/local/nagios
--with-cgiurl=/var/www/html/nagios/cgi-bin
--with-htmurl=/var/www/html/nagios/
--with-nagios-user=nagios --with-nagios-grp=nagios.

Kirjoitetaan komentotulkkiin käsky make all. Se kääntää Nagioksen ja cgi:t. Kirjoitetaan seuraavaksi make install, joka asentaa binäärit ja html-tiedostot. Kirjoitetaan sitten make install-init, joka muodostaa init-skriptin, jolloin Nagiosta voidaan ajaa demonina.

Siirrytään Nagioksen hakemistoon komennolla cd /usr/local/nagios. Listataan hakemiston sisältö komennolla –ls. Hakemiston pitäisi sisältää viisi alihakemistoa, kuten kuva 1 esittää.



Kuva 1 Hakemiston /usr/local/nagios sisältö

/bin sisältää Nagioksen ydinohjelman
/etc sisältää konfiguraatiotiedostot
/sbin sisältää cgi:t
/share sisältää html-tiedostot, joiden avulla Nagiosta voidaan käyttää selaimessa
/var on vielä tyhjä, sinne tulevat lokitiedostot.

Asennetaan ladatut plug-in:t samalla tavoin kuin ydinohjelma. Plug-in:t asentuvat asennuksen aikana automaattisesti luomaansa hakemistoon /usr/local/nagios/libexec.

4.3.2 WWW-palvelimen konfigurointi

Nagios ei toimi ilman oikein konfiguroitua www-palvelinta. Suositelluin vaihtoehto on Apache, johon voi tutustua lisää Apachen kotisivuilla www.apache.org. Apache tulee lähes kaikkien Linux-distribuutioiden mukana, ja sen konfigurointi on helppoa tekstieditoria käyttämällä. Nagiosta varten Apachen konfiguraatiotiedostoon (httpd.conf) tarvitsee tehdä muutamia muutoksia ja lisäyksiä. Kannattaa ottaa alkuperäisestä tiedostosta kopio, ja nimetä se esimerkiksi httpd.conf.orig.

Cgi:t tarvitsevat toimiakseen oikein määritellyt skriptialiakset. Lisätään Apachen konfiguraatiotiedostoon ScriptAlias kohtaan seuraavat rivit:

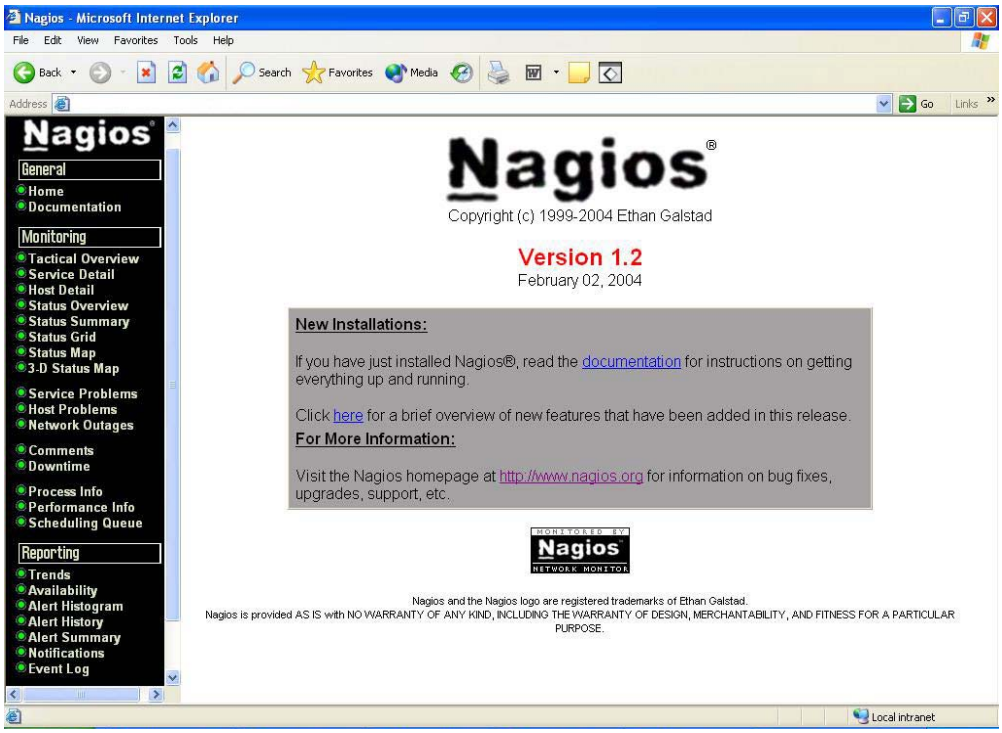
```
ScriptAlias /var/www/html/nagios/cgi-bin/
/usr/local/nagios/sbin/
<Directory “/var/www/html/nagios/cgi-bin”>
AllowOverride AuthConfig
Options ExecCGI
Order allow, deny
Allow from all
</Directory>
<Directory “/usr/local/nagios/sbin”>
AllowOverride AuthConfig
Options ExecCGI
Order allow, deny
Allow from all
</Directory>
```

Aliaksen määrittely täytyy laittaa vasta skriptialiasten jälkeen, muutoin Apache ei osaa lukea niitä oikeassa järjestyksessä. Lisätään heti edellisen perään seuraavat rivit:

```
Alias /nagios/ /usr/local/nagios/share/
<Directory “/var/www/html/nagios/”>
AllowOverride AuthConfig
Order allow, deny
Allow from all
</Directory>
```

<Directory “/usr/local/nagios/share”>
AllowOverride AuthConfig
Order allow, deny
Allow from all
</Directory>

Käynnistetään Apache uudestaan kirjoittamalla komentotulkkiin kehoite /etc/rc.d/init.d/httpd restart. Avataan www-selain, ja kirjoitetaan osoiteriville http://koneennimi/nagios tai vaihtoehtoisesti Nagios-koneen IP-osoite/nagios. Mikäli käytetään jotain muuta kuin oletusporttia 80, lisätään se myös osoitteeseen. Nagioksen aloitussivun pitäisi aueta selaimeen (Kuva 2) .



Kuva 2 Nagioksen aloitussivu www-selaimessa

4.3.3 Autentikointi Cgi:lle

Cgi-tiedostoille tarvitsee määritellä autentikoidut käyttäjät, joilla on oikeus katsella tiedostojen sisältöä. Tarkoituksena on, että kun avataan Nagios www-selaimessa, aukeaa ensin käyttäjätunnuksen ja salasanan kyselyikkuna. Näin estetään luvattomien käyttäjien pääsy tutkimaan Nagioksen sisältöä.

Tarkistetaan, että Apachen konfiguroinnissa on cgi-hakemistoille ja html-hakemistoille määritelty kohta ”AllowOverride AuthConfig” skriptialias- ja alias-kohtien määrittelyssä. Luodaan tiedosto .htaccess sbin-hakemiston ja share-hakemiston juureen (/usr/local/nagios/sbin sekä /usr/local/na-gios/share). Kirjoitetaan tekstieditorilla tiedostoihin

seuraava sisältö:

AuthName "Nagios Access"
AuthType Basic
AuthUserFile /usr/local/nagios/etc/htpasswd.users
require valid-user

Luodaan autentikoituja käyttäjiä kirjoittamalla komentotulkkiin kehoite htpasswd -c /usr/local/nagios/etc/htpass-wd.users nagiosadmin . Komento kysyy käyttäjälle salasanaa, luo tiedoston htpasswd.users hakemistoon /usr/local/nagios/etc ja lisää tiedostoon käyttäjän nagiosadmin ja sille annetun salasanan kryptattuna. Jos halutaan lisätä uusia autentikoituja käyttäjiä Nagiokselle, kirjoitetaan komentotulkkiin kehoite htpasswd /usr/local/nagios/etc/ht-passwd.users <username>, missä <username> on haluttu käyttäjätunnus.

Autentikoiduille käyttäjille annetaan oikeuksia cgi-tiedostoihin cgi.cfg-konfiguraatiotiedostossa. Määritetään käyttäjätunnuksen ja salasanan kysely päälle kohdassa use_authentication=<0> <1>, missä 0 tarkoittaa autentikoinnin pois kytkemistä ja 1 sen olevan päällä. Default User-kohdassa määritellään käyttäjä, jolla on oikeudet tutkia Nagioksen sivuja www-selaimessa ilman autentikointia. Turvallisuuden takaamiseksi kenellekään käyttäjälle ei kannata myöntää pääsyä sivuille ilman käyttäjätunnuksen ja salasanan kyselyä.

