Opinnäytetyö (AMK)

Tietotekniikan koulutusohjelma

Sulautetut järjestelmät

2013

Juuso Rintala

VERKKOTESTAUS AUTOMAATIOLAITTEIDEN SIJOITUSPAIKOISSA

Verkontestausohjelman asennus ja ohjelmointi

TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietotekniikan koulutusohjelma | Sulautetut järjestelmät

2013 | 40

TkL Jari-Pekka Paalassalo

Juuso Rintala

VERKKOTESTAUS AUTOMAATIOLAITTEDEN SIJOITUSPAIKOISSA

Tämän opinnäytetyön aiheena oli tutkia erään yrityksen automaatiolaitteiden ja palvelimien välisen verkon toimivuutta laitteiden sijoituspaikan näkökannalta. Tavoitteena oli edesauttaa uuden tietoliikenneprotokollan käyttöönottoa yrityksen automaatiolaitteiden ja palvelimien välillä. Nykyisen virtuaalisen yksityisverkon (VPN) tiedonvälityksen toteuttavan UDP-protokollan tilalle oli tulossa TCP/IP-protokolla, jonka toimintaperiaate ja -varmuus ovat erilaisia kuin nykyisen protokollan.

Työtä varten hankittiin viisi Linux-pohjaista testauslaitetta, joihin ohjelmoitiin verkontestausohjelma Nagios Core, joka testasi vuorokauden ajan toimintopaikan verkkoa yleisellä tasolla tunkeutumatta yritysasiakkaan sisäiseen verkkoon. Laite vietiin sattumanvaraisesti valittuun tai verkko-ongelmaiseen sijoituspaikkaan ja asennettiin automaatiolaitteiden kanssa samaan verkkopäätteeseen, joka oli yhteydessä yrityksen palvelimille. Testausohjelma automatisoitiin suorittamaan yrityksen palvelimille vuorokauden ajaksi erilaisia testejä, kuten PING sekä UDP- ja TCP-porttiskannaus. Samalla ohjelma keräsi saadun testausdatan erilliseen tiedostoon.

Tutkimuksen avulla saadaan tulevaisuudessa vähennettyä automaattilaiteverkon ongelmista aiheutuvia työmääräyksiä automaattilaitteiden kunnossapidolle ja koottua eri toimintopaikkaket-jujen verkko-ominaisuuksien yhtäläisyyksiä tulevaa protokollan vaihdosta varten.

ASIASANAT:

Linux, Ubuntu, Nagios, verkontestaus, TCP/IP, UDP

BACHELOR´S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Degree Programme in Information Technology | Embedded Software

2013 | 40

Lic.Tech Jari-Pekka Paalassalo

Juuso Rintala

NETWORK TESTING FOR M2M INSTALLATION

The subject of the thesis was to research network functionality between automatic devices and servers for a company. The research was implemented in placement locations of automatic devices. The aim of the project was to further the deployment of new telecommunication protocol between the devices and servers. The current virtual private network (VPN) transport layer protocol (UDP) was replaced by another protocol (TCP) which functions in a different way and is more secure than the current.

For this piece of research, five new Linux based testing devices were purchased. Nagios Core, a program for network monitoring, was installed to test the network. The devices were placed in the same network of automatic devices for twenty four hours. The placement locations were randomly selected or they had had problems in their network. The Nagios Core was programmed to execute different kind of network tests, such as PING and UDP/TCP port scanning. The test results were stored in a specific file for later use.

With the help of this piece of research, the assignments for maintenance caused by network problems will be reduced in the future. In addition, the similarities of network characteristics of placements are collected to be used for the change of the protocol.

KEYWORDS:

network, UDP, TCP, Linux, Nagios

SISÄLTÖ

KUVAT

TAULUKOT

KÄYTETYT LYHENTEET

ISAKMP	(Internet Security Association and Key Management Protocol) on tietoturva-avainpalveluprotokolla
ICMP	(Internet Control Message Protocol) on protokolla, jolla viestejä lähetetään tietokoneesta toiseen
IPsec	(Internet Protocol Security Architecture) on tiedonsalausprotokolla yhteyksien suojaamiseen
NAT	(Network Address Translation) eli osoitteenmuunnos piilottaa tai säästää julkisen liikenteen IP-osoitteita
NTP	(Network Time Protocol) välittää täsmällisen ajan laitteiden välillä
OSI	(Open System Interconnection Model) näyttää verkon rakenteen jakaen sen seitsemään kerrokseen
PING	Testi, jolla voidaan määrittää laitteen vastaanottavuus
TCP	(Transmission Control Protocol) yhteydellinen tiedostonsiirtoporokolla
UDP	(User Datagram Protocol) yhteydetön tiedostonsiirtoprotokolla
VPN	(Virtual Private Networking) on suojattu yhteys julkisen verkon yli

1 JOHDANTO

Erään kansallisen yrityksen automaatiolaitekanta on sijoitettu erilaisiin toimintopaikkoihin ympäri maan. Automaatiolaitteet ovat verkkoyhteydessä yrityksen eri palvelimiin suojatun yhteyden, virtuaalisen yksityisverkon kautta (VPN). Suojattu yhteys luo turvallisen yksityisen verkon julkisen verkon (internet) ylitse, jolloin viesti tunneloidaan erilaisten salausprotokollien avulla. Nykyinen automaatiolaitekanta käyttää IPsec-salausprotokollaa, joka käytännössä muuntaa viestit UDP-paketeiksi, jotta ne läpäisevät palomuurien osoitteenmuunnokset.

Uudistuvan laitekannan myötä on mietitty myös vaihtaa tunnelointijärjestelmän, joka tullaan ottamaan yrityksessä käyttöön tämän tutkimuksen valmistuttua. Sijoituspaikkoja on pyydetty aikaisemmin avaamaan palomuuriensa tietyt UDP- ja TCP-portit, mutta todellisuudessa näin ei ole tehty.

Tämä opinnäytetyö on luokiteltu pääosin salaiseksi. Julkiseen osaan sisältyy teoriaosuus ja toteutusosan ohjelmointivaihe pääosin. Luottamukselliseen osaan kuuluu toteutusosan yritystä koskeva ohjelmointi ja tulosten tarkastelu.

