

KARELIA-AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma

Erno Hyttinen

ZENOSS-JÄRJESTELMÄN PÄIVITYS JA KONFIGUROINTI

Opinnäytetyö
Marraskuu 2014



OPINNÄYTETYÖ
Marraskuu 2014
Tietotekniikan koulutusohjelma

Karjalankatu 3
80200 JOENSUU
+358 50 260 6800

Tekijä(t)
Erno Hyttinen

Nimeke
Zenoss-järjestelmän päivitys ja konfigurointi

Toimeksiantaja
Pieksämäen kaupunki

Tiivistelmä

Tämän opinnäytetyön tavoitteena oli päivittää Zenoss Core -valvontajärjestelmä säilyttäen mahdollisimman paljon vanhasta toiminnallisuudesta. Opinnäytetyössä käsitellään yleisesti Zenoss-projektia, Zenoss Coren asennusta ja konfigurointia sekä työn aikana ilmenneitä ongelmia.

Opinnäytetyön toimeksiantajana toimi Pieksämäen kaupunki. Alkuperäinen järjestelmä oli pystytetty myös opinnäytetyönä aikaisemmin. Järjestelmää ei ollut kuitenkaan päivitetty kunnollisesti aiemmin. Päivityksen tarkoituksena oli sekä päivittää tämä järjestelmä uusimpaan versioonsa että asentaa järjestelmä omalle erilliselle palvelimelleen, joka tarjoaisi kaivattua nopeutta järjestelmän toimintaan.

Tuloksena saatiin uusi hyvin ja sujuvasti toimiva järjestelmä, joka täytti toimeksiantajan asettamat tavoitteet. Työn aikana törmättiin myös erilaisiin ongelmiin, mutta ne onnistuttiin kuitenkin onnistuneesti ratkaisemaan. Lopuksi kuitenkin todettiin, että paremmalla suunnittelulla sekä toimeksiantajan että opinnäytetyön tekijän osalta olisi voitu päästä parempaan lopputulokseen.

Kieli
suomi

Sivuja 39

Asiasanat
monitorointi, palvelimet, päivitys, verkonhallinta



THESIS
November 2014
Degree Programme in
Information Technology
Karjalankatu 3
FI 80200 JOENSUU
FINLAND
+358 50 260 6800

Author(s)
Erno Hyttinen

Title
The Upgrade and Configuration of Zenoss Platform

Commissioned by
town of Pieksämäki

Abstract

The objective of this thesis was to upgrade the Zenoss Core monitoring platform while preserving as much as possible from the previous functionality. This thesis covers the Zenoss project in general, installation and configuration of Zenoss Core and the problems which occurred during the project.

The thesis was commissioned by the town of Pieksämäki. The initial system was also set up as a thesis before. The system has not been properly upgraded since. The purpose of the upgrade was to upgrade the aforesaid system and also place the system on a separate server which would grant much needed speed to the system's operation.

As a result, a new and swiftly working system was established which met the client's requirements. Several problems were encountered during the project but they were successfully resolved. Ultimately, it can be stated that with better planning from the client as well as from the author the project could have ended with even better results.

Language
Finnish

Pages 39

Keywords
monitoring, servers, upgrade, network management

Sisältö

1	Johdanto	6
2	Zenoss Core.....	7
2.1	Zenoss Coren historia	7
2.2	Zenoss Coren ominaisuudet	7
2.2.1	Zenoss Coren alustavaihtoehdot.....	8
2.2.2	ZenPack-lisäosat.....	9
3	Zenossin asennus ja konfigurointi	9
3.1	Nykyisen järjestelmän päivitettävyys	9
3.2	Uuden järjestelmän asennus.....	10
3.3	Uuden järjestelmän konfigurointi	17
3.3.1	Järjestelmän asennuksen jälkeiset toimenpiteet	17
3.3.2	Järjestelmän lisäkonfiguraatiot	21
3.3.3	Zenoss Coren konfigurointi	26
4	Ongelmat.....	32
4.1	Käyttöalusta- ja ZenPack-ongelmat	33
4.2	SNMP- ja WMI-ongelmat.....	34
4.3	Bad Counter -virheet	35
4.4	Zenwinperf-palvelun ongelma	36
4.5	VMware ESX -palvelimien seurannan ongelma	37
5	Tulokset.....	37
6	Pohdinta	38
	Lähteet	39

Lyhenteet

FTP	File Transfer Protocol, menetelmä, jota käytetään tiedonsiirtoon kahden tietokoneen välillä.
HTTP	Hypertext Transfer Protocol, menetelmä, jota käytetään selainten ja web-palvelimien väliseen tiedonsiirtoon.
NNTP	Network News Transfer Protocol, menetelmä, jota käytetään uutisryhmien lukemiseen ja kirjoittamiseen, sekä sisällön siirtoon uutispalvelimien välillä.
NTP	Network Time Protocol, menetelmä, jota käytetään täsmällisen aikatiedon välittämiseen tietokoneiden välillä.
POP3	Post Office Protocol version 3, menetelmä, jota käytetään sähköpostin hakemiseen.
RAID	Redundant Array of Independent Disks, tekniikka, jolla kasvatetaan tietokoneiden, erityisesti palvelinten vikasietoisuutta.
RSA	Tekijöidensä (Rivest, Shamir ja Adelman) mukaan nimetty julkisen avaimen salausalgoritmi.
SNMP	Simple Network Management Protocol, menetelmä, jota käytetään verkossa olevien laitteiden tilojen kyselyihin.
SSH	Secure Shell, menetelmä, jota käytetään mm. salatun etäyhteyden saamiseen palvelimille.
SSL	Secure Sockets Layer, menetelmä, jota käytetään Internet-tietoliikenteen salaamiseen IP-verkkojen yli.
WMI	Windows Management Instrumentation on Windows-pohjaisten käyttöjärjestelmien käyttämä infrastruktuuri tiedon- ja toimintojen hallintaan.

1 Johdanto

Tämän opinnäytetyön tarkoituksena oli päivittää toimeksiantajalle tällä oleva verkkolaitteiden valvontajärjestelmä sen uusimpaan versioon, säilyttäen aikaisempi toiminnallisuus mahdollisimman laajalti sekä tehdä mahdollisesti tarpeellisia muutoksia. Päivityksen yhteydessä järjestelmä siirretään pois virtuaaliselta alustalta, ja se asennettaisiin omalle palvelinlaitteelle. Tällä muutoksella halutaan parantaa käyttökokemusta ja järjestelmän käytettävyyttä.

Zenoss Core (käytetään myös käyttäjiensä keskuudessa pelkästään nimitystä Zenoss) on avoimeen lähdekoodiin perustuva verkkoresurssien (kuten verkon aktiivilaitteet sekä palvelimet) valvonta-alusta, jota hallitaan web-käyttöliittymän avulla. Alusta mahdollistaa erilaisten laitteiden suorituskykyyn ja toimintaan liittyvien komponenttien seurannan. Järjestelmän pääominaisuuksiin kuuluu erityisesti tapahtumien valvonta, joka ilmoittaa seurattavien laitteiden komponenttien tai ohjelmistojen tilasta, kuten mahdollisista virheistä.

Järjestelmä on alun perin pystytetty toimeksiantajalle opinnäytetyönä vuonna 2009. Tuolloin kyseisessä työssä toimeksiantajalle etsittiin sopivaa vapaaseen lähdekoodiin perustuvaa valvontajärjestelmää. Päätös tehtiin neljän eri järjestelmän väliltä, joista Zenoss valittiin sen ollessa kaikista ehdokkaista pätevin toimeksiantajan käyttötarkoituksiin. Järjestelmä valittiin, koska se täytti toimeksiantajan vaatimukset, ja muihin vertailun kohteena olleisiin järjestelmiin verrattuna Zenoss tarjosi kattavan dokumentaation. Sen vahvuuksiksi todettiin aktiivinen ja kookas käyttäjäyhteisö, monipuolinen laajennusvara, hyvä ohjeistus, monipuoliset hälytys- ja raportointitoiminnot sekä miellyttävä ja selkeä käyttöliittymä ja ulkoasu (Rautiainen 2009, 16).

Työssä käsitellyjä toimeksiantajan arkaluontoisia tietoja, kuten IP-osoitteita, salasanoja, sekä laitteiden domainnimiä ei työssä esitetä tietoturvan vuoksi.

2 Zenoss Core

2.1 Zenoss Coren historia

Zenoss Core -projektin aloittajana toimi tietoliikenteeseen keskittynyt insinööri Erik Dahl. Vuonna 2005 Erik Dahl tapasi vanhan kollegansa Bill Karpovichin ja yhdessä he päättivät perustaa Zenoss Inc. -yhtiön. Pian tämän jälkeen he julkaisivat ensimmäisen version Zenoss Core -ohjelmistosta GNU GPLv2 -lisenssin alaisena. (Zenoss Wiki 2013)

Zenoss Coren synty perustuu projektin aloittajan Erik Dahlin näkemykseen. Hän ei ollut vakuuttunut silloisesta verkon valvonta- ja hallintaohjelmistojen tilanteesta, jossa tarvitaan useita eri työkaluja esimerkiksi laitteiden saatavuuden valvontaan, suorituskyvyn valvontaan sekä tapahtumien valvontaan. (Zenoss Wiki 2013)

Zenossin historiaan mahtuu neljä suurempaa versiopäivitystä. Projektin aloituksen jälkeen julkaistiin versio 1.0, kesäkuussa 2007 versio 2.0, heinäkuussa 2010 versio 3.0 ja vuonna 2011 versio 4.0. Kirjoitushetkellä Zenoss Coren viimeisin ohjelmistoversio on 4.2.5, joka näki päivänvalon huhtikuussa 2014. (Zenoss Inc. 2014)

Nykyisin Zenoss Core on toinen Zenoss Inc. -yhtiön tarjoamista ohjelmistovaihtoehtoista. Zenoss Service Dynamics on Zenoss Coren laajennettu kaupallinen versio, joka vastaa jopa suurempien yritysten tarpeisiin. (Zenoss Inc. 2014)

2.2 Zenoss Coren ominaisuudet

Zenoss Core koostuu pohjaltaan moniin vapaan lähdekoodin tekniikoihin kuten Zope-sovelluspalvelimeen, Python-ohjelmointikieleen, Net-SNMP-ohjelmistoon, RRDtool-aineistonkeräystyökaluun, MySQL-tietokantaan, Twisted-ohjelmistokehykseen sekä Lucene-ohjelmakirjastoon. (Wikipedia 2014)

Näistä Zope on sovelluspalvelin, jonka päälle Zenoss on toteutettu. Zopelle tyypillinen piirre on ZODB-oliotietokanta, jota hyödynnetään myös Zenossissa. Pythonia vuorostaan hyödynnetään jo Zopessa, joten sen käyttö on luonnollista myös muissa Zenoss Coren osissa. Ohjelmiston moduulit on poikkeuksetta kaikki ohjelmoitu Pythonilla. Net-SNMP on sovellus, joka tarjoaa ominaisuudet SNMP-protokollan käyttöönottamiseen Unix-pohjaisissa järjestelmissä. Zenoss hyödyntää Net-SNMP:n tarjoamia ominaisuuksia toiminnassaan. RRDtool on työkalu, jonka avulla pyritään keräämään aikasarja dataa esimerkiksi prosessorin käyttöasteesta, verkon kaistanleveydestä sekä lämpötiloista. Zenoss Coren tiedonkeräysominaisuudet toimivat RRDtool-työkalun pohjalta. MySQL vuorostaan on yksi maailman laajimmin käytetyistä relaatiotietokantahallintajärjestelmistä. Zenoss käyttää MySQL-tietokantaa tietokantana kaikille verkon ja sovelluksen tapahtumille.