Turvallisuuden hallinnan kannalta on tärkeää, että vain valikoidut käyttäjät voivat tutkia Nagioksen antamia tietoja, ja varsinkin konfiguraatiotiedostojen sisältöä. Hyvänä ohjeena voidaan pitää, että vain järjestelmän pääkäyttäjällä on oikeus kaikkiin tietoihin, ja muille käyttäjille annetaan mahdollisimman rajoitettu pääsy Nagioksen antamiin tietoihin. Jos halutaan käyttäjän pääsevän katsomaan vain laitteiden ja palveluiden tilaa, lisätään käyttäjätunnus kohtaan authorized_for_all_services sekä kohtaan authorized_for_all_hosts.

4.3.4 Nagioksen konfigurointi

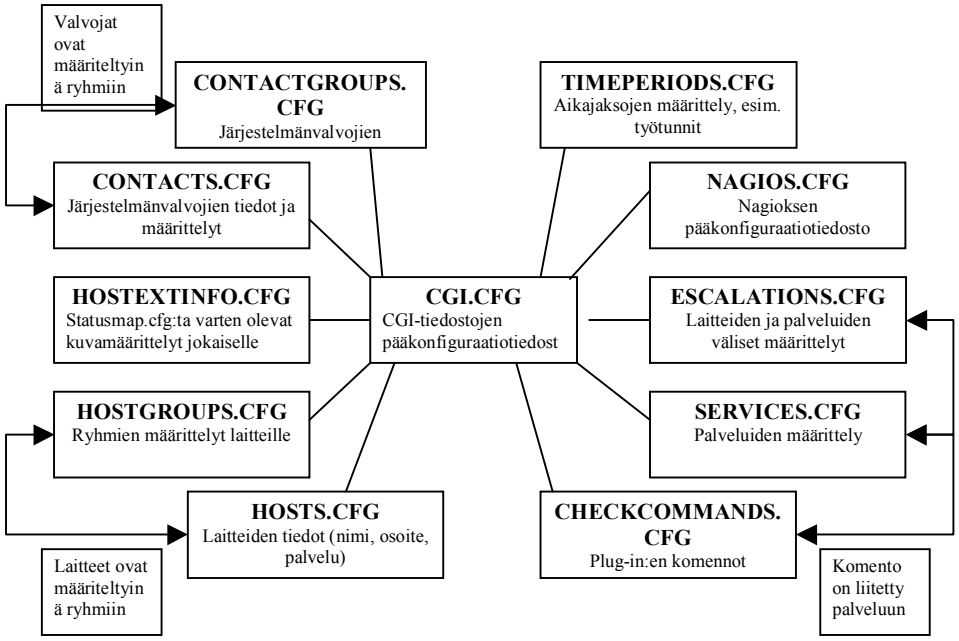
Asennuksen aikana Nagios luo hakemistoon /usr/local/nagios/etc esimerkkikonfiguraatiotiedostoja, jotka ovat muotoa cgi.cfg-sample. Näistä tiedostoista kannattaa ottaa varmuuskopio omaan hakemistoon. Varmuuskopioinnin jälkeen käyttöön tulevien konfiguraatiotiedostojen päätteeksi täytyy muuttaa .cfg, jolloin ne ovat muotoa cgi.cfg. Varmuuskopiot voidaan jättää muotoon .cgi-sample. Kaikki tiedostot ovat hyvin kommentoituja, ja lukemalla niitä tekstieditorissa saa selville mitä mikin komento tarkoittaa ja mihin tiedosto vaikuttaa. Linuxin tapaan kommenttirivit on aloitettu #-merkillä.

Nagioksen pääkonfiguraatiotiedosto on /usr/local/nagios/etc/nagios.cfg. Se sisältää määrityksiä kuinka Nagios toimii. Tätä konfiguraatiotiedostoa käsittelee niin Nagioksen ydinohjelma kuin cgi:tkin.

CGI:den konfiguraatiotiedosto on cgi.cfg. Loput tiedostot ovat objektien konfiguraatiotiedostoja, joihin määritellään esimerkiksi monitoroitavat koneet, palvelut, ryhmät, yhteydet, yhteysryhmät ja komennot. Näihin tiedostoihin määritellään mitä verkosta halutaan hallinnoida ja kuinka se tehdään.

4.3.5 Konfigurointitiedostojen väliset suhteet

Nagioksen konfigurointitiedostot ovat riippuvaisia toisistaan ja toisissa tiedostoissa määritellyistä muuttujista. Konfiguraatiot voidaan tehdä monella eri tavalla, riippuen esimerkiksi verkon rakenteesta, laitteista sekä hallinnoitavista palveluista. Kuva 3 esittää, kuinka tässä työssä tiedostot liittyvät toisiinsa.



Kuva 3 Konfiguraatiotiedostojen väliset suhteet

4.4 Nagioksen käynnistäminen

Kun kaikki konfiguraatiot on saatu valmiiksi, kannattaa Nagios käynnistää -v-optiolla. Näin tehdään vielä lopputarkastus ennen palvelun käynnistämistä, ja käydään läpi rivi riviltä löytyykö jostain virheitä.

Nagios suorittaa ennen tavallistakin käynnistymistään tarkistuksen, eikä lähde käyntiin mikäli jossain konfiguraatiotiedostossa on virheitä tai vääriä viittauksia. Kirjoitetaan komentotulkkiin rivi /usr/local/nagios/bin/nagios/ -v usr/local/nagios/etc/nagi-os.cfg, missä /usr/local/nagios/bin/nagios on Nagioksen ydinohjelman sijaintipaikka ja /usr/local/nagios/etc/nagios.cfg Nagioksen pääkonfiguraation hakemisto. Mikäli virheitä löytyy, tulee ilmoitus missä hakemistossa ja millä rivillä virhe on. Näin virheen paikallistaminen on helppoa avaamalla tiedosto tekstieditoriin ja etsimällä oikea rivi.

Kun lopputarkistus on tehty ilman virheilmoituksia, on aika käynnistää Nagios. Nagiosta voidaan käyttää neljällä tapaa: manuaalisesti etualan prosessina, mikä sopii testausajoon ja virheidenetsintään, manuaalisesti taustaprosessina, manuaalisesti demonina tai automaattisesti järjestelmän käynnistyessä.

Käynnistämisen jälkeen Nagios voidaan sammuttaa tai käynnistää uudelleen komentotulkissa tai www-selaimen kautta.Jos Nagios on määritelty toimimaan taustaprosessina asennuksen aikana annetulla komennolla make install-init, voidaan Nagios käynnistää, sammuttaa ja käynnistää uudelleen komennoilla /etc/rc.d/init.d/nagios <start>, <stop> tai <restart>.

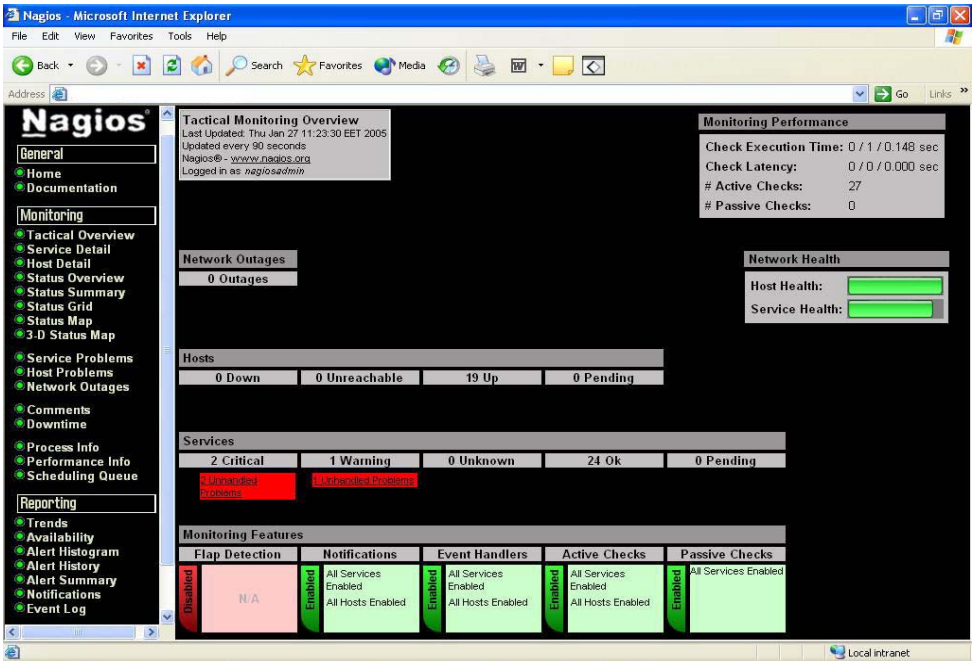
4.5 Nagioksen www-käyttöliittymä

Nagiosta on yksinkertaista käyttää www-selaimen avulla. Avataan selain, ja kirjoitetaan osoiteriville koneen nimi tai IP-osoite ja /nagios. Esimerkiksi http://10.10.10.10/nagios. Mikäli Apache on konfiguroitu käyttämään jotain muuta porttia kuin oletuksena olevaa porttia 80, lisätään portin numero heti IP-osoitteen perään kaksoispisteen jälkeen, esimerkiksi http://10.10.10.10:8000/nagios.