Työssä tutkitaan yrityksen automaatiolaitteiden ja palvelimien välistä verkkoyhteyttä ja verkon palveluja laitteiden sijoituspaikkojen osalta. Työn avulla saadaan mahdollisesti selville ne sijoituspaikkaketjut, joiden palomuureihin on tehtävä lisäavauksia, etenkin TCP-portteihin. Sijoituspaikkoja voidaan näin opastaa tekemään tarvittavat muutokset, jotta uusi laitekanta saa turvallisen yhteyden yrityksen palvelimiin.

Ensimmäiseksi työtä varten tullaan valitsemaan tarvittavat laitteet, joihin asennetaan Linux Ubuntu – käyttöjärjestelmä. Laitteiden tuli olla mahdollisimman pienikokoisia, jotta ne mahtuivat automaatiolaitteen sisään, jotta ne eivät joutuisi alttiiksi väärinkäytöille ja ilkivallalle.

Toiseksi laitteisiin on tarkoitus ohjelmoida verkontestaus- ja analysointiohjelma Nagios Core. Ohjelma on ilmainen ja sen käyttöliittymä on selainpohjainen sekä helppo käyttää. Nagios Core -ohjelmaan asennetaan tarvittavat testit, kuten

UDP- ja TCP-porttiskannaus, jotka ohjataan eri palvelimille, käytännössä IP-osoitteisiin.

Lopuksi saaduista testituloksista luodaan vertailutaulukko, joiden avulla tuloksia on helppo verrata keskenään. Loppuun on koottu myös palvelinkohtainen eritte-ly toimineista ja toimimattomista testeistä ja niiden vastausajoista.

2 VERKKOLIIKENTEEN TEORIA

2.1 Verkon rakenne

2.1.1 OSI-malli

Kaikille verkoille on yhteistä se, että ne voidaan pilkkoa eri kerroksiin OSI-mallin avulla (kuva 1). OSI-malli helpottaa verkon suunnittelua ja sitä on helppo ymmärtää, kun verkon jokainen kerros on oma kokonaisuutensa. Verkko voidaan jakaa seitsemään eri kerrokseen [1]:

1. Fyysinen kerros (kerros 1) määrittää eri verkkotekniikoille ominaisia bittien kuljetukseen liittyviä ominaisuuksia. Kerros on ainoa, joka sisältää nähtävissä olevia asioita, kuten lähetettäviä signaaleja, koska muiden kerrosten liikenteensiirrosta vastaa ohjelmisto.
2. Siirtoyhteyskerros (kerros 2) muodostaa kehyksen, jonka sisältö koostuu verkkokerrokselta saaduista tiedoista, ja huolehtii datan luotettavasta siirrosta fyysisiä siirtoteitä pitkin.
3. Verkkokerros (kerros 3) pakkaa kuljetuskerrokselta saadun datan paketteihin ja reitittää ne vastaanottajalle verkkokerroksen osoitetietojen mukaisesti eri aliverkkojen läpi.
4. Kuljetuskerros (kerros 4) pilkkoo alemmilta kerroksilta saadun tiedon sopivan kokoisiin segmentteihin ja luo yhteyden järjestelmien välille.
5. Yhteysjaksokerros (kerros 5) huolehtii sovellusten toimintojen koordinoimisesta eri laitteiden välillä esimerkiksi käynnistämällä ja pysäyttämällä lähetyksiä. Kerros huolehtii myös pakettien saapumisesta oikeassa järjestyksessä.
6. Esitystapakerros (kerros 6) määrittää sen, missä muodossa tietoliikennepaketti esitetään.
7. Sovelluskerros (kerros 7) tarjoaa sovellukset, joita varten koko yhteysketju luodaan, kuten tiedostojen siirto ja sähköposti.

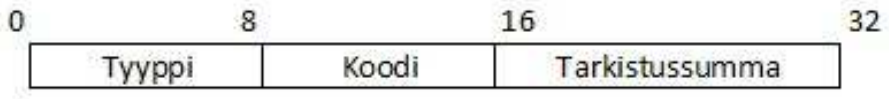
Kuva 1. OSI-malli yrityksen näkökannasta.

2.1.2 IP

Internet Protocol (IP) toimii selkärankana nykypäivän Internetille. Sen toiminta-periaate on reitittää aiempien OSI-mallin kerrosten luomia tietokehyksiä keskenään yhdistettyjen tietoverkkojen välillä eri laitteista toiseen. [2]

2.1.3 ICMP

Internet Control Message Protocol (ICMP) toimii tiiviisti yhdessä IP-protokollan kanssa. Se on luotu välittämään informaatiota erilaisista verkon häiriötilanteista ja sen otsikkokenttä on hyvin yksinkertainen (kuva 2). Se voi viestittää lähettäjää esimerkiksi saavuttamattomasta kohteesta, pakettien uudelleenohjauksesta tai lähetysnopeuden vähentämisestä. ICMP:n avulla pystytään tutkimaan, onko vastaanottava laite valmis tiedonsiirtoon PING-komennon avulla ja mikä on siirtoreitti vastaanottavaan laitteeseen traceroute-komennolla. Traceroute kertoo toisin sanoen sen, kuinka monta reititintä on kahden laitteen välillä. [3]



Kuva 2. ICMP-protokollan otsikkokenttä.