Zenossin tarjoamiin ydinominaisuuksiin kuuluvat erityisesti

- verkkolaitteiden valvonta SNMP-, SSH- ja WMI-protokollien avulla
- verkkopalveluiden valvonta HTTP-, POP3-, NNTP-, SNMP- ja FTP-protokollien avulla
- isännän resurssien monitorointi, esimerkkinä prosessorin ja levyn käyttöaste
- tapahtumienhallintatyökalut hälytysten esilletuomista varten
- automaattinen verkkoresurssien selvitys ja muutosten havainnointi verkon kokoonpanossa. (Wikipedia 2014)

2.2.1 Zenoss Coren alustavaihtoehdot

Zenoss Core on alustavaihtoehdoiltaan monipuolinen. Sen alustavaihtoehtoihin kuuluu useita Linux-jakeluita kuten Debian (Ubuntu), Fedora (RHEL, CentOS), Gentoo ja openSUSE (SuSE) sekä muita alustoja kuten Mac OS X, VMware, Solaris ja FreeBSD.

Virallisesti tuettuja alustoja ovat RHEL (RedHat Enterprise Linux) sekä CentOS. Muiden alustojen tuki on epävirallista ja tulee Zenoss-yhteisön kautta.

2.2.2 ZenPack-lisäosat

Ohjelmistossa on valmiina jo laajalti ominaisuuksia, mutta yksilöityä toiminnallisuuden laajentamista auttavat jälkikäteen lisättävät ZenPack-lisäosat.

ZenPackit ovat siis moduuleja ja ne voivat olla yhteisön käyttäjien itsensä tekemiä tai Zenoss Inc. -yhtiön tekemiä kaupallisia tai ei-kaupallisia lisäosia. ZenPackeilla voidaan laajentaa järjestelmän toiminnallisuutta haluttuun suuntaan, esimerkiksi lisätä järjestelmään valmistajakohtaisia laite- ja protokollatukia.

Saatavilla on kirjoitushetkellä 381 erilaista ZenPackia (Zenoss 2014). Kaikki ei-kaupalliset Zenoss Inc. yhtiön itsensä tekemät ZenPackit kuuluvat automaattisesti Zenoss Coren täyteen asennukseen. Zenossin kaupalliset ZenPackit ovat kuitenkin saatavilla vain Zenossin kaupalliseen Zenoss Service Dynamics versioon. (Zenoss Inc. 2014)

3 Zenossin asennus ja konfigurointi

3.1 Vanhan järjestelmän päivitettävyyden

Tämän opinnäytetyön yksi suurimpia kysymyksiä oli nykyisen järjestelmän päivitettävyyden. Kuinka ja miten järjestelmä päivitetään?

Työssä oli tarkoitus päivittää nykyinen järjestelmä uuteen versioon siirtäen ja asentamalla järjestelmä uudelle alustalle sekä säilyttäen mahdollisimman paljon nykyisestä toiminnallisuudesta.

Ennen varsinaista asennustyötä oli tarkoituksena selvittää, voiko nykyistä järjestelmää hyödyntää jotenkin päivityksessä. Zenoss Coren dokumentaatio verkos-

sa tarjosi esimerkkiratkaisuja käytössä olleen 3.x-version päivittämisestä 4.x-versioon (Zenoss Wiki 2014), mutta käytännön kokeiluissa tämän projektin aikana tässä epäonnistuttiin. Käyttöliittymältään uusin versio vastaa lähes täysin vanhempaa versiota, mutta sisäiset muutokset ovat olleet suurempia, sillä yritykset päivittää järjestelmää vanhemman version tietokannoilla ja ominaisuuksilla johtivat vain ongelmiin ja virheilmoituksiin.

Lopulta päädyttiin ratkaisuun, jossa uusi versio asennetaan puhtaalta pöydältä ja tarvittavat ominaisuudet ja sisältö tiputetaan uuteen järjestelmään manuaalisesti niin laajalti kuin mahdollista. Edellä mainituista sekä muista ilmenneistä ongelmista työn aikana kerrotaan lisää myöhemmin.

3.2 Uuden järjestelmän asennus

Zenoss Coren puhdasta asennusta varten Zenossin dokumentit tarjoavat hyvät ohjeet (Sourceforge 2013). Asennukseen käytettiin kuitenkin Marc Weiselin tekemää ohjetta (Weisel 2014), koska sitä oli allekirjoittaneen mielestä helpompi seurata. Ohjeen tyyli oli enemmän sopiva juuri Zenoss Coren ensi kertaa asennettavalle. Weiselin ohjeesta kuitenkin poiketen tietokantaa asennettaessa asennettiin perinteinen MySQL, ei MariaDB:tä.

Manuaalisen tavan lisäksi asennus on mahdollista suorittaa Zenossin tekemällä automaattisella asennuskriptillä, jota Zenoss tarjoaa tuetuille alustoille, mutta tähän liittyvien ohjeiden seuraamisesta huolimatta automaattinen asennus ei tuottanut tulosta vaan johti aina virhekoodiin. Tämän vuoksi päädyttiin asentamaan Zenoss käsin.

Zenoss Coren asennusta varten varattiin Dell PowerEdge -palvelin. Palvelin sisältää Intelin Xeon -prosessorin ja neljä gigatavua keskusmuistia. Palvelimella oli valmiiksi konfiguroituna kaksi RAIDia, joista pienemmälle Zenoss Core asennettiin. Isommalle RAIDille osoitettiin lopulta Zenoss Coren tietokanta.

Käyttöjärjestelmäksi valikoitui 64-bittinen CentOS 6.5. Zenoss Core on 4. versiostaan asti ainoastaan 64-bittinen, joten se tulee ottaa huomioon palvelinrautaa valittaessa.

CentOS-järjestelmän asentaminen on melko helppoa ja suoraviivaista. Järjestelmän asennus käynnistyy levyltä graafiseen käyttöliittymään, jonka kautta on hiirellä helppo tehdä asennusvalinnat. Tämän työn tarkoitukseen CentOS asennettiin minimal-asetuksella. Tämä tarkoittaa sitä, että vain kaikkein tärkeimmät ja keskeisimmät järjestelmän osat ja työkalut tulevat asennetuiksi. Kaikki, mikä ei ole järjestelmän käytön kannalta olennaista, jää siis asennuksesta pois. Asennuksen yhteydessä on myös helppo syöttää verkkokortille halutut IP-asetukset; IP-osoite, aliverkon peite ja oletusyhdyskäytävä sekä toimialue. Tämä helpottaa hieman asennusta, sillä verkkoyhteys on jo valmiina asennuksen jälkeen, eikä työtä vaadita enää kyseisten asetusten laittamiseen jälkikäteen.

Asennuksen aikana toimet on hyvä tehdä root-oikeuksilla, esimerkiksi suoraan root-käyttäjällä, sillä monet komennot vaativat root-oikeuksia.

Asennuksen jälkeen varmistetaan ensimmäisenä, että uusimmat päivitykset jo asennetuille osille tulevat asennetuiksi komendoilla

```
yum -y update  
yum -y upgrade
```

Tämän jälkeen asennetaan ohjeen mukaisesti muutama lisätyökalu:

```
yum -y install file ntp vim-enhanced man man-pages wget  
traceroute yum-utils
```

Edellä mainittujen asennuksen jälkeen asennettiin vielä nano-ohjelma helpottamaan ja nopeuttamaan komentorivipohjaista tekstinkäsittelyä. Järjestelmästä löytyy valmiina myös vim-sovellus perusteellisempaa tekstinkäsittelyä varten, joten nanon lataaminen ei ole pakollista.

Käyttöjärjestelmän asennuksen yhteydessä asetettiin jo palvelimelle nimi, mutta varmistetaan hosts-tiedostosta, että palvelimen IP-osoite ja nimi löytyvät sieltä.

```
nano /etc/hosts
```

Tarvittaessa lisätään oletusrivien perään asetukset muodossa IP-osoite, pitkä domainnimi (FQDN) sekä lyhyt nimi, esimerkiksi (sama kuin Weiselin ohjeessa):

```
10.1.206.43 zenoss.corp.example.com zenoss
```

Tämän jälkeen konfiguroidaan SELinux tarvittavalla tavalla. Zenossin dokumentaatio vaatii, että SELinux tulee poistaa käytöstä. Tämä voidaan tehdä vaihtamalla SELINUX-arvoa tiedostosta:

```
nano /etc/sysconfig/selinux
```

Vaihdetaan arvoksi "disabled", eli pois käytöstä. Tallennetaan muutokset ja muutoksen voimaantulemiseksi palvelin tulee käynnistää uudelleen. Tämän voi tehdä nopeasti komennolla "reboot". Uudelleenkäynnistämisen jälkeen tarkastetaan komennolla "sestatus" SELinuxin tila. Komennon pitäisi vastata, että SELinuxin tila on poissa käytöstä.