Aloitussivulta nähdään mikä versio Nagioksesta on asennettuna, sekä linkit dokumentaatiosivulle että Nagioksen kotisivuille osoitteseen www.nagios.org. Dokumentaatio on tallennettuna paikalliselle koneelle asennuksen yhteydessä hakemistoon /usr/local/nagios/share. Sitä pääsee lukemaan myös vasemman reunan Documentation-linkistä.

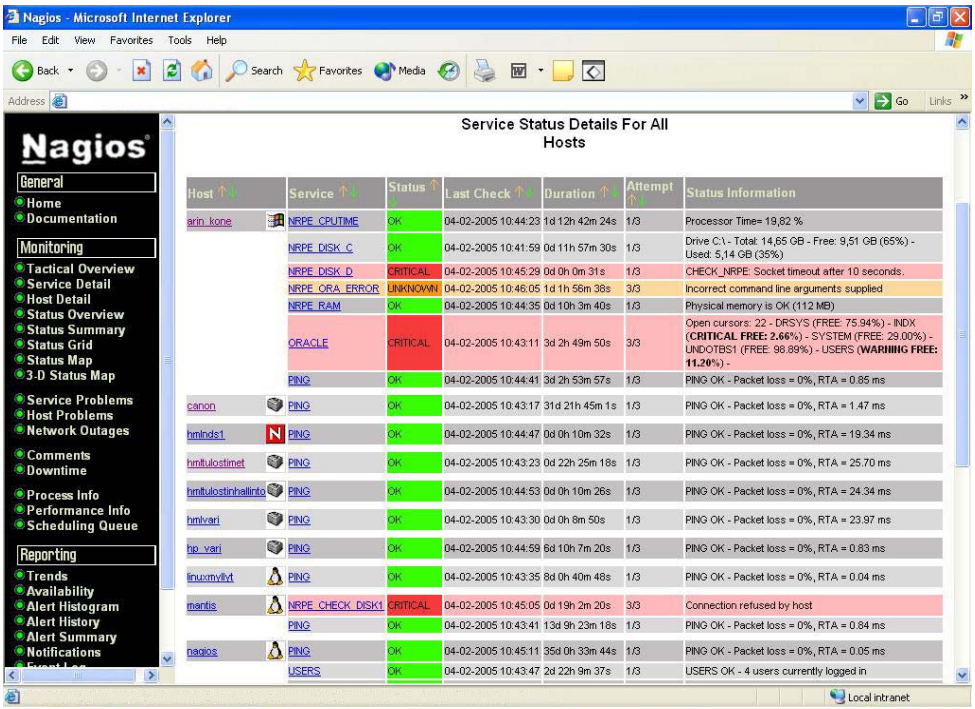
Monitoring-osio sisältää linkit verkon laitteiden monitorointiin ja hallintaan. Tactical Overwiev-linkki antaa yleiskatsauksen verkon tilaan, yhdellä silmäyksellä voi tarkistaa onko jokin asema alhaalla vai tavoittamattomissa, ja mitkä ovat palvelujen tarkistusten tilat.

Kuvassa 4 näkyy kuinka monta laitetta on kriittisessä tilassa tai toimii oikein sekä mikä on palvelujen tila.



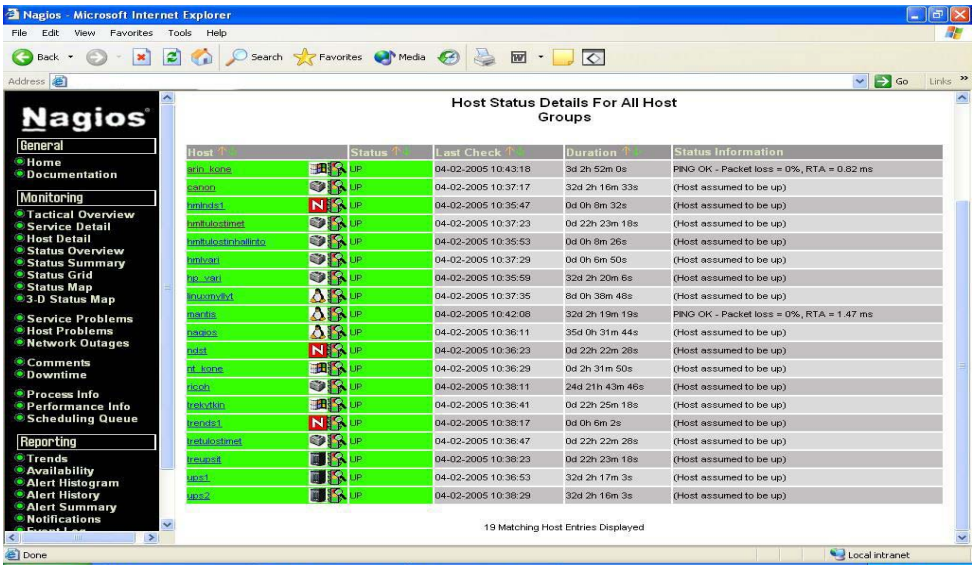
Kuva 4 Tactical Overview

Service Detail näyttää mitä palveluja on määritelty millekkin laitteelle, sekä palvelujen tarkistusten tilan. Kuvassa 5 näkyy, että esimerkiksi konelle nimeltään arin_kone on määriteltynä 7 palvelua, ja laitteelle nimeltään canon on määriteltynä 1 palvelu.



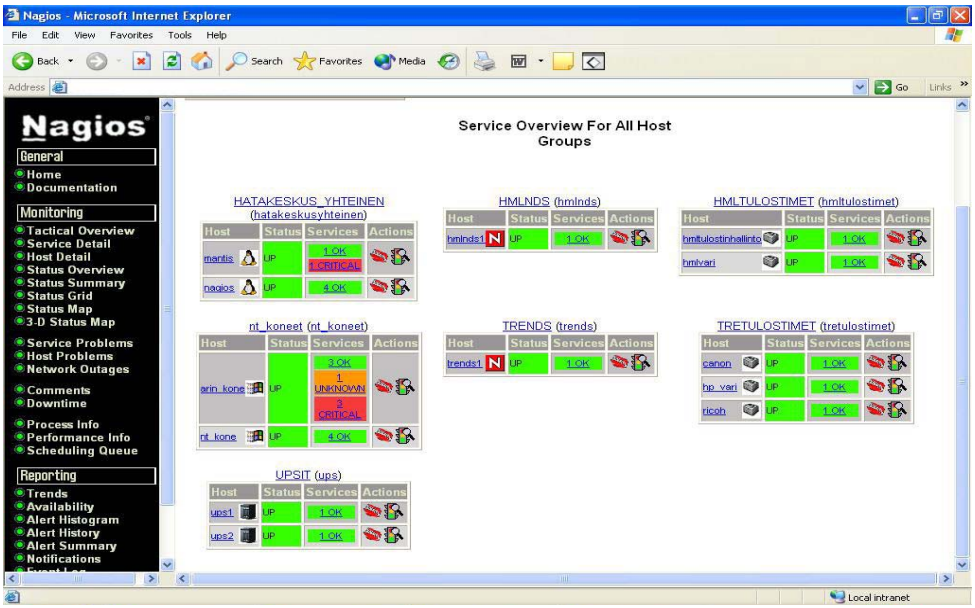
Kuva 5 Service Detail

Host Detail-linkistä voi katsoa hosts.cfg-tiedostoon konfiguroidun ensisijaisen palvelun tilan, tässä työssä kaikkiin tehdään ensisijaisesti check_ping-tarkistus (Kuva 6).