2.1.4 UDP

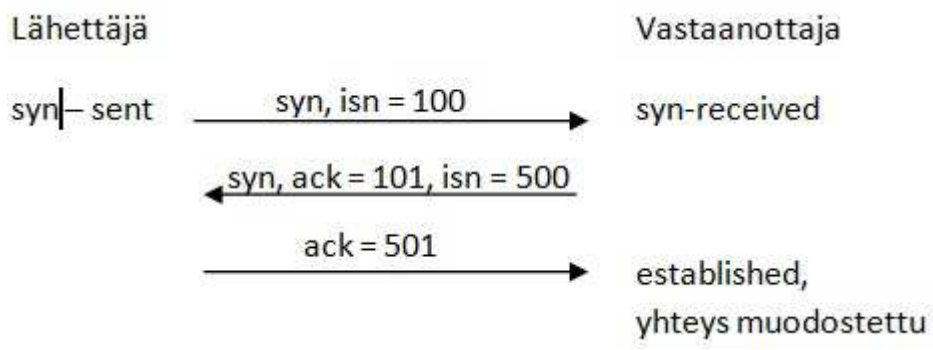
User Datagram Protocol (UDP) on yksinkertainen ja kevyt kuljetusprotokolla, joka tarjoaa yhteydettömän kuljetuspalvelun sovelluksille. UDP lisää verkkokerroksen (IP) päälle periaatteessa vain lähtö- ja vastaanottoportit, joiden avulla se osaa lähettää paketin oikeaan osoitteeseen ja vastaanottaja tietää, mistä paketti on tullut. Protokolla on suunniteltu toimimaan yhteydettömänä eli mahdollisimman vähäisillä mekanismien käytöllä. Yhteydetön toiminta tarkoittaa käytännössä, että kommunikoivien laitteiden välille ei luoda yhteyttä ennen pakettien välitystä vaan data lähetetään verkkoon ja toivotaan, että se menee perille. [1]

UDP käyttää datan kuljetukseen porttinumeroita. Kommunikointi laitteiden välillä tapahtuu porttinumeroiden ja IP-osoitteiden perusteella. Lähdeportti kertoo mistä porttinumerosta, eli miltä sovellukselta datapaketti lähtee ja vastaavasti vastaanottoportti kertoo mihin paketti osoitetaan. [1]

2.1.5 TCP

Transmission Control Protocol (TCP) on kuljetusprotokollista huomattavasti monipuolisempi, jonka vuoksi sitä käytetään enemmän vaativammissa kuljetuksissa. TCP on yhteydellinen protokolla, johon liittyy eri mekanismeja, joiden avulla se on luotettavampi kuin UDP. TCP takaa, että lähetettävät paketit pysyvät oikeassa järjestyksessä sekä sen, että paketit pääsevät vastaanottajalle. [1]

TCP-protokollan toiminta perustuu lähettävän ja vastaanottavan laitteen keskusteluun lähetyksen aikana (kuva 3). Lähettäjä toimittaa verkkoon aluksi yhteyden avauspyynnön, jonka vastaanottaja kuittaa ja lähettää kuittauksen takaisin. Yhteyden avauksen jälkeen lähettäjä tietää yhteyden olevan luotu ja se voi lähettää useampia paketteja vastaanottajalle. Tämän jälkeen lähettäjä jää odottamaan kuittauspyyntöä vastaanottajalta, joka pakettien perille saapumisen jälkeen lähettää verkkoon kuittauksen mitä pakettia se odottaa saavansa seuraavaksi. Jos lähettäjä ei saa kuittauspyyntöä tietyn ajan kuluessa, se lähettää aiemmin lähettämänsä paketit uudelleen. [1]



Kuva 3. TCP-protokollan yhteydenmuodostus.

2.2 Verkon suojaus

2.2.1 Virtual Private Networking

Virtual Private Networking (VPN) on yksityinen virtuaaliverkko, jolla voidaan luoda suojattu yhteys julkisen verkon yli. Suojatun yhteyden muodostamiseen tarvitaan suojattua tilaa eli tunnelointia. Tunnelin avulla IP-paketti pystytään kapseloimaan suojatun tilan sisään ja kuljettamaan eri salausprotokollien avulla laitteesta toiseen. [4]

2.2.2 IPsec

Internet Protocol Security Architecture (IPsec) on eräs verkkokerroksen salausmekanismi, jolla voidaan VPN-yhteys suojata. IP-paketin kuljetuksen salauksessa IPsec käyttää kahta protokollaa, jolla yhteys saadaan suojatuksi, Authentication Header (AH) ja Encapsulating Security Payload (ESP). IPsec-paketit kapseloidaan usein UDP-paketeiksi, jotta paketit pääsevät läpi palomuurien NAT-muunnoksista, sillä NAT-reitittimet saattavat hylätä muut kuin TCP/UDP-paketit. [4]

3 VERKONTESTAUSOHJELMAN ASENNUS

3.1 Laitteiden ja ohjelman valinta

3.1.1 Testilaite

Testauslaitteeksi valittiin kannettava minitietokone Asus Eee Pc Sheashell Series. Laitteita hankittiin 5 ja niihin asennettiin käyttöjärjestelmäksi Linux Ubuntu 10.3., sillä Linux on huomattavasti helpommin käyttäjän tarpeisiin muokattava käyttöjärjestelmä kuin esimerkiksi Windows. Toinen vaikuttava tekijä Linuxin valinnassa oli se, että opinnäytetyön tilaavan yrityksen automaatiolaitteiden käyttöjärjestelmänä on myös Linux.

Testauslaite vietiin joko verkko-ongelmalliseen tai sattumanvaraiseen sijoituspaikkaan, jossa se asennettiin yhden automaatiolaitteen sisälle, joten sen tarvitsi olla mahdollisimman pienikokoinen. Testilaitteen virtajohdon ei myöskään sopinut olla liian suuri, jotta sen pystyi vetämään pistorasiasta automaatin sisään. Jos sijoituspaikassa oli useampi kuin yksi, mutta vähemmän kuin neljä automaatiolaitetta, voitiin testilaitteen laajakaistakaapeli kytkeä suoraan sijoituspaikassa jo olleeseen automaattien käyttämään verkkomodeemiin. Useamman kuin kolmen automaatin sijoituspaikassa täytyi testilaitteen lisäksi asentaa verkkokytkin, jotta yksikään automaattilaite ei jäänyt ilman verkkoyhteyttä. Tämä oli tärkeää varsinkin, jos automaattiin oli asennettu korttimaksupääte. Yhden automaatiolaitteen sijoituspaikoissa oli asennettava erillinen verkkomodeemi, johon sekä testauslaite että automaatiolaite kytkettiin. Verkkomodeemiin tuli ohjelmoida automaatille annettu kiinteä ip-osoite, jotta sekä automaatti että testilaite saivat verkkoyhteyden.