Seuraavaksi varmistetaan, että NTP-palvelu toimii asettamalla halutut NTP-synkronointipalvelimet asetustiedostoon:

```
nano /etc/ntp.conf
```

NTP:n tarkoituksena on yksinkertaisesti pitää järjestelmän kello oikeassa ajassa. Pieksämäen kaupungin tilanteessa poistetaan oletukset pois käytöstä kommentoimalla ne pois ja asettamalla paikallisen NTP-palvelimen osoite niiden alle. Alla esimerkki (kuten Weiselin ohjeessa):

```
#server 0.centos.pool.ntp.org
```

```
#server 1.centos.pool.ntp.org
```

```
#server 2.centos.pool.ntp.org
```

```
server 10.1.206.11 iburst
```

Jos omaa NTP-palvelinta ei ole saatavilla, myös julkisia NTP-palvelimia, kuten edellä poiskommentoituja CentOSin NTP-palvelimia, on mahdollista käyttää.

Käynnistetään seuraavaksi NTP-palvelu:

```
service ntpd start
```

Asetetaan NTP-palvelu myös käynnistymään automaattisesti järjestelmän käynnistyksen yhteydessä:

```
chkconfig ntpd on; chkconfig --list ntpd
```

NTP:n tilan voi tarkistaa helposti komennolla "ntpd -pn".

Seuraava vaihe on valmistella palomuuuri Zenoss Corea varten. Zenossin toiminta edellyttää tiettyjen porttien sallimista palomuurista. Porttia numero 11211 tarvitsee memcached-sovellus, porttia 8080 web-käyttöliittymä, porttia 514 syslog, porttia 162 SNMP-trapit sekä porttia 25 zenmail-sovellus. Helpoin tapa asettaa portit haluttuun tilaan on muuttaa iptables-tiedostoa:

```
nano /etc/sysconfig/iptables
```

Asetetaan rivin "-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT" jälkeen seuraavat rivit ja tallennetaan muutokset:

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 11211 -j ACCEPT  
-A INPUT -m state --state NEW -m udp -p udp --dport 11211 -j ACCEPT
```

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8080 -j AC-  
CEPT  
-A INPUT -m state --state NEW -m udp -p udp --dport 514 -j AC-  
CEPT  
-A INPUT -m state --state NEW -m udp -p udp --dport 162 -j AC-  
CEPT  
-A INPUT -m state --state NEW -m tcp -p tcp --dport 25 -j ACCEPT
```

Käynnistetään palomuuuri uudestaan muutosten käyttöön ottamiseksi:

```
service iptables restart
```

Tämän jälkeen tarvitaan palvelimelle Java 6. Versio on jo nyt erittäin vanha, eikä Javan sivustoilta ole vierailevan käyttäjän helppoa saada tarvittavaa asennuspakettia käsiinsä useiden varoitustekstien ja -sivujen takaa. Zenoss kuitenkin tarvitsee erityisesti Java 6 -paketin, joten vaihtoehtoa ei ole. Oikea paketti on nimeltään "jre-6u45-linux-x64-rpm.bin". Ladataan paketti Windows-koneelle ja hyödynnetään erikseen Windowsille ladattua WinSCP-sovellusta tiedoston siirtämiseen SSH-yhteyden yli Linux-palvelimelle. Siirron jälkeen palataan palvelimelle ja muutetaan tiedoston oikeuksia ennen asennusta:

```
chmod u+x ./jre-6u45-linux-x64-rpm.bin
```

Suoritetaan seuraavaksi Javan asennus. Asennuksen pitäisi onnistua automaattisesti ja virheettömästi:

```
./jre-6u45-linux-x64-rpm.bin
```

Asetetaan seuraavaksi JAVA_HOME-muuttuja järjestelmän bash-profiiliin komennolla:

```
echo 'export JAVA_HOME=/usr/java/default' >> /etc/profile
```

Luetaan muuttuja ja otetaan se käyttöön:

```
./etc/profile
```

Javan toimivuuden voi nyt tarkistaa antamalla komennot "echo JAVA_HOME" ja "java -version". Ensimmäisen tulisi vastata antamalla Javan asennussijainti ja jälkimmäisen tulisi antaa Java-version tiedot.

Seuraavaksi asennetaan MySQL-tietokanta. Haetaan ensimmäisenä MySQL:n kotisivuilta 64-bittiset RPM-paketit "Client Utilities", "MySQL Server" ja "Shared Components". Ladataan tiedostot esimerkiksi Windows-koneelle ja siirretään ne jälleen palvelimelle WinSCP-sovelluksen avulla.

Asennus voidaan aloittaa seuraavaksi komennolla

```
yum -y --nogpgcheck localinstall MySQL*
```

Asennuksessa huomioitavaa on se, ettei vielä tässä vaiheessa tule asettaa tietokannan root-käyttäjälle salasanaa. Tämä tulee asettaa vasta Zenossin asennuksen jälkeen.

Kun asennus on suoritettu, luodaan seuraavaksi uusi tiedosto ja muokataan sitä:

```
touch /etc/my.cnf
```

```
nano /etc/my.cnf
```

Asetetaan tekstitiedostoon seuraavat rivit ja tallennetaan:

```
[mysqld]
```

```
max_allowed_packet=16M
```

```
innodb_buffer_pool_size=256M
```

```
innodb_additional_mem_pool_size=20M
```

Käynnistetään MySQL ja asetetaan se käynnistymään automaattisesti järjestelmän käynnistyessä seuraavilla komennoilla

```
service mysql start  
chkconfig --add mysql  
chkconfig --level 2345 mysql on
```

Zenossin asennusta varten varmistetaan seuraavaksi, että MySQL:n salasanat ovat tyhjiä (heittomerkkien väliin ei tule laittaa välilyöntiä komentoja annettaessa):

```
mysqladmin -u root password ''  
mysqladmin -u root -h localhost password ''
```

Tämän jälkeen päästäänkin asentamaan itse Zenoss Core. Haetaan ensimmäisenä uusin RHEL/CentOS Zenoss 4 RPM -paketti Zenossin kotisivuilta. Työtä suoritettaessa uusin versio oli 4.2.4. Ladataan paketti esimerkiksi Windows-koneelle ja siirretään se palvelimelle WinSCP:n avulla. Siirrytään takaisin palvelimelle ja asetetaan ensimmäisenä Zenossin riippuvuuksien tietolähteet seuraavalla komennolla:

```
rpm -Uvh http://deps.zenoss.com/yum/zenossdeps-4.2.x-1.el6.noarch.rpm
```

Tämä komento hakee verkosta paketin, joka asettaa yum-sovelluksen tietolähteisiin zenossdeps-repo sekä zenoss-update-repo tietolähteet. Tietolähteet voi näyttää komennolla "yum repolist".

Asennetaan seuraavaksi aikaisemmin haettu ja siirretty Zenoss Core 4 RPM-paketti, esimerkiksi:

```
yum -y --nogpgcheck localinstall zenoss_core-4.2.4-1897.el6.x86_64.rpm
```


Zenossin asennuksen pitäisi sujua ongelmitta. Asennus hakee myös tarpeelliset riippuvuudet automaattisesti asennuksen aikana.

Kun Zenoss Coren asennus on saatu päätökseen, asetetaan muutama palvelu nyt käynnistymään automaattisesti järjestelmän käynnistyksen yhteydessä (komento kokonaisuudessaan yhdelle riville):

```
for svc in memcached rabbitmq-server snmpd; do chkconfig $svc  
on; service $svc start; done
```

Zenoss voidaan nyt käynnistää ensimmäistä kertaa komennolla "service zenoss start". Ensimmäisen käynnistyksen yhteydessä Zenoss suorittaa erilaisia toimenpiteitä ja käynnistys kestää hiukan normaalia kauemmin. Palveluiden käynnistyttyä Zenoss Coreen on mahdollista tutustua selaimen kautta osoitteella <http://<Palvelimen IP-osoite>:8080>. Ensimmäisen käyttökerran yhteydessä pyydetään asettamaan esimerkiksi hallintakäyttäjän tiedot kirjautumista varten.

3.3 Uuden järjestelmän konfigurointi

3.3.1 Järjestelmän asennuksen jälkeiset toimenpiteet

Zenossin perusasennus on nyt suoritettu, mutta asennuksen jälkeiseksi toimenpiteeksi jää Zenoss-asennuksen salaaminen.

Vaihdetaan aktiivisena olevaa käyttäjää root-käyttäjältä zenoss-käyttäjälle:

```
su -l zenoss
```

Haetaan `secure_zenoss.sh`-tiedosto GitHubista komennolla:

```
wget --no-check-certificate https://raw.githubusercontent.com/osu-sig/zenoss-  
autodeploy-4.2.3/master/secure_zenoss.sh
```

Katsotaan ensimmäisenä oletussalasanat Zenossin global.conf-tiedostosta. Ne voidaan näyttää komennolla

```
egrep 'user|password' $ZENHOME/etc/global.conf | grep -v admin
```

Komento tulostaa näkyviin lyhyen listan, josta ilmenevät myös oletussalasanat, jotka ovat mitä todennäköisimmin "zenoss".

Annetaan suoritusoikeudet secure_zenoss.sh-skriptille:

```
chmod u+x secure_zenoss.sh
```

Käynnistetään skripti komennolla ./secure_zenoss.sh. Skripti ilmoittaa lopuksi, että MySQL:n root-salasana on tyhjä ja kysyy asetetaanko se. Jätetään salasana asettamatta vielä tässä vaiheessa antamalla vastaukseksi kysymykseen "n".

Tarkistetaan nyt muutokset salasanoihin aikaisemmalla komennolla "egrep 'user|password' \$ZENHOME/etc/global.conf | grep -v admin". Salasanat ovat nyt muuttuneet sattumanvaraiseksi merkkijonoksi. Otetaan tämä merkkijono talteen esimerkiksi kopioimalla, sillä sitä tarvitaan vielä. Uusien salasanojen tulee löytyä myös zodb_db_main.conf- sekä zodb_db_session.conf-tiedostoista. Seuraavalla komennolla voidaan nähdä, että kyseisissä tiedostoissa salasana on vielä oletusmuodossaan:

```
tail -n +1 $ZENHOME/etc/zodb_db_{main,session}.conf
```

Seuraavien komentojen avulla voidaan sijoittaa nopeasti uusi salasana äsken mainittuihin tiedostoihin oletussalasanojen tilalle:

```
zodbpw=$(grep zodb-password $ZENHOME/etc/global.conf | awk  
'{print $2}')  
sed -i.orig "5s/zenoss/$zodbpw/" $ZENHO-  
ME/etc/zodb_db_{main,session}.conf  
unset -v zodbpw
```

Aikaisemmalla komennolla voidaan nyt tarkastaa, että salasanat tulivat sijoitetuiksi (tail -n +1 \$ZENHOME/etc/zodb_db_{main,session}.conf). Lopuksi poistutaan zenoss-käyttäjältä komennolla "exit".