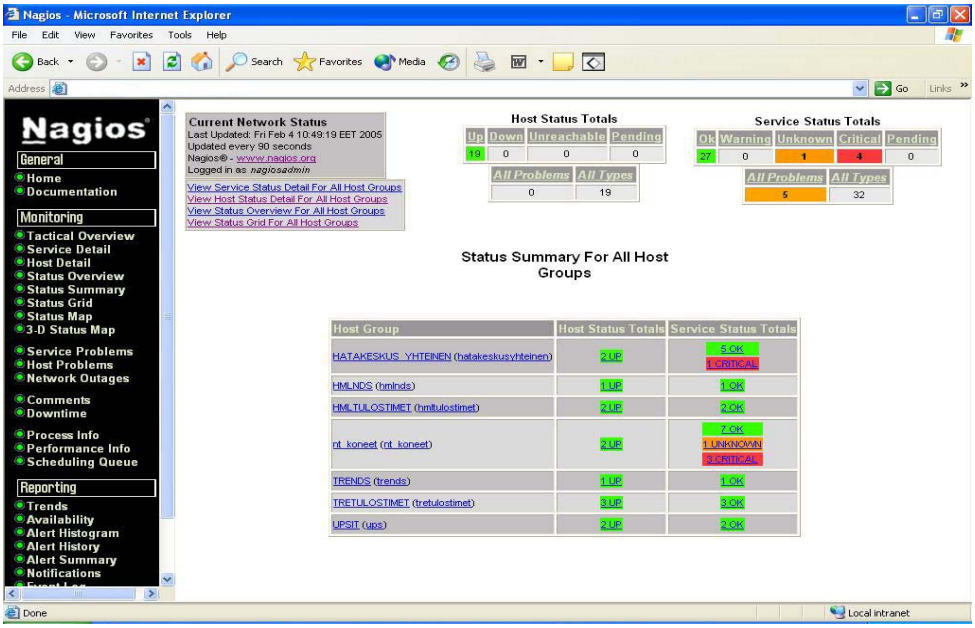
Kuva 6 Host Detail

Status Overview-kohdasta näkee asemat ryhmiteltyinä sekä palveluiden tilan. Nagios käyttää värejä ilmaistessaan palvelujen ja laitteiden tilaa. Vihreä tarkoittaa kaiken olevan kunnossa, keltainen on varoitusväri, oranssi ilmoittaa että konfiguroinnissa on virheitä, ja punainen ilmaisee kriittistä tilaa. Kuvassa 7 nähdään yhdellä silmäyksellä, että mantis-koneella palvelu on kriittisessä tilassa, kuten myös laitteella arin_kone yksi palvelu. Lisäksi arin_koneella on oranssilla värillä ilmaistu yksi tuntemattomassa tilassa oleva palvelu.



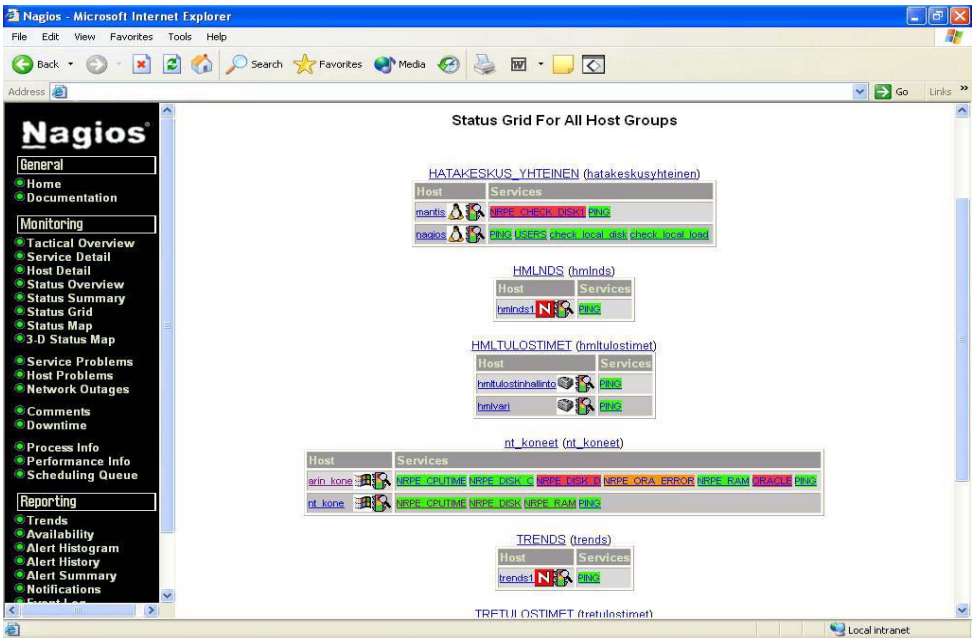
Kuva 7 Status Overview

Status Summary näyttää yhteenvedon ryhmistä ja ryhmiin kuuluvien laitteiden tilasta. Jokaisessa näkymässä laitteen ja palvelun nimi on linkkinä, jota napsauttamalla pääsee tutkimaan yksityiskohtaisempia tietoja (Kuva 8).



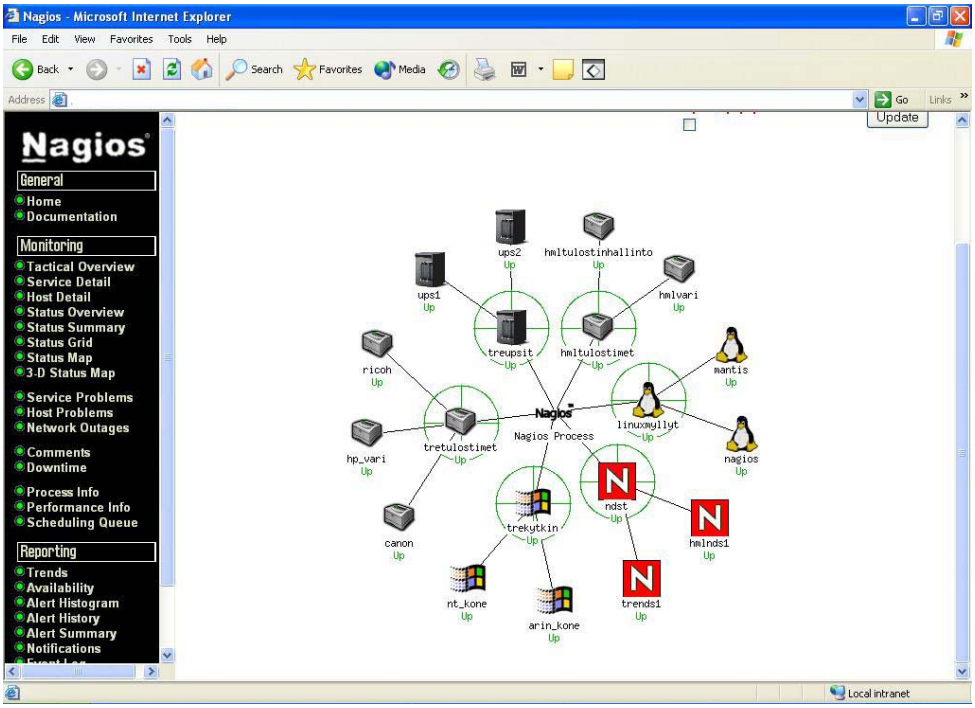
Kuva 8 Status Summary

Status Grid-linkistä näkee laitteet ryhmiteltyinä kuten Status Summary-linkistäkin, mutta Status Grid näyttää jokaiseen ryhmään kuuluvan laitteenkin tiedot erikseen (Kuva 9).



Kuva 9 Status Grid

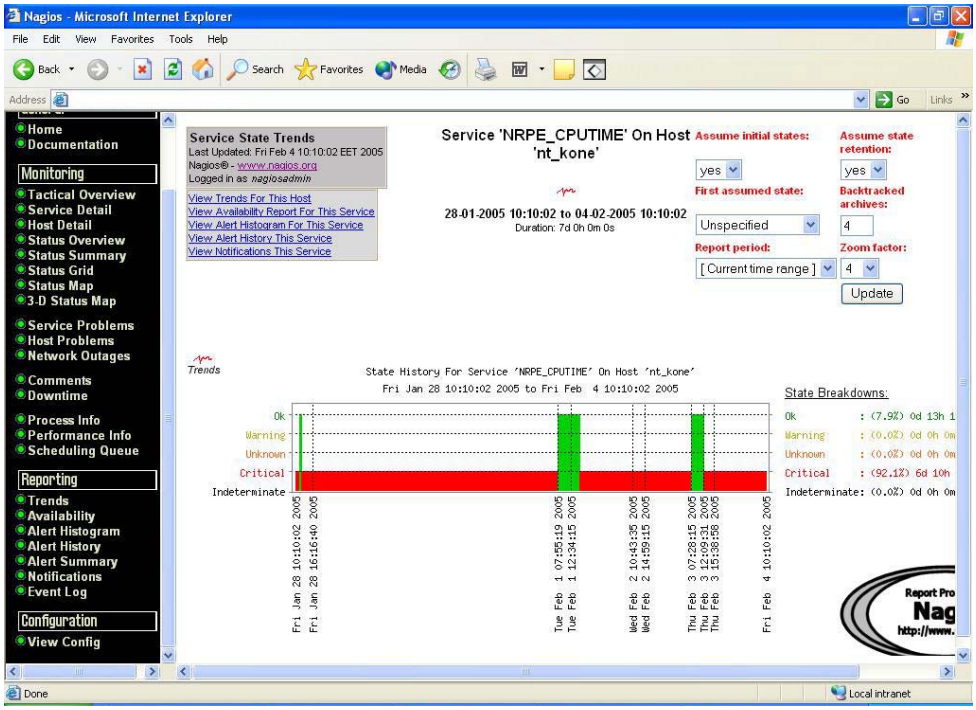
Status Map esittää graafisesti verkon kuvan. Drawing Layers-valikosta voidaan valita kuvan esitystapa. Kun viedään hiiri laitteen nimen päälle, ilmestyy pop-up, joka kertoo tietoja laitteesta ja sen tilasta. Myös Status Map:ssa ovat käytössä värit, joiden perusteella voidaan helposti nähdä huomiota vaativat laitteet tai palvelut (Kuva 10).