Laajakaistakaapelin kytkemisen jälkeen oli etsittävä testilaitteelle pistorasia. Kummankin johdon, sekä laajakaistakaapelin että virtajohdon, sai vietyä automaatin sisään kätevästi automaatin takana sijainneesta reiästä, josta automaa-

tin oma verkkokaapelikin on asennettu. Testilaite jätettiin automaatin sisään niin, ettei se sammunut eikä mennyt lepotilaan virtakäytössä. Tämän takia sijoituspaikkojen työntekijöitä ohjeistettiin jättämään automaatiolaitteisiin virrat päälle koko tutkimusajaksi, jottei testilaite joutuisi akkukäyttövirtaan, sillä laitteen akku ei kestänyt vuorokauden yhtämittaista käynnissäoloa. Testiä pyrittiin ajamaan noin 24 h, jonka jälkeen laite haettiin pois ja vietiin uuteen paikkaan.

3.1.2 Nagios Core – testiohjelma

Opinnäytetyön testilaitteisiin ohjelmoitiin Nagios Core 3.2.3., joka oli sen hetken uusin versio. Nagios Core on vapaaseen lähdekoodiin perustuva monipuolinen valvontaohjelma, jolla verkkoinfrastruktuurin ongelmia voidaan tunnistaa, ratkaista ja ennaltaehkäistä. Ohjelma on rakennettu toimimaan juuri Linux-käyttöjärjestelmässä. Ohjelmalla voidaan valvoa ja testata erilaisia verkkopalveluja, kuten SMTP, UDP, TCP, HTTP ja PING. Nagios Coren avulla voi myös tarkkailla esimerkiksi eri palvelimien levyn tilaa ja prosessoreiden levyn käyttöä, jotta huomataan esimerkiksi palvelinten ylikuormittuminen ennen verkon kaatumista.

Nagios toimii erillisten lisäosien avulla, jotka käyttäjä saa itse lisätä ohjelman ytimeen. Muokkausominaisuuden myötä testiohjelmasta saa valmistettua käyttäjän tarpeisiin sopivan. Suurin vaikuttava tekijä Nagioksen valintaan oli se, että ohjelma oli ilmainen. Nagios Core -testiohjelman ehdottomia positiivisuuksia oli myös ohjelman käyttöliittymän avautuminen www-selaimeen. Erillistä käyttöliittymää ei siten tarvittu, vaan käyttäjä sai kaiken ohjelmoidun näkyville suoraan selaimeen.

3.2 Asennusvaihe

Luvun 3 asennusvaiheet ovat lainattu samasta lähteestä. [5]

3.2.1 Apachen asennus

Nagios-verkonhallintaohjelma tarvitsee toimiakseen erillistä www-palvelinsovellusta, joksi valittiin Apache HTTP Server. Valinta oli helppo, sillä Apache on ilmainen ja Linux-pohjaisille järjestelmille suosituin vaihtoehto.

Ensin tarvitsi asentaa Linux-alustalle kääntäjä ja tarvittavat GD-kirjastot, jotka muodostavat Nagioksen selainpohjaiseen käyttöliittymään esikatseltavia pikkukuvia. Sen jälkeen asennettiin Apache:

root@nagios:~# apt-get install build-essential

root@nagios:~# apt-get install libgd2-xpm-dev

root@nagios:~# apt-get install apache2

Apachen lisäosaksi asennettiin PHP, jota tarvitaan dynaamisten www-sivujen luontiin:

root@nagios:~# apt-get install php5-common php5 libapache2-mod-php5

Apache loi asentamisen yhteydessä konfigurointikansion apache.conf, jonne lisättiin yksi rivi, jota ei asennus osannut muodostaa. Rivin lisäyksen jälkeen Apache käynnistettiin taustalle restart-komennolla, joka ajaa saman asian kuin pelkkä start-komento:

root@nagios:~# gedit /etc/apache2/apache2.conf → DirectoryIndex index.html index.php index.cgi

root@nagios:~# /etc/init.d/apache2 restart

3.2.2 Nagios Core -testausohjelman asennus

Ennen Nagios Core 3.2.3 – version lataamista ja asennusta tarvitsi luoda palvelun käyttäjäksi nagios, salasana ja hallintaryhmäksi nagcmd, johon liitettiin käyttäjät nagios ja apache:

root@nagios:~# useradd -m nagios

root@nagios:~# passwd nagios

root@nagios:~# groupadd nagcmd

root@nagios:~# usermod -a -G nagcmd nagios

root@nagios:~# usermod -a -G nagcmd www-data

Käyttöasetusten luonnin jälkeen ladattiin Nagioksen www-sivuilta ohjelman viimeisin versio. Pakattu kansio purettiin, jonka jälkeen purettuun kansioon luotiin konfigurointikomennolla aiemmin tehty hallintaryhmä nagcmd. Puretun kansion alla ajettiin asennuskomennot, jotka luovat init script –komennot, asetustiedostot ja asettavat kansiolle oikeudet saada ulkoisia komentoja. Niiden lisäksi luotiin asetustiedostot Apachelle:

root@nagios:~# tar -zxvf nagios-3.2.3.tar.gz

cd usr/src/nagios-3.2.3

root@nagios: usr/src/nagios-3.2.3#./configure --with-command-group=nagcmd

root@nagios: usr/src/nagios-3.2.3# make all

root@nagios: usr/src/nagios-3.2.3# make install

root@nagios: usr/src/nagios-3.2.3# make install-init

root@nagios: usr/src/nagios-3.2.3# make install-config

root@nagios: usr/src/nagios-3.2.3# make install-commandmode

root@nagios: usr/src/nagios-3.2.3# make install-webconf

Seuraavaksi vaihdettiin kansiota ja luotiin Nagioksen verkkopalvelulle käyttäjä nagiosadmin ja sille salasana. Näitä tarvittiin, kun kirjauduttiin ohjelman selainpohjaiseen käyttöliittymään:

root@nagios: /usr/local/nagios/etc# htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin

New password:

Re-type new password:

Käyttäjän luonnin ja salasanan syöttämisen jälkeen lisättiin Nagios-käyttäjätiedot ja -hakemistot Apachen asetustiedostoon:

/etc/apache2/apache.conf

ScriptAlias /nagios/cgi-bin /usr/local/nagios/sbin

Options ExecCGI

AllowOverride None

Order allow,deny

Allow from all

AuthType Basic

AuthUserFile /usr/local/nagios/etc/htpasswd.users

Require valid-user

Alias /nagios /usr/local/nagios/share

Options None

AllowOverride None

Order allow,deny

Allow from all

AuthType Basic

AuthUserFile /usr/local/nagios/etc/htpasswd.users

Require valid-user

Tallentamisen onnistuttua voitiin Apache uudelleen käynnistää:

root@nagios:~# /etc/init.d/apache2 restart

3.2.3 Nagios Core Pluginien asennus

Nagios Core – ohjelman ytimen asennuksen jälkeen tarvitsi ladata erilliset lisäosat, joiden avulla ohjelmasta saa muokattua eri käyttötarpeisiin sopivan. Plugini-nit ladattiin myös Nagioksen www-sivuilta, minkä jälkeen ne purettiin omaan kansioon:

root@nagios:~# tar -zxvf nagios-plugins-1.4.14.tar.gz

cd usr/src/nagios-plugins-1.4.14

Pluginien omassa kansiossa annettiin konfigurointikomento, jolla plugineille annettiin samat käyttäjä- ja ryhmäinformaatiot kuin aiemmin Nagioksen ytimelle. Kun käyttäjätiedot oli syötetty, voitiin suorittaa asennuskomennot:

root@nagios: usr/src/nagios-plugins-1.4.14# ./configure --with-nagios-user=nagios --with-nagios-group=nagios

root@nagios: usr/src/nagios-plugins-1.4.14# make

root@nagios: usr/src/nagios-plugins-1.4.14# make install

3.3 Tarkistusvaihe

3.3.1 Asennusten varmistaminen

Ennen varsinaisten ohjelmointitiedostojen muokkausta oli hyvä suorittaa ohjelmalle tarkistus, jossa varmennettiin, että aikaisemmin asennetut ohjelmat ja lisäosat oli asennettu oikein. Sen lisäksi oli myös hyvä tehdä oletusasetustiedostoista varmuuskopiot eri kansioon:

root@nagios:~# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg

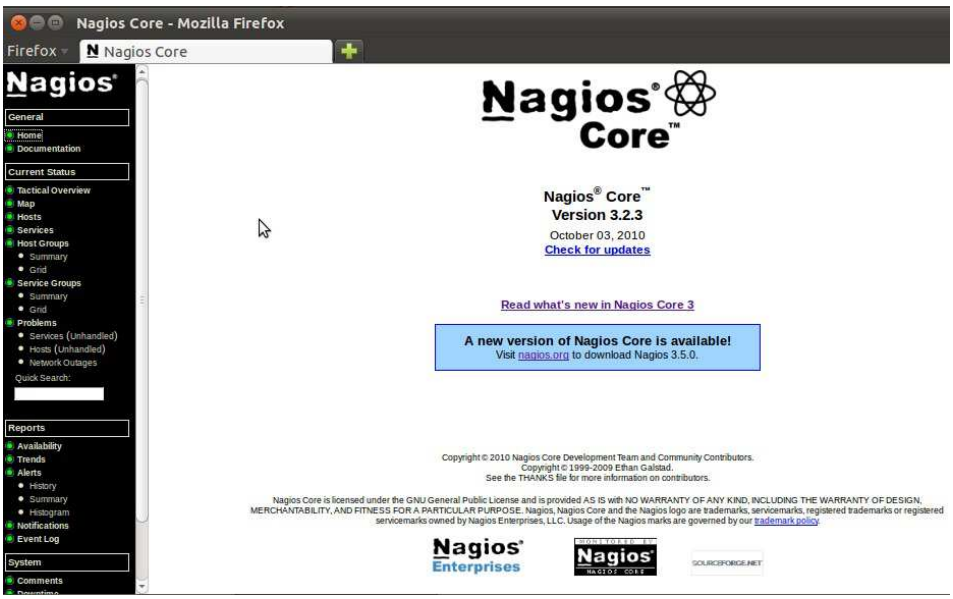
Varmistuskomennon jälkeen Linux lähti kääntämään ohjelmia ja tarkistamaan asennustiedostoja. Varmistuskomento piti aina suorittaa, kun asetustiedostoja muuteltiin. Kun varmentaminen oli ohi ja ohjelmasta ei löytynyt virheitä, voitiin Nagios käynnistää ja avata ohjelman käyttöliittymä (kuva 4):