Salataan seuraavaksi MySQL. Tämän voi tehdä helpoimmin MySQL:stä löytyvällä työkalulla:

mysql_secure_installation

Ohjattu toiminto esittää suoritettaessa suojaukseen liittyviä kysymyksiä, joihin kannattaa vastata omien tarpeiden mukaan, esimerkiksi asettamalla haluttu root-salasana. Suojauksen jälkeen kirjaudutaan MySQL-komentoriville:

mysql -u root -p

Annetaan kirjautuessa root-salasana. Aikaisemmin talteen otettua salasanaa tarvitaan seuraavaksi. Tietokannan zenoss-käyttäjän salasana tulee vastata aikaisemmin määriteltystä Zenossin salasanoja. Tämä voidaan tehdä MySQL:n komentorivillä seuraavalla komennolla asettamalla sulkuihin heittomerkkien sisälle oma salasana, esimerkiksi:

```
SET PASSWORD FOR 'zenoss'@'localhost' = PASSWORD('18zmcTgYsA+AjczljwQd');
```

Poistutaan MySQL-komentorivillä komennolla \q. Käynnistetään tämän jälkeen MySQL uudelleen:

service mysql restart

Luodaan seuraavaksi skripti, joka varmistaa, että RabbitMQ-server toimii halutulla tavalla.

nano set-rabbitmq-perms.sh

Kopioidaan seuraavat rivit luotuun tiedostoon ja tallennetaan:

```
#!/usr/bin/env bash

set -e

VHOSTS="/zenoss"
USER="zenoss"
PASS="grep amqppassword \${ZENHOME}/etc/global.conf | awk
'{print \$2}'"

if [ $(id -u) -eq 0 ]
then
    RABBITMQCTL=$(which rabbitmqctl)
    $RABBITMQCTL stop_app
    $RABBITMQCTL reset
    $RABBITMQCTL start_app
    $RABBITMQCTL add_user "$USER" "$(su -l zenoss -c
"$PASS")"

    for vhost in $VHOSTS; do
        $RABBITMQCTL add_vhost "$vhost"
        $RABBITMQCTL set_permissions -p "$vhost" "$USER" '.*' '.*'
'.'
    done
    exit 0
else
    echo "Error: Run this script as the root user." >&2
    exit 1
fi
```

Annetaan luodulle skriptille suoritusoikeudet ja ajetaan se:

```
chmod u+x set-rabbitmq-perms.sh  
./set-rabbitmq-perms.sh
```

Käynnistetään tämän jälkeen RabbitMQ-server uudelleen:

```
service rabbitmq-server restart
```

Kaikkien asennusten jälkeisten muutosten käyttöönottamiseksi käynnistetään vielä Zenoss Core uudelleen ja tämän jälkeen tarkistetaan, että sovelluksen kaikki prosessit ovat käynnissä:

```
service zenoss restart  
su -l zenoss -c 'zenoss status'
```

Jos kaikki on kunnossa, Zenoss Coren asennus on valmis. Tämän jälkeen suoritettavat toimenpiteet ovat vain juuri toimeksiantajalle spesifisiä muutoksia ja asetuksia.

3.3.2 Järjestelmän lisäkonfiguraatiot

Kun Zenoss Coren asennus on nyt saatu valmiiksi, on aika ryhtyä yksilöimään palvelinta toimeksiantajan tarpeisiin. Kuten aikaisemmin todettiin, tarkoituksena oli säilyttää toiminnallisuus hyvin pitkälti samankaltaisena kuin alkuperäisessä järjestelmässä. Tämän vuoksi katsottiin Rautiaisen opinnäytetyön avulla vaiheita, jotka suoritetaan asennuksen jälkeen. Ensimmäisenä asennetaan käyttöön Apache web-palvelin. Rautiaisen työssä Apache asennettiin toimimaan välityspalvelimena, joka tarjoaa Zenoss Coreen suojatun yhteyden ja varmenteen, joten samaa hyödynnetään myös nyt.

Rautiaisen työhön verrattuna erona on eri Linux-jakelun käyttäminen (aikaisemmin Debian), jonka vuoksi jotkin komennot ovat erilaisia.

Aloitetaan asentamalla Apache sekä tarvittava SSL-lisämoduuli sekä OpenSSL, jos se ei ole jo asennettuna. CentOS-käyttöjärjestelmällä tämän voi tehdä komennolla:

```
yum install httpd mod_ssl openssl
```

Aloitetaan seuraavaksi itse allekirjoitetun varmenteen luominen OpenSSL-työkalun avulla. Debian ja CentOS:n hakemistorakenteen eroavaisuuksien vuoksi varmistetaan, että seuraavat hakemistot ovat olemassa: /etc/ssl/private ja /etc/ssl/certs. Todennäköisimmin jälkimmäinen puuttuu käyttämässämme CentOS-jakelussa joten luodaan se komennolla:

```
mkdir /etc/ssl/certs
```

Luodaan seuraavaksi 2048-bittinen yksityinen RSA-avain.

```
openssl genrsa -out /etc/ssl/private/privkey.pem 2048
```

Seuraavaksi luodaan varmennepyyntötiedosto hyödyntäen äsken luotua avainta. Komennon antamisen jälkeen kysytään varmenteelle annettavia tietoja, johon on hyvä täyttää varmenteen haltija, organisaatio jne.

```
openssl req -new -key /etc/ssl/private/privkey.pem -out  
/etc/ssl/certs/cert.csr
```

Koska palvelin on vain organisaation sisäisessä käytössä, voidaan varmennepyyntötiedosto allekirjoittaa itse sen sijaan, että se lähetettäisiin allekirjoitettavaksi jollekin ulkopuoliselle taholle. Tehdään tämä seuraavalla komennolla:

```
openssl x509 -req -days 1825 -in /etc/ssl/certs/cert.csr -signkey  
/etc/ssl/private/privkey.pem -out /etc/ssl/certs/server.crt
```

Komento luo itseallekirjoitetun ja viisi vuotta voimassa olevan varmenteen.

Muunnetaan seuraavaksi Apache välityspalvelimeksi. Avataan ensin ssl.conf-tiedosto:

```
nano /etc/httpd/conf.d/ssl.conf
```

Etsitään tiedostosta kohta "SSLEngine", jonka jälkeen muokataan tai lisätään rivejä tarvittaessa seuraavasti asettaen myös palvelimen IP-osoite lopussa tarvittaviin kohtiin (Huom. RewriteRule kokonaisuudessaan yhdelle riville):

```
SSLEngine on
SSLProxyEngine on
SSLCertificateFile /etc/ssl/certs/server.crt
SSLCertificateKeyFile /etc/ssl/private/privkey.pem
RewriteEngine On
RewriteRule ^/(.*) http://<Palvelimen IP-osoite>:
8080/VirtualHostBase/https/<Palvelimen IP-osoite>:
443/VirtualHostRoot/$1 [L,P]
```

Tallennetaan tehdyt muutokset tiedostoon. Avataan seuraavaksi httpd.conf-tiedosto:

```
nano /etc/httpd/conf/httpd.conf
```

Lisätään myös tähän tiedostoon muutama rivi. Tarkistetaan samalla, että SSL-moduulin latausmääritys löytyy konfiguraatiosta: "LoadModule ssl_module ..."

```
ServerName <Palvelimen FQDN-nimi>
Redirect permanent / https://<Palvelimen IP-osoite>
```

Lisätään vielä palomuurin iptables-konfiguraatioon (/etc/sysconfig/iptables) seuraavat rivit aikaisempien sinne tehtyjen muutosten yläpuolelle:

```
-A INPUT -m state --state NEW -p tcp --dport 80 -j ACCEPT
-A INPUT -m state --state NEW -p tcp --dport 443 -j ACCEPT
```

Käynnistetään vielä Apache uudestaan komennolla "restart service httpd" Näiden määrytyksien jälkeen Zenoss Coren käyttöliittymään saa varmennetun yhteyden navigoimalla selaimella pelkästään palvelimen IP-osoitteeseen. Kaikki HTTP-pyynnöt ohjataan nyt automaattisesti porttiin 443.

Seuraavaksi siirretään MySQL-tietokanta toiselle kiintolevylle. Järjestelmä asennettiin pienemmälle RAID-konfiguraatiolle, jotta isompi RAID saataisiin kokonaan tietokantaa varten. Tietokanta voi ajan myötä kasvaa suureksikin, joten sitä varten on järkevää varata myös runsaammin tilaa.

Aloitetaan pysäyttämällä MySQL-palvelu komennolla "service mysql stop". Tämän jälkeen tehdään varmuuskopio vanhasta tietokannasta:

```
cp -arv /var/lib/mysql /var/lib/oldmysql
```

Komento kopioi mysql-kansion sisällön, säilyttäen kaikki attribuutit, oldmysql-kansioon. Poistetaan tämän jälkeen /var/lib/mysql-kansio ja luodaan se uudelleen:

```
rm -rf /var/lib/mysql  
mkdir /var/lib/mysql
```

Seuraavaksi tehdään uudelle käyttöön otettavalle asemalle uusi osio. Käytän tähän vaiheeseen Alex Rolekin tekemää ohjetta (Rolek 2013). Minimiasennuksen vuoksi järjestelmästä löytyy valmiina fdisk-työkalu, mutta tarvittava mkfs-työkalu puuttuu joten asennetaan se nyt.

```
yum install mkfs
```

Luodaan ensimmäisenä kiintolevylle uusi koko levyn kattava osio. Komennolla "ls /dev/sd*" voidaan nähdä kaikki liitetyt asemat. Tulosten perusteella voidaan nähdä, että järjestelmässä on esimerkiksi /dev/sda, jonka yksi osio on

/dev/sda1. Tilanteessamme nähdään myös /dev/sdb, mutta ei yhtään osiota kyseiselle medialle. Aloitetaan osion luominen fdisk-työkalulla

```
fdisk /dev/sdb
```

Ohjelma odottaa nyt komentoja. Aloitetaan antamalla komento "c" sekä "u". Työkalu suosittelee näiden komentojen käyttämistä ennen aloittamista. C-komento poistaa DOS-yhteensopivuuden ja u-komento muuntaa käytettävät yksiköt sektoreiksi.

```
Command (m for help): c
DOS Compatibility flag is not set
Command (m for help): u
Changing display/entry units to sectors
```