Kuva 10 Status Map

Service Problems ja Host Problems esittävät minkälaisia ongelmia esiintyy koneilla tai palveluilla. Comments-kohdassa käyttäjät voivat kirjoittaa huomioitaan tai kommenttejaan muiden käyttäjien luettavaksi. Downtime-linkistä voidaan ajastaa jokin laite tai palvelu pysäytetyksi halutuksi aikaa. Process Info kertoo Nagioksesta ja sen tilasta, ja sieltä voidaan sammuttaa ja käynnistää uudelleen Nagioksen prosessi. Performance Info näyttää yhteenvedon kaikista määritellyistä aktiivisista ja passiivisista tarkistuksista. Aktiivinen tarkistus suoritetaan tietyin väliajoin automaattisesti, passiivinen tarkistus vain käyttäjän toimesta. Scheduling Queue esittää koska viimeisin tarkistus laitteelle tai palvelulle on tehty, ja milloin seuraava tarkistus tehdään.

Reporting-kohdan linkeistä voidaan tutkia erilaisia tilastoja ja logeja ja luoda raportteja eri laitteista ja palveluista (Kuva 11).



Kuva 11 Raportti palvelun toiminnasta

Configuration-kohdan linkistä View Config käyttäjä voi katsoa www-selaimessa Nagioksen konfiguraatiotiedostoja. Tiedostojen sisältämän tiedon takia on tärkeää, etteivät asiattomat pääse käsiksi niihin.

Kokonaisuudessaan Nagioksen www-käyttöliittymä on yksinkertainen käyttää. Se on hyvin informatiivinen ja antaa yhdellä silmäyksellä tiedon verkon tilasta. Värien käyttö on tehokas tapa ilmaista laitteen tai palvelun asema. Häiritsevästi puutteellinen asia tietoturvan kannalta on uloskirjautuminen. Käyttöliittymässä ei ole erillistä uloskirjautumiskohtaa. Mikäli halutaan kirjautua sisään jollain toisilla tunnuksilla, se voidaan suorittaa vain sulkemalla selain ja avaamalla se uudestaan.

4.6 Nagioksen perus-plug-in:t

Nagioksen asennuksen yhteydessä ladattiin ja purettiin /usr/local/nagios/libexec-hakemistoon joukko verkon valvonnassa tarpeellisia plug-in:a. Plug-in:lle on valmiiksi määritelty perusasetukset tiedostossa checkcommands.cfg. Lisäparametrit määritellään service.cfg-tiedostoon, kun määritellään mitä laitetta kutsutaan milläkin palvelulla. Helpoiten plug-in:n toimintaperiaatteen saa selville kirjoittamalla komentotulkkiin plug-in:n nimi ja attribuutti –h, esimerkiksi check_http –h. Huomioitavaa on, että sijaitaan oikeassa hakemistos-

sa, /usr/local/na-gios/libexec.

Tarpeellisimpia plug-in:ja ainakin aluksi ovat esimerkiksi check_disk, check_dns, check_ftp, check_http, check_load, check_ping, check_tcp ja check_users.

Check_disk-plug-in palauttaa levyn käytetyn tilan prosentteina, ja generoi varoituksen mikäli prosenttiluku ylittää määritellyn arvon. Optiolla –w *kokonaisluku* % määritellään varoitusviestin aiheuttava levyn käyttöaste prosentteina, ja optiolla –c *kokonaisluku* % kriittisen viestin aiheuttava levyn käyttöaste prosentteina. Optiot –w ja –c ovat pakollisia.

Check_dns-plug-in käyttää nslookup-kyselyä hakeakseen annetulle domain-nimelle sitä vastaavan IP-osoitteen. DNS-palvelimen osoite voidaan määritellä –s -optiolla. Mikäli DNS-palvelinta ei määritellä, plug-in käyttää /etc/re-solv.conf –tiedostoon määriteltyä oletuspalvelinta. Optiolla –H annetaan se nimi tai osoite mitä halutaan kysellä. Osoite on pakollinen tieto kyselyä varten.

Jos halutaan testata FTP-yhteyttä tiettyyn asemaan, voidaan käyttää check_ftp-plug-in:a. Optiot –H ja –p ovat pakollisia. Optio –H määrittelee aseman osoitteen, ja optio –p portin, mitä ftp-yhteys käyttää. Lisäksi voidaan antaa optioita, joilla määritellään rajat varoitus- ja kriittisyysviestejä varten.

Check_http-plug-in testaa tietyn aseman http-palvelun. Sillä voidaan testata niin http- kuin https-palvelimia, tehdä edelleenohjauksia, etsiä merkkijonoja ja tavallisia ilmauksia, tarkistaa yhteydenottoaikoja ja raportoida sertifikaatin päättymisajasta. Pakollisia optioita check_http-plug-in:lle ovat joko –H, millä annetaan laitteen domain-nimi, tai –I, millä määritellään IP-osoite. –s –optiolla annetaan haettava merkkijono, -p –optio määrittelee käytettävän portin, mikäli se ei ole oletuksena käytettävä 80. –w- ja –c –optioilla annetaan varoitusviestejä varten raja-arvot sekunteina. Mikäli sivuilla vaaditaan kirjautumista, voidaan käyttäjätunnus ja salasana määritellä –a –optiolla muodossa käyttäjätunnus:salasana. Optio –S luo yhteyden käyttämällä ssl-tekniikkaa, ja optiolla –C määritellään kokonaislukuna kuinka monta päivää sertifikaatin tulee minimissään olla voimassa.

Paikallisen koneen kuormaa saadaan valvottua check_load- plug-in:lla. Se testaa koneen keskimääräistä kuormitusta, ja palauttaa tiedon mikäli –w- ja –c –optioihin annetut arvot ylittyvät.

Yleisin tarvittavista plug-in:sta on check_ping. Se lähettää ICMP ECHO-paketteja määritellyille laitteille, ja palauttaa hävinneiden pakettien määrän sekä keskimääräisen kulutetun ajan millisekunneissa. Ping-komennolla testataan onko verkossa oleva laite ”hereillä”, eli vastaako se lähetettyihin paketteihin. Pakollisia optioita check_ping-plug-in:lle ovat –H, -w ja –c. –H –optioon määritellään

pingattavan aseman nimi tai IP-osoite. –w- ja -c –optioihin liitetään arvot rta ja pl%. Rta tarkoittaa viestin kiertoajalle määriteltyä keskiarvoa, jonka ylityttyä plug-in generoi varoitus- tai kriittisyysviestin, ja pl% on kadonneiden pakettien maksimiprosenttimäärä, minkä ylityttyä annetaan varoitus. Esimerkiksi:

check_ping –H 10.10.10.1 –w 3000.0, 80% -c 5000.0, 100% -p 1

Optiolla –p määritellään kokonaislukuna kuinka monta ICMP ECHO-pakettia lähetetään. Oletuksena luku on viisi. Optiolla –t (timeout) voidaan antaa kokonaisluku sekunteina, minkä jälkeen komento ei lähetä paketteja. Oletuksena timeout on 10 sekuntia.

Tcp-yhteyksien testaamista varten on check_tcp-plug-in. Sen pakollisia optioita ovat –H testattavan aseman osoitetta varten sekä –p, millä määritellään portin numero. Optioilla –w ja –c voidaan antaa varoitus – sekä kriittisyysarvot sekunteina vastausaikoja varten.

Paikallisen koneen käyttäjämääriä voidaan seurata check_users-plug-in:lla. Se tarkkailee koneelle kirjautuneiden käyttäjien määrää ja generoi virheestä, mikäli annetut rajat ylittyvät. Optiolla –w määritellään kokonaislukuna varoitusviestin aiheuttava käyttäjämäärä, ja optiolla –c kriittisyysviestin aiheuttava käyttäjien määrä.