Total Warnings: 0

Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check

root@nagios:~# /etc/init.d/nagios start

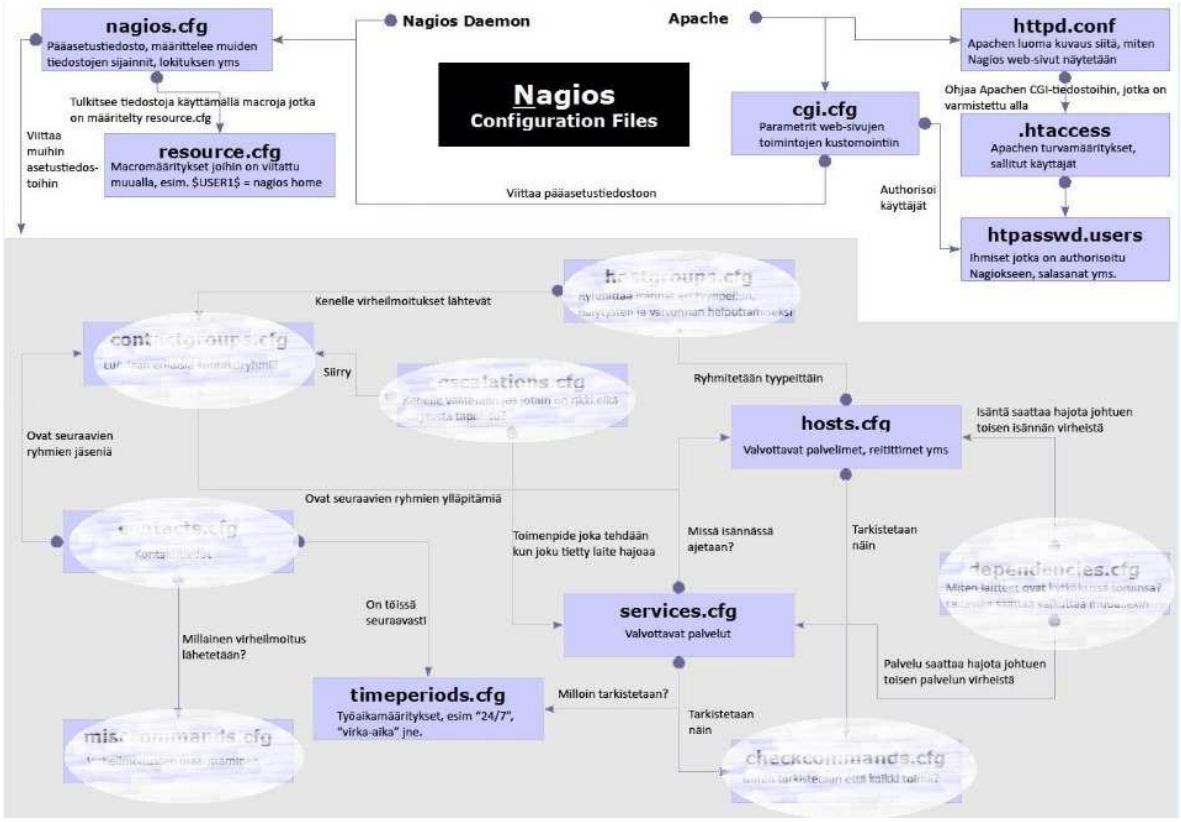


Kuva 4. Nagios Core –testausohjelman selaimen etusivu.

3.3.2 Ohjelmointivaihe

Nagioksen Core -testiohjelman räätälöiminen suoritettavaa tutkimusta varten tapahtui muokkaamalla ohjelman asetustiedostoja (kuva 5). Tiedostot oli asennettu aikaisemmin testiohjelmaan *make install-config* – komennolla ja niiden hakemistokansio Linuxissa oli */usr/local/nagios/etc/objects/.* Asetustiedostoja ohjelma loi kahdeksan kappaletta, joista tässä testaustyössä tarvittiin vain kolmea:

services.cfg	***hosts.cfg***	***timeperiods.cfg***
contacts.cfg	*printer.cfg*	*switch.cfg*
templates.cfg	*windows.cfg*	



Kuva 5. Nagioksen ja Apachen pääasetustiedostot sekä työssä käytetyt asetustiedostot.[6]

Asennusosio jatkuu tarkemmin luottamuksellisessa osiossa.

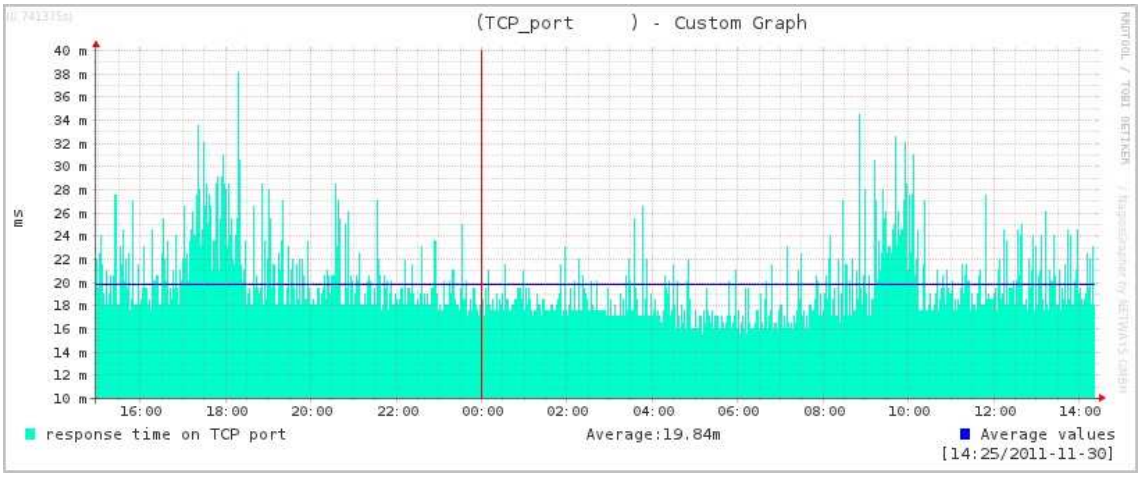
4 TULOKSET

Suoritetuista testeistä syntyi kuvaajia (kuva), joiden avulla testien vastausajat näkyivät vuorokauden ajalta. Kuvaajista analysoitiin tulosten keskiarvot ja huiput, jotka tallennettiin erilliseen taulukkoon (taulukko)

4.1 Testaukset

4.1.1 TCP-testi

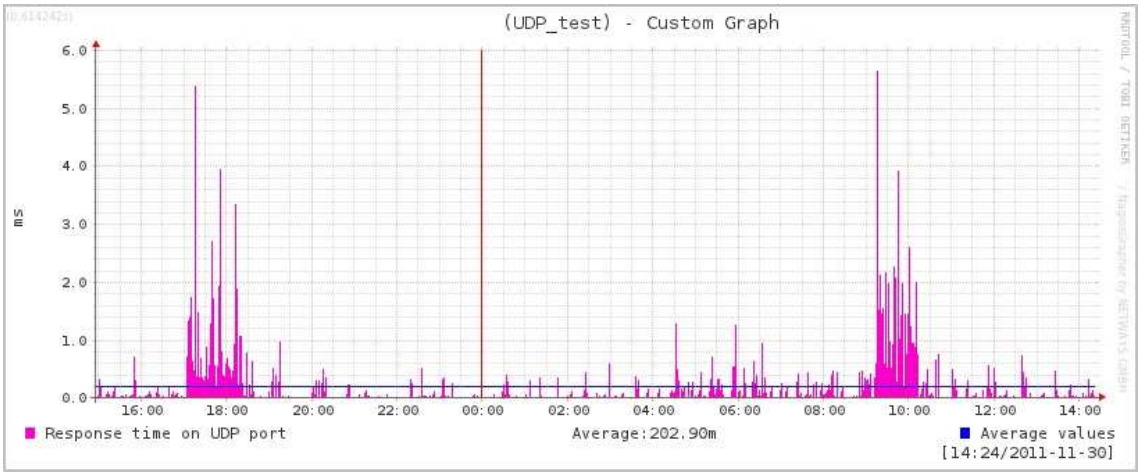
Sijoituspaikkoja on ennestään pyydetty avaamaan palomuurista automaatiolaitteiden verkkoliikennettä varten tiettyjä TCP-protokollan portteja. Nykyisen yhteysmuodon vuoksi portit ei välttämättä ole tarvinneet olla auki. Kuitenkin uuden virtuaalisen yksityisverkon tiedonsiirtoprotokollan vaihtumisen myötä sijoituspaikassa tarvitsee tulevaisuudessa olla joku tarvituista porteista auki. TCP-testi kohdistui portteihin, jotta nähtiin missä ketjupaikoissa tarvitsi asiakasta pyytää tekemään lisäavauksia palomuuriin.
Testi lähetti 1 min välein testipaketin haluttuun porttiin ja tulosti saadun vastausajan kuvaajaan (kuva 6). Kuvaaja laski noin 24 h:n ajalta vastausajan keskiarvon, joka otettiin ylös taulukkoon.