Annetaan komento "n", eli new. Työkalu kysyy tämän jälkeen toimenpidettä. Komennolla "e" voidaan laajentaa, komennolla "p" luoda uusi ensisijainen osio. Valitaan "p". Tämän jälkeen työkalu kysyy osion numeroa, annetaan arvoksi 1. Seuraavaksi kysytään aloitus- ja lopetus-sektoreita. Annetaan näiden mennä oletuksilla, eli jätetään kohdat tyhjiksi. Silloin osio kattaa koko aseman automaattisesti. Esimerkiksi:

```
Command (m for help): n
Command action
  e extended
  p primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-123456, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-123456, default
123456):
Using default value 123456
```

Tallennetaan tehdyt muutokset komennolla "w" eli write. Komennolla myös automaattisesti poistutaan työkalusta. Formatoidaan seuraavaksi luodulle osiolle tiedostojärjestelmä mkfs-työkalun avulla. Luomamme osio kulkee nyt nimellä /dev/sdb1. Valitaan tiedostojärjestelmäksi ext4, sillä se on Linux-järjestelmissä kaikkein yleisin.

```
mkfs.ext4 /dev/sdb1
```

Komennon antamisen jälkeen työkalu aloittaa tiedostojärjestelmän luomisen ja tulostaa samalla ruudulle tietoa toimistaan. Formatointiin voi kulua jonkin aikaa. Liitetään osio kiinni lopuksi /var/lib/mysql-kansioon.

```
mount /dev/sdb1 /var/lib/mysql
```

Nyt koko toinen kiintolevy on yhdistetty MySQL-tietokannan hakemistoon. Liitos on kuitenkin väliaikainen. Merkintä liittämistä tulee lisätä fstabiin muiden asemien joukkoon, jotta liitos on aina automaattinen järjestelmän käynnistyessä. Avataan tiedosto /etc/fstab esimerkiksi nanoilla, ja lisätään uusi rivi muiden asemien jälkeen:

```
/dev/sdb1 /var/lib/mysql          ext4          defaults    0 0
```

Tallennetaan muutokset ja poistutaan nanoista. Lopuksi kopioidaan varmuuskopioitu tietokanta takaisin ja käynnistetään MySQL uudelleen.

```
cp -arv /var/lib/oldmysql /var/lib/mysql  
service mysql start
```

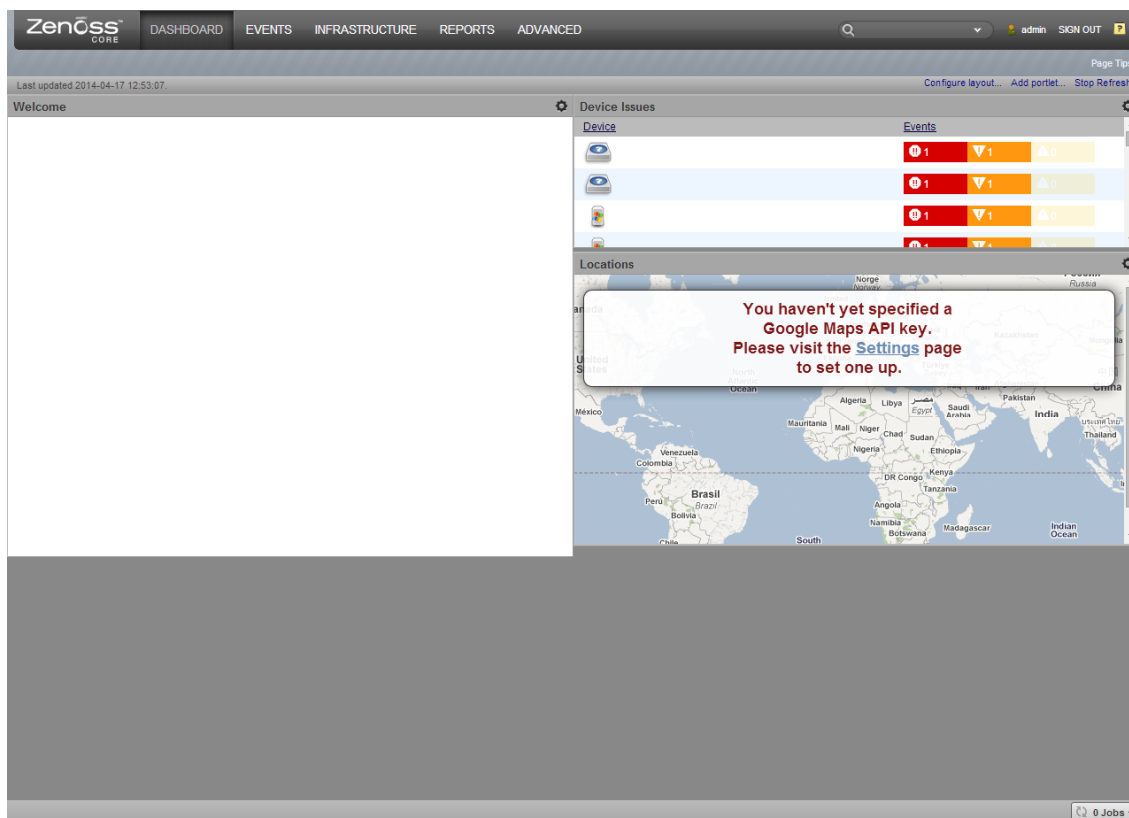
Jos kaikki vaiheet oli tehty oikein, MySQL käynnistyy ilman virheilmoituksia.

3.3.3 Zenoss Coren konfigurointi

Tarvittavien esikonfigurointien jälkeen pääsee lopulta yksilöimään Zenoss Coren asennusta toimeksiantajan tarpeisiin. Navigoidaan selaimella Zenoss-

palvelimen IP-osoitteeseen. Aikaisemmin ei vielä käyty läpi "Get Started"-velhoa, joten tehdään se nyt.

Ensimmäisessä vaiheessa pyydetään antamaan admin-käyttäjälle salasana sekä lisäämään henkilökohtainen käyttäjä. Haluttujen asetusten jälkeen siirrytään seuraavaan vaiheeseen, jossa on mahdollista lisätä haluttuja laitteita. Tässä tapauksessa kuitenkin laitteet on hyvä lisätä myöhemmin, koska vanhassa järjestelmässä on ZenPackeja, jotka vaikuttavat laitteista saatuun dataan. Tämän vuoksi tarvittavien ZenPackien on hyvä olla asennettuna etukäteen ennen laitteiden lisäämistä. Ohitetaan vaihe ja siirrytään itse Zenosiin.

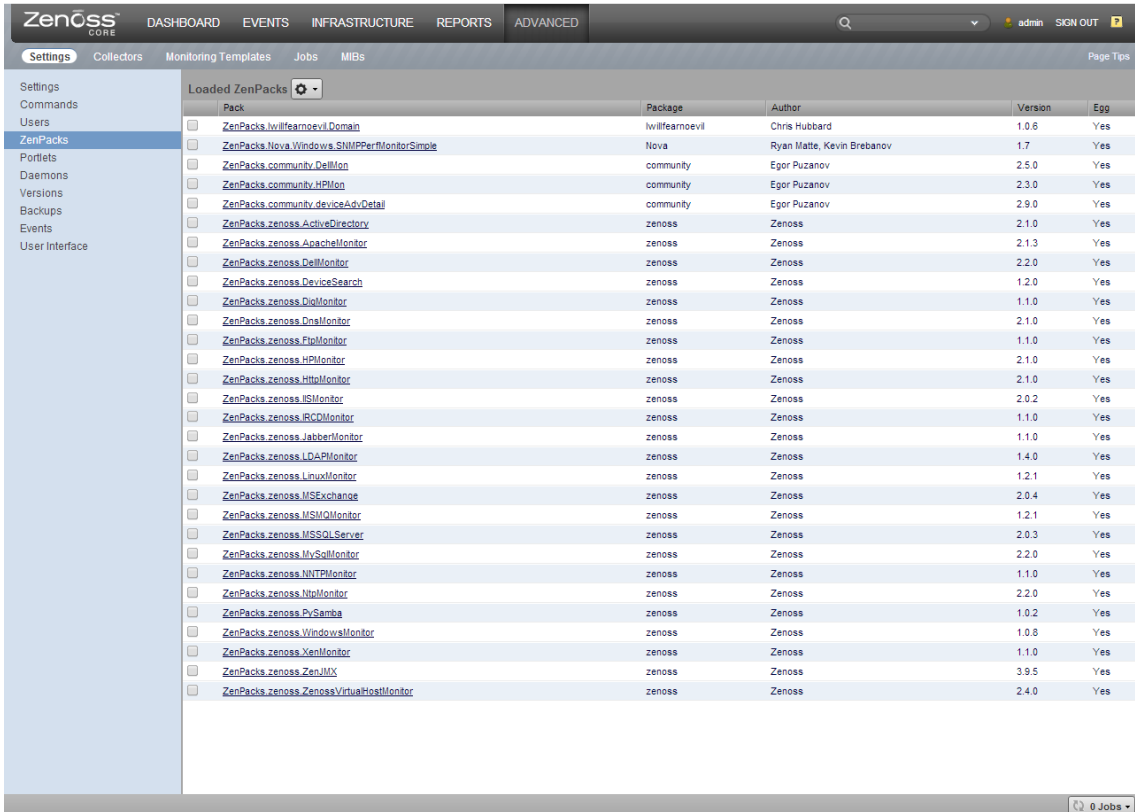


Kuva 1. Zenoss Coren kojelaudan oletusnäkyvä.

Ensimmäisenä nähdään järjestelmän Dashboard, eli kojelautta. Kuvan 1 mukaisesta näkymästä on mahdollista jokaisen käyttäjän muokata mieleisekseen erilaisilla objekteilla, kuten esimerkiksi laitegraafeilla. Aloitetaan järjestelmän muokkaaminen ZenPackien asennuksella. Siirrytään kohtaan Advanced ja ZenPacks. Tällä sivulla nähdään kaikki asennetut ZenPackit (ks. kuvaa 2). Sivulta voi havaita, että ominaisuuksia on jo valmiina laajalti. Oletuksena Zenoss Coren

asennuksessa tulee mukana kaikki ilmaiset Zenoss Inc. -yhtiön tekemät ZenPakit.