4.7 NRPE

NRPE tulee sanoista Nagios Remote Plugin Executor, eli se nimensä mukaisesti hakee tietoa valvottavilta koneilta erilaisten plug-in:en avulla. Jos Nagiosta halutaan käyttää verkossa olevien Windows-koneiden valvontaan, tarvitaan nrpe_nt-nimistä agenttia. NRPE:n toiminta Windows-ympäristössä koostuu useasta osasta. Check_nrpe-plug-in asennetaan Nagios-koneelle, nrpe_nt-agenttia ajetaan Windows-koneella taustaprosessina, sekä Windows-koneelle asennetaan plug-in:ja nrpe_nt-agenttia varten. Check_nrpe on plugin, mitä ajetaan Nagios-koneella muiden plug-in:en tapaan. Se ottaa yhteyttä Windows-koneilla käynnissä olevaan nrpe-prosessiin ja palauttaa agentin hakeman tiedon Windows-koneen plug-in:lta Nagios-koneelle.

NRPE:n avulla voidaan tutkia Windows-koneesta esimerkiksi levyn käyttöä, prosessorin kulutusta sekä muistin määrän käyttöä. Agentti palauttaa Nagios-asemalle kyselyn tiedon, esimerkiksi levyn vapaan ja käytetyn tilan määrän prosentteina. Komennolle pitää asettaa tietyt varoitus- tai kriittisyysarvot, joiden ylityttyä annetaan Nagioksen käyttöliittymässä viesti tapahtuneesta.

4.7.1 NRPE_NT:n asennus Windows-koneelle

Ladataan Windows-koneelle nrpe_nt-paketti osoitteesta http://www.miwi-dv.com/nrpent/. Jos halutaan käyttää ssl-suojattua yhteyttä nrpe:n kanssa, valitaan paketti mikä sisältää ssl:n vaatimat kirjastot. Puretaan paketti sopivaan kansioon, tässä työssä on valittu kansio D:\nrpe. Vähimmäisvaatimuksena nrpe:n toiminnalle ovat tiedostot nrpe_nt.exe sekä konfiguraatiotiedosto nrpe.cfg. Paketin mukana tulee testiajoa varten tiedosto test.cmd, mikä palauttaa Nagios-koneelle viestin hallo from cmd.

Ladataan Windows-koneelle haluamasi plug-in:t, riippuen siitä mitä halutaan koneelta valvoa. Hyvä valikoima NRPE-plug-in:ja löytyy osoitteesta http://www.nagiosexchange.org/NRPE_Plugins.66.0.html Chkwin_cputime palauttaa prosessorin käyttöasteen prosentteina, check_disk palauttaa halutun levyosion koon, sekä vapaan ja käytetyn tilan joko megatavuina tai gigatavuina sekä prosentteina. Check_ram palauttaa fyysisen muistin käytettävissä olevan vapaan määrän. Puretaan paketit samaan tiedostoon nrpe_nt-paketin kanssa, tässä tapauksessa D:\nrpe\bin. Jokaisen plug-in:n mukana tulee readme-tiedosto, mikä kannattaa lukea lisäohjeiden saamiseksi.

Avataan nrpe_nt:n konfigurointitiedosto nrpe.cfg muistioon ja tehdään tarvittavat muutokset. Oletusporttina käytetään porttia 5666, mitä ei kannata vaihtaa. Palvelimen osoitetta ei tarvitse erikseen kertoa. Nagios-koneen IP-osoite lisätään kohtaan ”allowed_hosts”. Komennot, joita halutaan ajaa, lisätään kohtaan command definitions. Plug-in:en mukana tulevissa readme-tiedostoissa on lisänä komentorivi command[check_komento], mikä voidaan kopioida nrpe.cfg-tiedostoon. Kannattaa tarkistaa, että tiedostopolut sekä argumentit ovat oikein. Esimerkkinä muistin tilan tarkistamisen suorittava komento:

command[check_ram]=c:\WINDOWS\system32\cscript.exe //NoLogo //T:10 D:\NRPE\bin\check_ram.wsf /w:20 /c:10

Ensin kerrotaan mistä löytyy Windowsin tiedosto, jonka avulla tarkistus voidaan suorittaa. T tarkoittaa timeout-argumenttia, mikä on hyvä määritellä, ettei palvelu jää ikuisesti hakemaan tietoa vaikkei sitä löydy. Sitten kerrotaan missä hakemistossa sijaitsee check_ram-plugin, mikä suorittaa pyydettäessä halutun komennon. W tulee sanasta warning, c sanasta critical. Ne ovat raja-arvoja prosenttilukuina, missä tapauksessa annetaan varoitus ja milloin järjestelmän tila on kriittinen.

Avataan komentokehoite, ja siirrytään oikeaan hakemistoon, mikä sisältää nrpe_nt.exe-tiedoston (tässä D:\nrpe\bin). Kirjoitetaan komento nrpe_nt -i. Komento asentaa nrpe_nt-agentin palveluksi, mikä pyörii taustalla Windowsin ollessa käynnissä. Tutkimalla käynnissä olevia palveluita, pitäisi löytyä palvelu nimeltään "Nagios Remote Plugin Executor for NT/W2K". Palvelun saa käynnistettyä hallintapa-

neelista, mmc:sta tai kirjoittamalla komentokehotteeseen "net start nrpe_nt".

Kun nrpe_nt on saatu käynnistettyä, siirrytään Nagios-koneen puolelle asentamaan check_nrpe-plug-in:a.

4.7.2 Nagios-koneen konfigurointi nrpe:ta varten

Check_nrpe-plugin ei sisälly Nagioksen perus-plug-in-pakettiin. Ladataan Nagios-koneelle check_nrpe-plugin, puretaan paketti ja ajetaan konfigurointiskripti perusmuodossa ./configure. Check_nrpe-plugin asentuu muiden Nagioksen plugin:en kanssa /usr/local/nagios/libexec –hakemistoon. Muokataan escalations.cfg, checkcommands.cfg ja services.cfg -konfiguraatiotiedostoja seuraavasti:

Checkcommands.cfg:hen lisätään jokaista nrpe:n käyttämää komentoa varten oma määrittely. Jos käytössä ovat muistin, levyn tilan sekä prosessorin kuorman tarkistukset, lisätään seuraavat rivit tiedostoon:

```
#check_disk
define command{
    command_name        check_disk
    command_line        /usr/local/nagios/libexec/check_nrpe    H
                $HOSTADDRESS$ -c check_disk
}

#check_cputime
define command{
    command_name        check_cputime
    command_line        /usr/local/nagios/libexec/check_nrpe    H
                $HOSTADDRESS$ -c check_cputime
}

#check_ram
define command{
    command_name        check_ram
    command_line        /usr/local/nagios/libexec/check_nrpe    H
                $HOSTADDRESS$ -c check_ram
}
```

Check_nrpe –plugin:lla tehdään kysely koneelle, missä ajetaan nrpe-agenttia. Argumentilla –c check_komento määritellään mitä halutaan nrpe-koneelta palauttaa.

Services.cfg-tiedostoon määritellään mitä palveluita halutaan millekin koneelle suorittaa. Lisätään halutulle nrpe-koneelle palvelu, jonka kuvaus (service description) on sama, mikä määritellään escalati-

ons.cfg-tiedostoon. Esimerkkinä rivit, joilla määritellään nrpe-koneelle palvelu, jolla tarkistetaan muistin tila:

```
# Service definition
define service {
    use                     generic-service
    host_name               XP-Kone
    service_description     NRPE_RAM
    .
    .
    .
    check_command           check_ram
    }
```

check_ram viittaa checkcommands.cfg –tiedostossa määriteltyyn komentoon, millä tarkastetaan nrpe-koneen muistin tila. Lisätään samanlaiset rivit kaikille palveluille, joita halutaan käyttää. Vaihdetaan host_name, service_descrption ja check_command palvelua vastaavaksi.

Escalations.cfg –tiedostoon lisätään määrittelyt service-escalation-kohtaan. Siihen viitataan services.cfg-tiedostossa kohdassa service_description. Esimerkkinä check-_ram-palvelun määrittely:

```
#Serviceescalation definition
define serviceescalation {
    host_name               XP-Kone
    service_description     NRPE_RAM
    .
    .
    .}
```

Luodaan kaikkia palveluita varten samanlainen määrittely. Muokataan kohdat host_name ja service_description palvelua vastaavaksi.

4.7.3 Linux-koneen valvonta NRPE:lla

NRPE:n avulla voidaan valvoa myös Linux-koneita. Toimintaperiaate on samanlainen kuin Windows-ympäristössä, tietysti joitain erilaisuuksia löytyy johtuen erilaisista alustoista. Linux-koneelle ladataan nrpe-agentti sekä tarvittavat plug-in:t, ja agenttia kutsutaan Nagioksen puolella check_nrpe-plug-in:lla. Valvottavalle koneelle voidaan asentaa Nagioksen perus-plug-in-paketti, jota käytetään yleensä paikallisen koneen valvontaan. NRPE:n avulla näitä paikallisia tarkistuksia voidaan kuitenkin kutsua ja niiden antamaa tietoa esittää Nagioksen käyttöliittymässä.