Kuva 6. TCP-testistä syntynyt kuvaaja palvelimelle.

4.1.2 UDP-testi

UDP-testi kohdistettiin yhteen tiettyyn porttiin. Sen avulla nykyinen tiedonvälitysprotokolla luo turvallisen ja salatun yhteyden (IPsec) yrityksen palvelimille. Sijoituspaikoissa edellä mainitun portin tarvitsi olla auki vähintään yhteen yrityksen palvelimista, jotta automaatiolaite onnistui luomaan yhteyden verkkoon. UDP-palvelun vastausajat (kuva 7) olivat pääosin moninkertaisia verrattuna TCP-palvelun tuloksiin.

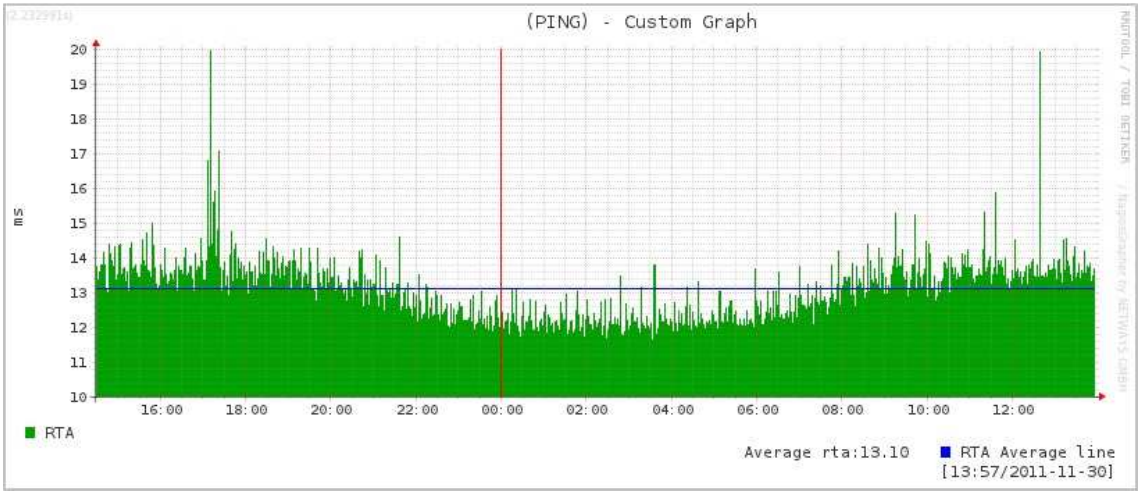


Kuva 7. UDP-testistä syntynyt kuvaaja palvelimelle

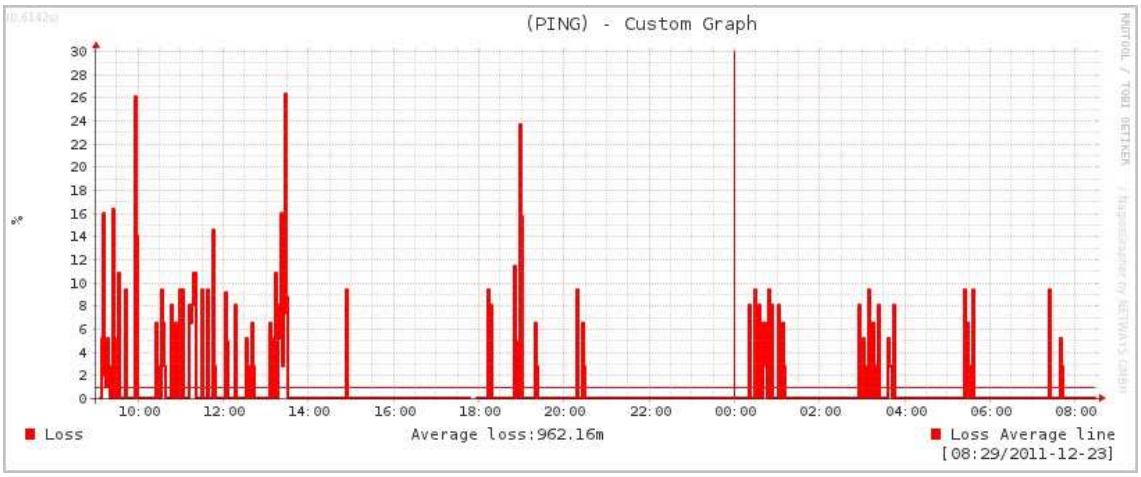
4.1.3 PING-testi

Ping-testillä haluttiin testata palvelimien vastaanottokykyä, verkon kuormituskykyä, mahdollisia kaapelihäiriöitä ja yleistä verkon toimivuutta. Testi lähetti 1 min välein testilaitteelta yrityksen palvelimelle 5 paketin pakettijoukon, jonka yhteiskoko oli 160b. Näin ollen 5 min aikajaksoon mahtui 5x160b, eli 800b ja esimerkiksi 4 %:n häviö vastasi yhden paketin (32b) epäonnistunutta lähetystä/vastaanottamista kuten kuvassa 12 on esitetty. 24 h:ssa lähetettyjen pakettien kappalemäärä nousi 7200:ksi ja yhteispakettikoko 230 Mb:ksi. Testi muodos-

ti vastaanottoajoista kuvaajan (kuva 8), johon se piirsi testitulokset minuutin välein, ja toisen kuvaajan (kuva 9), johon tallentui viestien häviöt prosentteina.