Tässä vaiheessa katsottiin vanhaa järjestelmää ja siihen asennettuja ZenPackeja ja tehtiin vertailu siitä, mitä paketteja puuttuu uudesta. Tämän jälkeen tarkastettiin paketti kerrallaan, onko näistä kyseisistä ZenPackeista olemassa uudempi versio, joka tukisi Zenoss 4-versiota. Mahdollisten ongelmien välttämiseksi on suositeltavaa asentaa vain paketteja, jotka ovat tuettu käytetyssä järjestelmäversiossa. Tämän vuoksi joitakin paketteja ei voitu asentaa, joten ne tiputettiin lopullisesta järjestelmästä. Kuvassa 2 nähdään listaus järjestelmän ZenPackeista.



Pack	Package	Author	Version	Egg
<input type="checkbox"/> ZenPacks.Ivillfearnoevil.Domain	Ivillfearnoevil	Chris Hubbard	1.0.6	Yes
<input type="checkbox"/> ZenPacks.Nova.Windows.SNMPPerfMonitorSimple	Nova	Ryan Matte, Kevin Brebanov	1.7	Yes
<input type="checkbox"/> ZenPacks.community.DellMon	community	Egor Puzanov	2.5.0	Yes
<input type="checkbox"/> ZenPacks.community.HPIMon	community	Egor Puzanov	2.3.0	Yes
<input type="checkbox"/> ZenPacks.community.deviceAdvDetail	community	Egor Puzanov	2.9.0	Yes
<input type="checkbox"/> ZenPacks.zenoss.ActiveDirectory	zenoss	Zenoss	2.1.0	Yes
<input type="checkbox"/> ZenPacks.zenoss.ApacheMonitor	zenoss	Zenoss	2.1.3	Yes
<input type="checkbox"/> ZenPacks.zenoss.DellMonitor	zenoss	Zenoss	2.2.0	Yes
<input type="checkbox"/> ZenPacks.zenoss.DeviceSearch	zenoss	Zenoss	1.2.0	Yes
<input type="checkbox"/> ZenPacks.zenoss.DigMonitor	zenoss	Zenoss	1.1.0	Yes
<input type="checkbox"/> ZenPacks.zenoss.DnsMonitor	zenoss	Zenoss	2.1.0	Yes
<input type="checkbox"/> ZenPacks.zenoss.FixMonitor	zenoss	Zenoss	1.1.0	Yes
<input type="checkbox"/> ZenPacks.zenoss.HPIMonitor	zenoss	Zenoss	2.1.0	Yes
<input type="checkbox"/> ZenPacks.zenoss.HttPMonitor	zenoss	Zenoss	2.1.0	Yes
<input type="checkbox"/> ZenPacks.zenoss.IISMonitor	zenoss	Zenoss	2.0.2	Yes
<input type="checkbox"/> ZenPacks.zenoss.IRCMonitor	zenoss	Zenoss	1.1.0	Yes
<input type="checkbox"/> ZenPacks.zenoss.JabberMonitor	zenoss	Zenoss	1.1.0	Yes
<input type="checkbox"/> ZenPacks.zenoss.LDAPMonitor	zenoss	Zenoss	1.4.0	Yes
<input type="checkbox"/> ZenPacks.zenoss.LinuxMonitor	zenoss	Zenoss	1.2.1	Yes
<input type="checkbox"/> ZenPacks.zenoss.MSEXchange	zenoss	Zenoss	2.0.4	Yes
<input type="checkbox"/> ZenPacks.zenoss.MSNOIMonitor	zenoss	Zenoss	1.2.1	Yes
<input type="checkbox"/> ZenPacks.zenoss.MSSQLServer	zenoss	Zenoss	2.0.3	Yes
<input type="checkbox"/> ZenPacks.zenoss.MySQLMonitor	zenoss	Zenoss	2.2.0	Yes
<input type="checkbox"/> ZenPacks.zenoss.NNTPMonitor	zenoss	Zenoss	1.1.0	Yes
<input type="checkbox"/> ZenPacks.zenoss.NtpMonitor	zenoss	Zenoss	2.2.0	Yes
<input type="checkbox"/> ZenPacks.zenoss.PySamba	zenoss	Zenoss	1.0.2	Yes
<input type="checkbox"/> ZenPacks.zenoss.WindowsMonitor	zenoss	Zenoss	1.0.8	Yes
<input type="checkbox"/> ZenPacks.zenoss.XenMonitor	zenoss	Zenoss	1.1.0	Yes
<input type="checkbox"/> ZenPacks.zenoss.ZenJMX	zenoss	Zenoss	3.9.5	Yes
<input type="checkbox"/> ZenPacks.zenoss.ZenossVirtualHostMonitor	zenoss	Zenoss	2.4.0	Yes

Kuva 2. Asennettujen ZenPackien lista.

Kun valmis lista asennettavista ZenPackeista oli saatu tehtyä, ladataan ja asennetaan ne seuraavaksi. Kaikki ZenPakit ja niiden latauslinkit löytyvät kätevästi Zenoss Wikin kautta. Ladataan tarvittavat paketit paikalliselle tietokoneelle. Jos paketit ovat pakatussa tiedostossa, puretaan ne. Kaikkien pakettien tiedostomuoto tulee olla ".egg".

ZenPackien asennus on helppoa. Painetaan yläreunassa olevaa painiketta, jossa on mutteri, ja valitaan "Install ZenPack". Tämä aukaisee valintaikkunan, johon selataan resurssienhallinnan kautta haluttu paketin sijainti. Valitaan paketti ja painetaan valintaikkunasta "OK". Tämän jälkeen järjestelmä asentaa paketin ja ilmoittaa yläreunassa onnistuneesta tai epäonnistuneesta asennuksesta. Asennuksen tulisi onnistua ongelmitta. On tärkeää ottaa kuitenkin huomioon, että jotkin ZenPackit vaativat jonkin toisen ZenPackin esivaatimuksena, tai muita toimenpiteitä, ennen kuin paketti voidaan asentaa. Ilman esivaatimuksia paketin asennus todennäköisimmin epäonnistuu. Mahdollisista vaatimuksista kerrotaan erikseen Zenoss Wikissä paketin lataussivulla.

Asennetaan kaikki paketit yksi kerrallaan edellä mainitulla tavalla. Useammat paketit vaativat Zenossin uudelleenkäynnistyksen, jotta ne tulevat voimaan. Annetaan palvelimella siis komento "service zenoss restart".

Seuraavaksi voidaan lisätä järjestelmään tulevat laitteet. Vanhassa järjestelmässä on paljon laitteita, joten niiden käsin ylöskirjoittaminen ei ole järkevää. Tämän vuoksi käytetään hyväksi Zenossin zendmd-työkalua, jonka avulla Zenoss Coreen voi antaa komentoja ja pyyntöjä Python-kielellä. Vaihdetaan aktiiviseksi käyttäjäksi vanhalla palvelimella zenoss-käyttäjä komennolla "su -l zenoss". Käynnistetään zendmd komennolla "zendmd". Seuraavaksi syötetään komentoriville seuraavat rivit (cluther 2009):

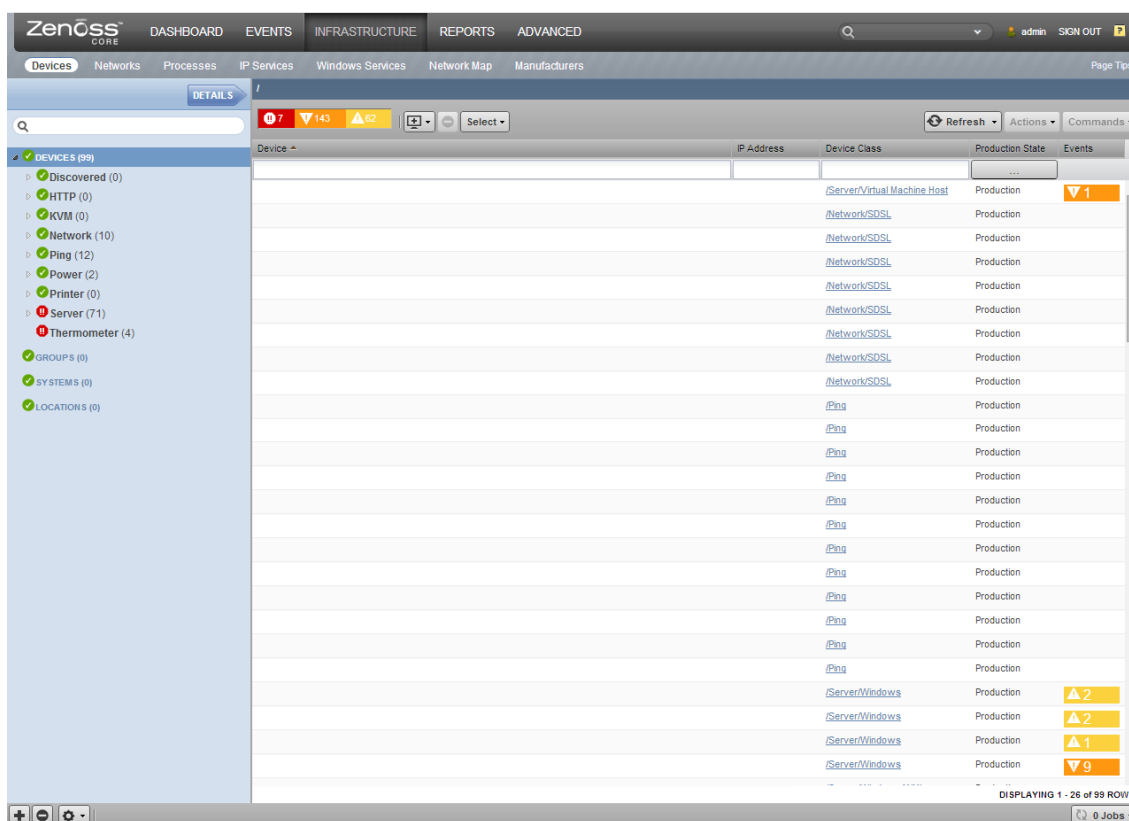
```
out = open("devices.txt", "w")
for d in dmd.Devices.getSubDevices():
    out.write("%s:%s:%s\n" % (d.id, d.managelp,
d.getLocationName() or
"Unknown"))

out.close()
```

Tämän jälkeen zendmd hakee kaikki järjestelmässä olevat laitteet, niiden IP-osoitteet ja määritellyn sijainnin (tai vaihtoehtoisesti antaa sijainnin nimeksi Unknown, jos sijaintia ei ole määritetty) ja tulostaa tulokset devices.txt-

tiedostoon käytössä olevaan työhakemistoon. Tämä kätevä koodi on cluther-nimimerkkiä käyttävän keskustelupalstan käyttäjän neuvoma ohje alkuperäiselle kysyjälle (cluther 2009).