Ladataan NRPE-paketti Linux-koneelle osoitteesta www.nagiosexchange.org/NRPE.77.0.html . Valitaan versio, jonka numero on pie-

nempi kuin 2.0, sillä versiot 2.0 ylöspäin eivät ole yhteensopivia vanhempien versioiden kanssa. Ladataan Nagioksen perus-plug-in:t osoitteesta www.nagios.org/download, ja valitaan niistä esimerkiksi versio 1.3. Puretaan paketit /tmp –hakemistoon ja ajetaan konfigurointiskirptit käskyllä ./configure, sekä käännetään binäärit komennolla make all. Binäärit sijoittuvat /src –hakemistoon, ja ne pitää manuaalisesti siirtää haluttuun paikkaan. Luodaan hakemisto mihin halutaan sijoittaa NRPE:n ja plug-in:t, esimerkiksi /usr/local/nagios. Kopioidaan tähän hakemistoon binäärit, sekä luodaan alihakemisto /libexec plug-in:ja varten.

Jos halutaan ajaa NRPE:ta demonina inetd:n tai xinetd:n alla, tarvitsee niiden konfiguraatiotiedostoihin tehdä lisäyksiä. Lisätään /etc/services-tiedostoon rivi, jossa määritellään mitä porttia NRPE käyttää:

```
nrpe            5666/tcp    #NRPE
```

Mikäli järjestelmä käyttää inetd:ta, lisätään seuraava rivi /etc/inetd.conf-tiedostoon:

```
nrpe stream tcp nowait <user> /usr/sbin/tcpd <nrpebin> -c <nrpecfg> --inetd
```

Korvataan <user> käyttäjätunnuksella, jolla NRPE-palvelu ajetaan, esimerkkinä Nagios. Korvataan <nrpebin> binäärien sijaintipaikalla ja <nrpecfg> NRPE:n konfiguraatiotiedoston osoitteella.

Jos järjestelmä käyttää xinetd:ta inetd:n sijaan, luodaan tiedosto nimeltään nrpe /etc/xinetd.d-hakemistoon. Tiedoston sisällöksi tulee seuraavat rivit:

```
#default:on
#description: NRPE
service nrpe
{
    flags                   =  REUSE
    socket_type             = stream
    wait                    = no
    user                    = <user>
    server                  = <nrpebin>
    server_args             = -c <nrpecfg> --inetd
    log_on_failure          += USERID
    disable                 = no
    only_from               =<ipaddress1> <ipaddress2>..
```

Korvataan <user> nrpe-palvelun käyttäjän nimellä, <nrpebin> nrpe:n binäärien hakemistolla, <nrpecfg> konfiguraatiotiedoston hakemistolla ja <ipaddress> sen koneen IP-osoitteella, millä on oikeus ottaa yhteyttä NRPE demoniin. Käynnistetään lopuksi inetd- tai xinetd-palvelu uudelleen.

Konfiguroidaan Nagios-koneella checkcommands.cfg, escalations.cfg ja services.cfg –tiedostot palvelua vastaaviksi.

4.8 Nagioksen vaatimat kirjastot

Nagios toimii hyvin ilman erikseen asennettuja kirjastoja (libraries), mutta jos halutaan tutkia graafisia esityksiä statusmap-, trends- ja histogram-linkeistä, täytyy koneelle hakea ja asentaa kuvien katsomista varten tehtyjä erityisiä kirjastoja.

Jos ajetaan konfigurointiskripti ./configure Nagioksen asennuksen aikana, eikä kirjastoja ole asennettuna, ilmoittaa skripti että GD-kirjastoja ei löydy. Nagios voidaan silti asentaa ja konfiguroida normaalisti. Kirjastojen puuttuminen estää statusmap.cgi:n, trends.cgi:n ja histogram.cgi:n asentumisen /usr/local/nagios/sbin–hakemistoon. Kun napsautetaan hiirellä www-selaimessa statusmap-, trends-, tai histogram–linkkejä, tulee ilmoitus ettei sen nimisiä tiedostoja löydy. Syynä on se, että niitä ei ole asennettu tiettyjen kirjastojen puutteesta johtuen.

Graafiset cgi:t vaativat toimiakseen vähintään gd- , jpeg-, sekä png-kirjastot. GD tulee sanoista graphics library. Sen avulla voidaan näytöllä esittää .jpeg- sekä .png–päätteisiä kuvia. GD-kirjasto tarvitsee toimiakseen asennetut jpeg- ja png-kirjastot.

Aloitetaan kirjastojen lataaminen ja asennus jpeg- ja png-kirjastoilla, koska GD-kirjaston asennus etsii onko näitä kirjastoja asennettuna koneelle. Ladataan jpeg-kirjasto osoitteesta www.ijg.org ja png-kirjasto osoitteesta www.libpng.org/pub/png/ /tmp –hakemistoon. Puretaan molemmat paketit gunzip ja tar xf – komennoilla.

Ajetaan konfigurointiskripti png-kirjastolle siirtymällä oikeaan hakemistoon komennolla cd /tmp/libpng-1.2.8-config ja kirjoittamalla komentotulkkiin ./configure (ilman mitään attribuutteja) sekä lopuksi make install. Png-kirjasto asentuu automaattisesti oikeaan paikkaan ja on nyt toimintakunnossa.

Jpeg-kirjaston konfigurointiskriptiin lisätään attribuuteiksi oikea hakemisto, eli skripti on muotoa ./configure –pre-fix=/usr/local. Kirjoitetaan tämän jälkeen komennot make, make install, make-install-lib, make install-headers sekä lopuksi cp libjpeg.* /usr/lib (kopioi tiedostot /usr/lib – hakemistoon).

Ladataan GD-kirjasto (.tar.gz) osoitteesta http://www.boutell.com/gd/ /tmp-hakemistoon. Puretaan paketti gunzip gd-2.0.33.tar.gz-komennolla sekä luodaan hakemisto asennusta varten tar xf gd-2.0.33.tar–komennolla. Siirrytään komentotulkissa gd-2.0.33–hakemistoon komennolla cd /tmp/gd-2.0.33/ . Ajetaan konfigurointiskripti ./configure argumentilla –with-jpg=/usr/lib. Konfigurointi-skripti antaa yhteenvedon, jossa kerrotaan GD2.0.33:sen tukevan png- ja jpeg-kirjastoja. Kirjoitetaan seuraavaksi komento make install. Kirjasto asentuu hakemistoon /usr/local/lib.

Varmistetaan, että /etc/ld.so.conf–tiedostossa on merkintä gd-, jpeg-, ja png-kirjastojen hakemistoista. Ajetaan komento ldconfig, jolla päivitetään ajon aikaisten linkkien asetukset. Ajetaan komento make clean Nagioksen distribuutiossa, jotta nykyisen asennuksen väärät viittaukset korjaantuvat.

Siirrytään hakemistoon /tmp/nagios, minne on ladattu Nagios ohjelman asennuksen alkuvaiheessa. Ajetaan Nagioksen konfigurointiskripti uudelleen lisäparametreillä. Skripti on muotoa ./configure –prefix=/usr/local/nagios
--with-cgiurl=/var/www/html/nagios/cgi-bin
--with-htmurl=/var/www/html/nagios/
--with-nagios-user=nagios
--with-nagios-grp=nagios
--with-gd-lib=/usr/local/lib
--with-gd-inc=/usr/local/include.

Skripti ilmoittaa, että GD-kirjastot ovat löytyneet. Kirjoitetaan komento make all jotta ydinohjelma ja CGI:t kääntyvät uudelleen. Kirjoitetaan komento make install jotta ydinohjelma, cgi:t sekä html-tiedostot asentuvat. Nyt ennen kirjastojen asentamista puuttuneiden statusmap.cgi:n, trends.cgi:n sekä histogram.cgi:n pitäisi myös asentua. Kirjoitetaan komento make install-init jotta init-skripti asentuu /etc/rc.d/init.d–hakemistoon.

Käynnistetään lopuksi Nagios ja www-palvelin uudelleen komennoilla /etc/rc.d/init.d/nagios restart sekä /etc/rc.d/init.d httpd restart. Kun nyt avataan selain ja siirrytään osoitteeseen http://localhost/nagios, pitäisi linkkien statusmap, trends ja histogram toimia oikein.

4.9 Statusmap.cgi:n konfigurointi

Statusmap.cgi esittää Nagiokseen konfiguroidun verkon rakenteen graafisena esityksenä png-muodossa. Cgi kerää kuvaa varten tarvittavat tiedot Nagioksen peruskonfigurointitiedostoista. Mikäli halutaan luoda monipuolisempi kuva, statusmap.cgi:ta varten voidaan määritellä myös extendend host information-asetukset. Alkuun pääsee hyvin pelkillä perusasetuksilla, jotka määritellään cgi.cfg, hosts.cfg ja hostextinfo.cfg –tiedostoihin.