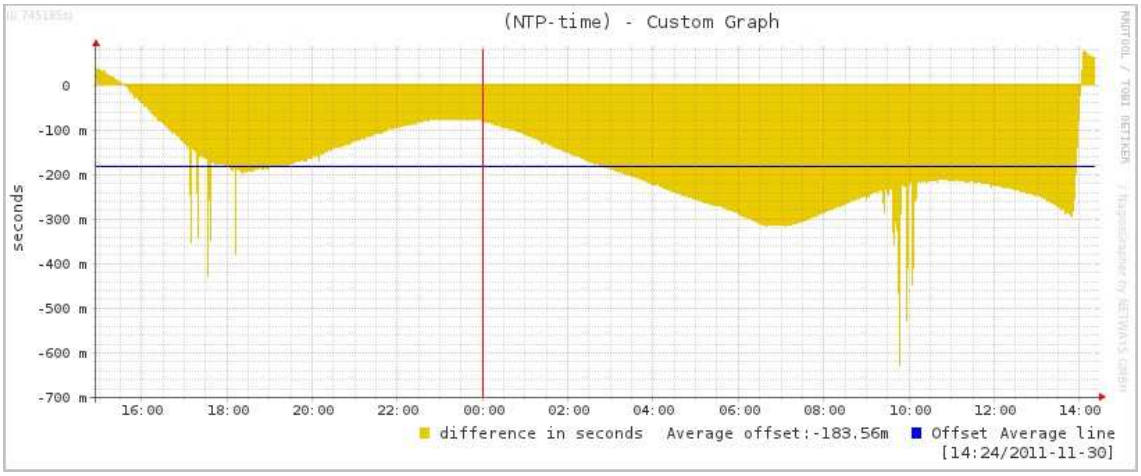
Kuva 8. PING-testistä syntynyt kuvaaja palvelimelle.



Kuva 9. PING-testin häviöprosenteista syntynyt kuvaaja palvelimelle.

4.1.4 NTP-testi

NTP-testin (kuva) avulla saatiin tietoa testattavan palvelimen ja sijoituspaikassa olevien automaatiolaitteiden välisestä aikaerosta. NTP-palvelu pyrkii tahdistamaan kahden laitteen välisen kellonajan samanaikaiseksi, jotta liikkuva data kulkisi täsmällisesti. (kuva 10)



Kuva 10. NTP-testistä syntynyt kuvaaja palvelimelle

4.2 Palvelimet

Jokaiselle palvelimelle kerääntyi siis sama määrä testejä, joiden testitulokset siirrettiin erilliseen taulukkoon. Taulukossa (taulukko 1) oli sekä sijoituspaikka-, että palvelinkohtaisesti luokiteltuna onnistuneet ja epäonnistuneet testaustulokset, joista sai helposti selville, missä sijoituspaikoissa on tehtävä vielä muutoksia uutta yhteyttä varten.

Taulukko 1. Erään palvelimen testitulokset sijoituspaikoittain.

													AVERAGE
NTP	s		-0,16				-0,26			-0,13	-4,04	-0,24	-9,98	-2,47
PING RTA	ms	avg	7,95	15,26	23,97	18,26	9,56	11,99			17,79	19,14	16,51	15,60
		max	80,95	243,17	67,75	44,17	29,30	58,5			68,50	33,60	19,20	71,68
PING LOSS	%	avg	0,06	0,11	0,04	0,07	0,05	29,24			0,09	13,56	0,14	4,82
		max	8,00	8,43	6,35	8,83	8,70	59,5			15,00	47,00	9,40	19,02
UDP	ms	avg												
		max												
TCP	ms		9,16	20,81			10,77	21,54			16,98		18,53	16,30
TCP	ms		9,10	17,97			12,26	59,28			19,73		18,92	22,87
TCP	ms		10,32	26,21			13,87	43,03			14,81		24,22	22,08
TCP	ms		12,15	19,16			11,92	62,84			19,89		21,29	24,54
TCP	ms		11,22	19,34			11,54	31,77			17,30		19,36	18,42
TCP	ms		12,22	19,75			10,29	31,27			21,10		18,66	18,88
TCP	ms		8,72	24,95			14,17	54,59			14,82		16,18	22,24
TCP	ms		11,45	19,84			12,26	54,72			14,69		20,87	22,30
TCP	ms		11,34	20,48			13,54	48,45			21,75		20,46	22,67
TCP	ms		10,48	20,49			15,55	55,31			16,96		18,56	22,89
average tcp			10,61	20,89			12,36	46,28			17,80		19,71	21,28

LÄHTEET

[1] Silventoinen, Atte. 2012. Lähi- ja reititinverkot. Opinnäytetyö. Tietotekniikan koulutusohjelma. Mikkeli: Mikkelin ammattikorkeakoulu. Saatavissa myös http://publications.theseus.fi/bitstream/handle/10024/40187/Silventoinen_Atte.pdf?sequence=1

[2] Internet Protocol 1981. DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION. Viitattu 21.5.2013 http://www.ietf.org/rfc/rfc791.txt

[3] Internet Control Message Protocol 1981. DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION. Viitattu 21.5.2013 http://www.ietf.org/rfc/rfc792.txt

[4] Kyntölä, Tommi. 2010. Transport Layer VPN Cluster for Electronic Gaming Machines. Opinnäytetyö. Teknillisen fysiikan koulutusohjelma. Espoo: Aalto-yliopisto

[5] Nagios Install Guide. Ubuntu Quickstart. Viitattu 21.5.13 http://nagios.sourceforge.net/docs/3_0/quickstart-ubuntu.html

[6] Adding hosts to Nagios 2010. The Tech Tutorial. Viitattu 21.5.13 http://www.the-tech-tutorial.com/?p=414