Tämän jälkeen devices.txt on helppo käydä noutamassa Windows-koneelle WinSCP:n avulla. Listan siistimisen jälkeen listan laitteet käytiin läpi toimeksiantajan kanssa ja siihen tehtiin muutoksia. Joitakin listan laitteista otettiin pois, toisten tietoja korjattiin ja lopuksi listaan lisättiin kokonaan uusia laitteita. Valmiin laitelistan avulla lisätään seuraavaksi laitteet uuteen järjestelmään. Siirrytään "Infrastructure"-välilehteen. Kuvassa 3 näkyy avautuva laitelistan näkymä. Kuvasta on peitetty tietoturvan vuoksi laitteiden nimet ja IP-osoitteet.



Kuva 3. Zenossin laitelista.

Laitteiden lisäys onnistuu yksitellen tai useampi laite kerrallaan. Tätä järjestelmää varten tietysti parhaita lisätä useampi laite kerrallaan. Laitteiden lisäys on tehty helpoksi. Klikataan laitelistan yläpuolelta painiketta, jossa on kuva monitorista, jonka sisällä on plus-merkki. Valitaan sitten avautuvasta valikosta "Add multiple devices". Tämän jälkeen avautuu ikkuna, jossa on mahdollista syöttää yksi tai useampi laite kerralla.

Add Devices

Manually find devices Autodiscover devices

Hostnames/IP Addresses

Enter a hostname or IP address for each device you want to add.

+

Details

Device Type: Linux Server (SNMP)

If your device type is not listed, use the default selection. You can add devices of different types from the Zenoss dashboard.

SNMP Credentials

Zenoss will try each of these community strings in turn when connecting to the device.

Community Strings:

Kuva 4. Useamman laitteen lisääminen.

Hostname-kohtaan annetaan laitteen IP-osoite ja Details-laatikossa valitaan sopivat tiedot laitteelle. Oletuksena tässä ei ole kuin muutama vaihtoehto laitteen tyyppiä, mutta laitteen tyyppiä voi vaihtaa myöhemmin oikeaksi, joten sen voi jättää tarvittaessa oletusasetuksiinsa. Yksittäin lisättäessä laitetta vaihtoehtoja laitteen ominaisuuksia varten on paljon enemmän. Kun halutut tiedot on annettu, painetaan plus-painiketta ja saadaan uusi rivi uutta laitetta varten. Kun haluttu määrä laitteita on asetettu, painetaan vain "Save"-painiketta ja järjestelmä lisää laitteet laitelistaan.

Tässä työssä lisättiin laitteet muutama laite kerrallaan ryhmittäin, esimerkiksi ensimmäisenä järjestelmään lisättiin laitelistalta palvelimet. Jos lisätyllä laitteella on domain nimi DNS-palvelimella, Zenoss automaattisesti ottaa tämän nimen käyttöön laitelistassa pelkän IP-osoitteen sijaan.

Palvelimelle on nyt saatu perusasiat kuntoon. Lopuksi luotiin vielä katsaus Rautiaisen opinnäytetyöhön ja tarkasteltiin järjestelmään tehtyjä yksilöintejä, jotka olisi kannattavaa lisätä myös uuteen järjestelmään. Rautiainen oli mm. tehnyt laitelistauksen laiteluokkiin paljon muokkauksia, kuten jaotellut laiteluokkia SNMP-version mukaan, joka mielestäni sai laiteluokkanäkymän näyttämään hiukan sekaiselta, joten kyseiset kohdat jätettiin oletuksiinsa yleisen selkeyden lisäämiseksi.

Lisätään kuitenkin SDSL-laitteille sekä lämpömittareille omat luokkansa, kuten Rautiainen oli tehnyt.

Infrastructure-välilehdeltä katsotaan ensin, että ollaan päänäkyvässä ja valinta on kohdassa Devices. Vasemmasta alakulmasta plusmerkkiä painamalla voidaan lisätä uusi luokka. Tälle annetaan nimeksi "Thermometer" ja hyväksytään. SDSL-laitteet halutaan Network-luokan alle, joten valitaan se aktiiviseksi ja lisätään jälleen uusi luokka, SDSL. Nyt Devices-puun alimmaisena on Thermometer, sekä Networkin alta löytyy SDSL. Luodut luokat ovat vielä tyhjiä, mutta niihin tarkoitetut laitteet voidaan siirtää niihin nopeasti. Helpon laitteen siirtäminen käy valitsemalla se listasta ja vetämällä se uuteen luokkaan. Toinen vaihtoehto on vaihtaa luokkaa manuaalisesti laitteen asetussivulta Zenossista.

Lopuksi vielä haluttiin lisätä muutama kuvaaja helpottamaan langattomien verkkojen IP-osoitevarantojen, UPS-laitteiden akkujen varausten sekä lämpömittareiden arvojen seuraamista. Näitä varten oli helppo käydä katsomassa asetukset vanhasta järjestelmästä ja kopioida ne uuteen järjestelmään.

4 Ongelmat

Vaikka projekti oli kokonaisuudessaan lopulta onnistunut, mahtui työskentelyn varrelle myös monia erilaisia ongelmia. Suurin osa ongelmista kuitenkin liittyi järjestelmän asennukseen, yhteensopivuuteen sekä toimivuuteen.

Alkuperäisen suunnitelman mukaisesti järjestelmä oli tarkoitus päivittää suoraan vanhemmasta versiosta uudempaan uudelle palvelinlaitteelle, jos vain suinkin mahdollista. Päivitystoimenpidettä varten oli olemassa dokumentteja Zenossilta ja tämä antoi uskoa helposta päivitysprosessista. Tämä oli kuitenkin suunnitelluvaiheessa virhearviointi ja asia olisi tarvinnut tarkempaa jatkotutkimusta esimerkiksi sen osalta, oliko muilla järjestelmää käyttävillä päivitys onnistunut halutulla tavalla. Tämän mahdollisuuden kokeilemisen pois jättäminen olisi säästänyt projektista paljon aikaa.

4.1 Käyttöalusta- ja ZenPack-ongelmat

Järjestelmä oli tarkoitus pitää myös Debian-linuxia käyttävänä, kuten aikaisempi järjestelmä oli. Debian ja Ubuntu eivät olleet kuitenkaan enää Zenossin itsensä tukemia alustoja, vaan näille Linux-jakeluille tuki tulee käyttäjäyhteisön kautta. Käyttäjäyhteisöltä oli tarjolla järjestelmän asennusskripti Debian-linuxeille.

Yllättäen automaattinen asennus ei toiminut vaan päättyi virheilmoitukseen, siitäkin huolimatta, että ohjeita oli siihen asti noudatettu huolellisesti. Ongelma automatisoiduissa asennusskripteissä Linux-alustalla voi olla se, että jos järjestelmä ei ole täsmälleen samanlainen kuin millä tai mille asennus on tarkoitettu, sen käytön onnistumisesta ei ole takeita. Epäonnistuminen voi johtua jonkin sovelluksen puutteesta tai järjestelmän määrittelyn erilaisuudesta.

Päädyttiin seuraavaksi manuaaliseen vaihtoehtoon, jota kehiteltiin itse asennusskriptin komentojen pohjalta. Järjestelmää asennettaessa tuli vastaan ongelmia useaan otteeseen, joita joutui ratkomaan ennen kuin pystyi jatkamaan eteenpäin. Tämä jälleen kulutti aikaa hukkaan. Asennus saatiin kuitenkin suoritettua loppuun, mutta lopputuloksena oli järjestelmä, joka ei toiminut.

Ongelmista oppineena seuraavaksi aloitettiin järjestelmän asentaminen alusta käyttöjärjestelmää myöten. Tällä kerralla asennus saatiinkin suoritettua vähemmällä ongelmalla ja Zenoss Core käynnistyi. Myöhemmin kuitenkin ZenPackien asennuksen aikana huomattiin, miksi yhteensopimattomia ZenPackeja ei kannata asentaa. Yksi ZenPack, joka lopullisesta työstä jätettiin pois, aiheutti vain virheilmoituksia ja vaikka se poistettiin, virheilmoitukset eivät lakanneet. Tarkemmin tutkiessa virheilmoitukset johtuivat lähdekoodissa olevista kutsuista moduuleihin, joita ei enää ole käytössä. Poiston jälkeiset virheilmoitukset johtuivat siitä, ettei ZenPack ollut poistanut kaikkia symbolisia linkkejä itsestään.

Tässä vaiheessa oli hyvä pitää toimeksiantajan kanssa palaveria esille tulleista ongelmista. Keskusteltaessa todettiin, että vaikka ongelmat olisi saatu ratkais-

tua nyt, ei se tarkoittaisi sitä, ettei niitä ilmenisi jatkossa. Samalla ilmeni, että alkuun projektille tarkoitettu palvelin (HP ProLiant) ei välttämättä ole se paras mahdollinen, vaan tehokkaampikin laite olisi tarjolla ja se olisi käyttötarkoitukseen sopivampi ollen myös uudempi. Keskustelussa otettiin allekirjoittaneen toimesta myös esille seikka, ettei Debian ole Zenossin itsensä tukema, mutta sen sijaan CentOS-linux on ja se saattaisi olla parempi ja ongelmattomampi vaihtoehto projektille. Päädettiin siis vaihtamaan palvelin, sekä siihen asennettava käyttöjärjestelmä CentOS:ksi.

Heti alusta alkaen CentOS ja Zenoss tuntuivat istuvan paremmin yhteen. Eri-laista jatkuvaa konfigurointia oli paljon vähemmän. Vaikka CentOS:lle tarkoitettu automaattinen Zenossin asennuskripti ei toiminutkaan, manuaalinen asennus onnistui ongelmitta.

4.2 SNMP- ja WMI-ongelmat

Zenoss on erittäin tarkka järjestelmään lisätyistä laitteista ja niiden asetuksista järjestelmässä. Tämä on tietysti myös hyvä asia, sillä kun halutaan seurata laitteiden toimivuutta, ei ole syytä löysäillä.