Nagioksessa hallinnoitaville koneille voidaan koneen nimen yhteyteen lisätä pieni kuva, mikä kertoo onko kyseessä esimerkiksi reititin, kytkin, Windows-kone, Linux-kone vai tulostin. Ladataan ja puretaan kuvapaketti imagepak-base.tar.gz koneelle. Kopioidaan base-hakemiston sisältö hakemistoon /usr/local/nagios/share/images/logos.

Nagiokselle täytyy seuraavaksi kertoa mitä kuvaa käytetään minkäkin hostin kanssa. Avataan tiedosto cgi.cfg, ja lisätään sinne rivi xedtemplate_config_file=/usr/local/nagios/etc/hostextinfo.cfg. Lisätään hakemistoon /usr/local/na-gios/etc tiedosto hostextinfo.cfg, mihin määritellään mitä kuvaa käytetään milläkin hostilla. Esimerkki:

```
define hostextinfo{
    host_name           XP-Kone
    icon_image          win40.png
    icon_image_alt      XP-KONE
    statusmap_image     win40.gd2
    }
```

Host_name on laitteen nimi, sama mikä sille on määritelty hosts.cfg-tiedostossa. Icon_image kertoo mikä kuva /usr/local/nagios/share/images/logos/-hakemistosta halutaan lisätä laitteelle. Icon_image_alt määrittelee laitteen kuvauksen, mikä näkyy kuvan alla. Statusmap_image on kuva, mikä liitetään statusmap.cgi:ssa koneeseen. Tämän kuvan pitää olla *.gd2-muotoa.

Cgi.cfg-tiedostossa löytyy kohta, missä voidaan määritellä statusmap:n oletuslayout (default_statusmap_layout=1-5). Alussa pelkät perusmäärittelyt riittävät hyvin.

Jos halutaan esittää verkon kuva sellaisessa muodossa kuin verkko on oikeasti rakennettu, tarvitaan parents-määrittelyjä. Parents:t ovat koneita, joiden alapuolella muut koneet ovat, esimerkiksi reittittimiä ja kytkimiä. Parents-määrittely lisätään hosts.cfg–tiedostoon sen laitteen tietoihin mikä halutaan esittää olevan jonkin toisen laitteen alaisuudessa. Esimerkkinä XP-kone, mikä on kiinni switch1–kytkimessä:

```
#xpkone host definition
define host{
    use                 generic-host
    host_name           XP-Kone
    alias               XP-Kone
    address             10.10.10.2
    parents             switch1
    check_command       check_ping
    .
    .
}
```

5 Yhteenveto ja kehitysideoita

Työn tärkeimpänä tavoitteena oli tutkia mitä kaikkea Nagioksella voidaan valvoa Windows-koneista, sekä selvittää kuinka Nagios-verkonhallintajärjestelmä laitetaan toimintaan. Nagios on erittäin monipuolinen ja laaja verkonhallinnan työkalu, jota voidaan muokata loputtomiin käyttäjän tarpeita vastaaviksi. Suurimpana vahvuutena Nagioksella onkin sen Open Source-pohjainen ympäristö, mikä sallii ohjelman muuntelun hyvinkin erilaisista laitteista ja alustoista koostuvien verkkojen hallintaan. Erilaisia Plug-in:ä on valmiiksi olemassa satoja, ja jokainen käyttäjä voi vapaasti kirjoittaa uusia plug-in:ja tai muokata jo olemassa olevia, mikäli valmiista vaihtoehdoista ei löydy käyttötarkoitukseen sopivia.

Nagioksen heikkoutena saattaa olla sen ilmaisuus sekä vain käyttäjien varassa tapahtuva ohjelman kehittyminen. Kaupallisilla versioilla saadaan maksua vastaan uusi parannettu versio ohjelmasta, avoimen lähdekoodiin perustuvien ohjelmien kanssa toimiessa joutuu odottamaan jonkun toisen käyttäjän tekemiä päivityksiä tai ne on tehtävä itse. Lisäksi käyttäjätukea ei löydy kuin Internetin postituslistoilta, keskustelupalstoilta tai dokumentaatiota selaamalla. Kaupalliseen ohjelmaan päätymällä käyttäjä voi odottaa saavansa ongelmatilanteissa apua ja neuvoa puhelimitse tai paikanpäälle saapuvan asiantuntijan kautta. Toisaalta Linux-maailman vahvuutena ovat erinomaiset dokumentaatiot, laajat postituslistat sekä innokkaasti avustavat asioihin enemmän perehtyneet käyttäjät.

Nagioksen laajuudesta ja loputtomista erilaisista muokkausmenetelmistä johtuen tässä työssä ei ole päästy kovinkaan syvälle Nagioksen maailmaan. Mukaan on otettu ne toimenpiteet, millä järjestelmä saadaan nostettua toimintakuntoon, sekä rakennettua pieni testiympäristö erilaisten toimintojen kokeilua varten. Monia kiinnostavia ja kokeilun arvoisia asioita jäi tässä työssä tekemättä, esimerkkinä hälytysten lähettäminen verkonvalvojan matkapuhelimeen.

Kokonaisuudessaan Nagios on verkonhallintajärjestelmänä suositeltava vaihtoehto. Sen konfiguroiminen toimintavalmiuteen vaatii käyttäjältä paljon aikaa ja asiaan perehtymistä, mutta ohjelman monipuolisuus sekä skaalautuvuus hyvinkin erilaisiin verkkoihin ja laitealustoihin muodostaa siitä vahvan vaihtoehdon kaupallisille ohjelmille. Lisäksi se ei vaadi kovinkaan erikoista laitteistoa toimiakseen verkonhallinnan komponenttina, tavallinen verkkoon liitetty työasema Linux-käyttöjärjestelmällä riittää hyvinkin pitkälle. Nagioksen käyttöönottaminen antaa mahdollisuudet hallita verkkoa tehokkaasti ja taloudellisesti.

Toimeksiannossa pyydettyä selvitystä Nagioksen tarjoamista vaihtoehdoista en pystynyt täyttämään niin hyvin kuin alun perin kuvittelin. Suurimpana tekijänä olivat ohjelman suunnattomat mahdollisuudet rakentaa toimiva ja käytännöllinen verkonhallintajärjestelmä. Mikäli

olisin ottanut kaikki Nagioksen tarjoamat mahdollisuudet mukaan työhöni, olisi siitä tullut liian laaja opinnäytetyöksi. Testiympäristön rakentaminen sujui alkuhankaluuksista huolimatta hyvin, ja sen päälle on toimeksiantajan helppoa jatkaa Nagioksen muokkausta ja konfigurointia verkonhallinnan tarpeiden mukaan.

Lähteet

112 – Hätäkeskuslaitos – Hätäkeskusuudistus. [online] [viitattu 24.2.2005].
www.112.fi/index.php?pageName=hatakeskusuudistus

112 – Hätäkeskuslaitos – Perustietoa. [online] [viitattu 24.2.2005].
www.112.fi/index.php?pageName=pirkanmaa1

Common Management Information Protocol. [online] [viitattu 3.3.2005]
http://www.sei.cmu.edu/str/descriptions/cmip_body.html

Duck, Michael & Read, Richard 2003. Data Communications and Computer Networks for Computer Scientists and Engineers.
Essex, UK: Pearson Education Limited

Feldman, Jonathan 1999. Verkonhallinta. Suom. Tapani Lahtinen.
Jyväskylä: Gummerrus Kirjapaino Oy

Hautaniemi, Mika 1994. Diplomityö: TKK/Atk-keskuksen TCP/IP-verkon valvonta ja hallinta. [online] [viitattu 23.2.2005].
www.hut.fi/~hau/thesis/diplomityo.book.html

Hunt, Craig 1998. TCP/IP-verkonhallinta – Tehokäyttäjän opas. Suom. Erkki Suominen.
Jyväskylä: Gummerrus Kirjapaino Oy

Jaakohuhta, Hannu & Lahtinen, Tapani 1997. Tietoliikenneverkot – Tehokäyttäjän opas.
Jyväskylä: Gummerrus Kirjapaino Oy.

Leinwald, Allan & Fang Conroy Karen 1996. Network Management A Practical Perspective.
Addison Wesley Longman Inc.

Nagios : About Nagios. [online] [viitattu 23.2.2005].
www.nagios.org/about.php

Tietoverkkolaboratorio – TKK Verkonhallintakurssin luentomateriaali [online] [viitattu 23.2.2005].
www.netlab.hut.fi/opetus/s38188/1997/luento12/verkonhallinta-x6.pdf