Ongelmana esiintyi uuden järjestelmän käytössä paljon SNMP- ja WMI-virheitä. Ongelmat eivät yleisesti suoraan johtuneet Zenossista itsestään. Korjaamalla laitekohtaisista asetuksista SNMP- tai WMI-määrittelyt (salasanat ja ryhmänimet) oikeiksi ongelmia saatiin vähennettyä merkittävästi. Lopuista virheilmoituksista antavista laitteista saattoi puuttua tuki SNMP:lle kokonaan, joten kyseisten laitteiden kohdalla SNMP-kyselyt otettiin pois käytöstä.

Laitteita lisättäessä järjestelmään onkin hyvä jo varmistua siitä, että jos mahdollista, laitteessa on jo käytössä ja konfiguroituna SNMP ja/tai WMI. Mikäli laitteessa ei ole mahdollisuutta kumpaankaan, virheilmoitusten välttämiseksi ominaisuudet kannattaa kytkeä pois käytöstä Zenossissa, kuten edellä mainittiin.

4.3 Bad Counter -virheet

Yksi järjestelmän tapahtumalistan kummallisuuksista oli "Bad Counter"-virheilmoitus. Virhe koskee ainoastaan Windows-palvelimia.

Kyse on virheestä, jossa järjestelmä odottaa WMI-palvelun kautta ensisijaisesti tietynlaista informaatiota tietyiltä ohjelmakomponenteilta, mutta kaikkea haluttua informaatiota ei saadakaan. Tämä tarkoittaa sitä, että nämä palvelut tai komponentit puuttuvat palvelimilta, yleisesti siitä syystä, että niitä ei todennäköisesti palvelimen käytössä tarvita. Virhe voidaan nähdä siis turhana ilmoituksena ja koska se oli yleinen, on se parasta piilottaa tärkeämpien virheiden tieltä.

Zenossissa on myös mahdollista muokata ilmoituksia, joten halutunlaisten ilmoitusten huomiotta jättäminen on mahdollista. Avataan järjestelmän Events-välilehti ja valitaan tapahtumaluokat, eli Event Classes -sivu. Virhe oli tyyppiä Status/WMI, joten SubClasses-kohdan alapuolelta navigoidaan kyseiseen luokkaan avaamalla ensin Status-luokka, jonka alta avataan WMI-luokka.

Name	SubClasses	Instances	Events
App	18	70	9
Archive	0	2	0
Capacity	0	0	0
change	7	0	0
clac	1	2	0
cluster	1	5	0
cmd	3	0	0
Config	0	2	0
Conn	1	0	0
DB	1	5	0
Environ	0	3	0
Heartbeat	0	3	0
HW	19	227	1
Ignore	1	58	0
License	0	1	0
Net	10	30	0
OS	2	4	0
Part	7	0	9
Security	9	24	22
Status	24	2	19
Storage	2	8	0
Unknown	0	0	152
Users	0	0	0
Win	14	38	0

Kuva 5. Tapahtumaluokat-sivu.

Tämän jälkeen muokataan tapahtumaluokkaa valitsemalla alakulmasta mutterinkuvasta painamalla "Transform", eli muunna. Seuraavaksi aukeaa sivu, jossa on tekstilaatikko, johon on nyt mahdollista kirjoittaa Python-koodilla haluttuja komentoja. Lisätään seuraava rivi:

```
if evt.summary.startswith("Bad Counter"): evt._action = "drop"
```

Tallennetaan muutokset painamalla "Save". Tämän jälkeen kaikki Status/WMI-luokan alla tulevat tapahtumat, jotka alkavat sanoilla "Bad Counter", hylätään järjestelmästä automaattisesti eikä niistä enää jatkossa ilmoiteta.

4.4 Zenwinperf-palvelun ongelma

Järjestelmässä havaittiin myös zenwinperf-palveluun liittyvä ongelma. Tämä voi johtua jo siitä, että järjestelmässä on suuri määrä Windows-palvelimia. Ongelmana oli, että kyseinen palvelu ei pysynyt päällä.

Ongelman voi korjata tällä hetkellä asettamalla palvelun watchdog-ominaisuus päälle. Avataan zenwinperf-palvelun konfigurointitiedosto Zenoss-palvelimelta:

```
nano /opt/zenoss/etc/zenwinperf.conf
```

Etsitään rivi "#watchdog False", poistetaan kommentointi ja muutetaan arvoksi True, eli "watchdog True" ja tallennetaan. Tämän jälkeen tulee käynnistää palvelu palvelimella uudestaan komennolla:

```
zenwinperf restart
```

Watchdog eli "vahtikoira" nimensä mukaisesti tarkkailee palvelua, ja jos se satuu pysähtymään, se käynnistetään automaattisesti uudelleen.

4.5 VMware ESX -palvelimien seurannan ongelma

Zenoss-palvelimelle oli myös tarkoitus lisätä ESX-palvelimien virtuaalipalvelimien seuranta. Tästä kuitenkin muodostui ongelma, sillä edellisessä järjestelmässä käytetty yhteisön tuottama ZenPack ei ollut tuettu uudessa versiossa.

Zenoss tarjoaa vakiona tätä käyttöä varten omaa ZenPackia, jonka toiminta on kuitenkin erilainen. Moduulin toiminta on myös saanut paljon moitteita käyttäjäyhteisöltä. Ero on huomattava, sillä jo ensimmäisen ESX-palvelimen virtuaalipalvelimien seuranta nosti Zenoss-palvelimen prosessorin sekä muistin käyttöasteen maksimiin. Tämä sai järjestelmän lähes käyttökelvottomaksi. Palvelimen toiminta hidastui merkittävästi.

Tämän jälkeen tehtiinkin päätös, että virtuaalipalvelimien seuraaminen unohdetaan uudesta Zenossista, jättämällä kuitenkin ESX-palvelimet itsessään järjestelmään niiden toiminnan seuraamiseksi.

5 Tulokset

Työn ensisijaisena tarkoituksena oli päivittää toimeksiantajan Zenoss Core -valvontajärjestelmä uusimpaan versioon säilyttäen vanhan järjestelmän toiminnallisuudesta niin paljon kuin mahdollista. Kysymyksenä oli, kuinka ja miten järjestelmä päivitetään. Projektin aikana tutkittiin ja kokeiltiin vaihtoehtoja järjestelmän päivittämistä varten ja lopulta päädyttiin kokonaan uuden järjestelmän pystyttämiseen, vanhan järjestelmän hyödyntämisen sijasta. Kaikkea järjestelmän aikaisemmasta toiminnallisuudesta ei saatu uuteen järjestelmään mm. yhteensopivuusongelmien takia, mutta tärkeimmät ominaisuudet saatiin kuitenkin mukaan.

Lisäksi työssä oli tarkoituksena uutena ominaisuutena sijoittaa järjestelmä omalle erilliselle palvelimelleen. Vanhan järjestelmän toiminta virtuaalipalvelimena ei ollut toimeksiantajan mielestä kovinkaan sujuvasti toimiva ratkaisu. Tämän

vuoksi palvelin toimi toivottua huonommin. Uuden järjestelmän sijaitessa omalla palvelimellaan, saatiin järjestelmään haluttua nopeutta ja sujuvuutta.

6 Pohdinta

Projekti oli kokonaisuudessaan onnistunut. Ennen työn aloittamista toimeksiantaja loi projektille minimitavoitteet, joihin vähintään pyritään, sekä lisätavoitteet, joita katsottaisiin siinä tapauksessa, että projekti onnistuisi odotettua nopeammin ja ongelmitta.

Lopputuloksena oli toimeksiantajan mielestä jopa minimitavoitteita parempi, työn aikana ilmenneistä ongelmista ja niiden viemästä ajasta huolimatta. Ongelmien yli päästiin tavalla tai toisella ja tuloksena oli toimiva ja käyttökelpoinen järjestelmä erityisesti toimeksiantajan palvelimien toiminnan seuraamista varten.

Projektin tulokset antavat viitteitä ja pohjaa toimeksiantajalle Zenoss-järjestelmän jatkokehityksestä sekä ylläpidosta. Tämän projektin sekä aikaisemman järjestelmän pohjalta toimeksiantajalla on varmasti näkemys siitä, kuinka helppoa tai vaikeaa järjestelmää on ylläpitää ja päivittää tulevaisuudessa.

Vaikka projekti sujui loppujen lopuksi suunnitelmien mukaan, olisi suunnittelua varmasti voinut tehdä tarkemmin. Tähän ajatukseen päädyttiin sekä toimeksiantajan, että allekirjoittaneen osalta. Projektin molemmilla osapuolilla olisi mahdollisesti voinut olla jo projektin alkaessa selkeämmät päämäärät ja tiedot projektin suorittamisesta, materiaaleista, ohjelmistoista sekä laitteistosta.

Lähteet

- cluther. 2009. Re: Export Device List.
<http://permalink.gmane.org/gmane.comp.monitoring.zenoss.user/27835>. 17.4.2014.
- Rautiainen, I. 2009. Järjestelmien valvonnan suunnittelu ja toteutus. Jyväskylä: Jyväskylän ammattikorkeakoulu. Tietoverkkotekniikan koulutusohjelma.
- Rolek, A. 2013. Formatting, Partitioning, and Mounting Drives on CentOS.
<http://docs.tinyfactory.co/centos/2013/01/11/formatting-partitioning-and-mounting-drives-on-cent-os.html>. 17.4.2014.
- Sourceforge. Release Notes for Zenoss Core Version 4.2.4.
http://sourceforge.net/projects/zenoss/files/Documentation/zenoss-4.2.x-docs/zendocs-4.2.4/Zenoss_Core_Release_Notes_00-062013-4.2-v08.pdf. 4.11.2014.
- Weisel, M. 2014. Zenoss Core 4 Installation.
<http://binarynature.blogspot.fi/2012/11/zenoss-core-4-installation.html>. 17.4.2014.
- Wikipedia. 2014. Zenoss. <http://en.wikipedia.org/wiki/Zenoss>. 5.11.2014.
- Zenoss Inc. 2014. Core vs Commercial. <http://www.zenoss.com/solution/core-vs-commercial>. 17.4.2014
- Zenoss Wiki. 2013. History of Zenoss Core.
http://wiki.zenoss.org/History_of_Zenoss_Core. 17.4.2014.
- Zenoss Wiki. 2014. All ZenPacks. http://wiki.zenoss.org/All_ZenPacks. 17.4.2014.
- Zenoss Wiki. 2014. Upgrading Zenoss.
http://wiki.zenoss.org/Upgrading_Zenoss. 4.11.2014