SETTING UP A SMALL OFFICE NETWORK

Tomi-Jaakko Hongell

Bachelor's Thesis
May 2013

Degree Programme in Software Engineering
The School of Technology

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖN
KUVAILULEHTI

Tekijä(t) Hongell Tomi-Jaakko	Julkaisun laji Opinnäytetyö	Päivämäärä 13.05.2013
	Sivumäärä 32	Julkaisun kieli Englanti
	Luottamuksellisuus ( ) saakka	Verkkojulkaisulupa myönnetty (X)

Työn nimi
SETTING UP A SMALL OFFICE NETWORK

Koulutusohjelma
Ohjelmistotekniikka

Työn ohjaaja(t)
PELTOMÄKI, Juha

Toimeksiantaja(t)
Iwa Labs Oy

Tiivistelmä

Opinnäytetyön tarkoituksena oli suunnitella ja toteuttaa Iwa Labs Oy:n tytäryhtiölle Iwa Labs Thailand:lle uuteen toimistoon tietoturvallinen ja vakaa lähiverkko. Tietoturvallisuutta oli tarkoitus lisätä useilla erilaisilla ohjelmilla(anti-virus ja monitorointi -ohjelmat). Lähiverkon vakautta suunniteltiin lisättävän kaistanrajoituksilla, kahden internet-yhteyden yhtäaikaisella käytölla sekä tiettyjen yhteyksien erityisohjaamisella.

Opinnäytetyöstä käy selville miksi valittiin käyttöjärjestelmäksi Ubuntu Server 12.04 Lts ja kuinka asennetaan serverille tietoturvallisuutta parantavia ohjelmia. Opinnäytetyösta käy myös selville kuinka saadaan ohjattua verkon liikennettä sekä kuinka asetetaan kaistanrajoitukset.

Työskentelyn päätteeksi saatiin toimiva lähiverkko päivitettyjen tietoturva-asetusten kanssa. Opinnäytetyö tulee jatkossa toimimaan Iwa Labs Thailand:n muistilistana uutta lähiverkkoa pystyttäessä. Työn lopusta löytyy työn kirjoittajan mielipide työn onnistumisesta.

Avainsanat (asiasanat)
Tietokoneverkko, serveri, palomuuri, tietoturva, kaistanleveys.

Muut tiedot

JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES

DESCRIPTION

Author(s) Hongell, Tomi-Jaakko	Type of publication Bachelor´s Thesis	Date 13.05.2013
	Pages 32	Language English
	Confidential ( ) Until	Permission for web publication (X)

Title
SETTING UP A SMALL OFFICE NETWORK

Degree Programme
Software Engineering

Tutor(s)
PELTOMÄKI, Juha

Assigned by
Iwa Labs Oy

Abstract

The purpose of this bachelor’s thesis was to plan and implement safe and stable network systems to the Iwa Labs Oy's joint venture Iwa Labs Thailand's office. To make the network safer the plan was to install and configure anti-virus programs and monitoring tools. Limiting bandwidth using two different internet connections at the same time and forwarding certain connections were to make the network more stable.

The thesis discusses why Ubuntu Server 12.04 Lts was selected to be the operating system and how the installation of security and monitoring programs were done. The thesis also guides how to make limitations and how to control the traffic over the network.

In the future the thesis will be a guide for the Iwa Labs Thailand on implementing a new network and how to set basic security settings to the servers. The last pages of the thesis present the author's personal opinion of this thesis project.

Keywords
Network, server, firewall, security, bandwidth.

Miscellaneous

CONTENTS

TERMINOLOGY AND ABBREVIATIONS

Network – A system of computers interconnected by telephone wires or other means in order to share information.

> ”A system of computers and peripherals, such as printers, that are linked together. A network can consist of as few as two computers connected with cables or millions of computers that are spread over a large geographical area and are connected by telephone lines, fiberoptic cables, or radio waves. The Internet is an example of very large network.”
> (Network, The Free Dictionary, 2013)

Server – Server is computer that manages specific services and network's resources.

Switch – Switch connects computers together using layer two from OSI model.

SSH – Secure Shell, protocol which gives secure connection between client and server.

Zero Day Exploit – Called either Day Zero or Zero-Day, it is an exploit that takes advantage of a security vulnerability on the same day that the vulnerability becomes publicly or generally known.
(Zero Day Exploit, Webopedia, n.d.)

RootKit – Malicious software which is difficult to notice and find because it will turn itself on same time as operating system is starting, before anti-virus programs starts.

1 INTRODUCTION

1.1 Objectives of this thesis

This thesis was assigned by Iwa Labs Oy, which is small software company in Finland to implement network systems in their new joint venture Iwa Labs Thailand's new office. Iwa Labs Thailand is small software company which makes websites and mobile applications. The Office of Iwa Labs Thailand is located in Khon Kaen, a city on the Khorat Plateau, in the central-northwestern area of Isaan.

In this thesis the goal was to build a network system to small office. The network was to have one PC with Ubuntu Server as router and firewall, two different internet connections from different companies, one switch where the main connections to the workstations are and one wireless router giving a connection for mobile phones and visitors. The main objectives are shaping the bandwidth for a fast and reliable internet use and taking care of most of the basic security issues.

1.2 Hardware

The main router was a basic PC, with 4 NICs. Using Ubuntu Server 12.04 as an operator system, the server does not require any special components and it should work with Pentium class computers.

The motherboard is ASRock 880GM-LE FX and it has one gigabit Ethernet card embedded in it and the card has supported drivers for Ubuntu. Other NICs are two TP-LINK TG-3468 and one TP-LINK TL-SF1024 and they also have drivers compatible with Ubuntu.

1.3 Motivation

Iwa Labs Thailand just moved to a new office and they needed a network that is safe and fast. In their last office there was too much traffic in their network and there was no administration there.

A problem with the old network was that everyone used wireless connections with three different DSL routers. The routers were all the time congested with too much data and stopped working.

The Internet connections went down at least few times a week. The employees used many services which needed a great deal of bandwidth (YouTube, Spotify, etc), for which reason normal web surfing and important SSH connections went down many times. The old solution did not have any security improvisation or any backup service for the lost connections.

2 GOALS

2.1 Sharing multiple internet connections

There are two different ISPs and those two were to be merged together. With this kind of setup both connections can be used at same time to make sure that there is a working connection, whenever there is at least either one of the connections working and that the bandwidth can be used from both connections.

2.2 Limiting bandwidth

With limiting bandwidth it can be made sure that everyone has enough bandwidth for important connections, such as using SSH connections and basic web surfing. By limiting upload and download bandwidths can be made sure that there will not be any delays of the total bandwidth. Limiting and sharing bandwidth will be carried out with IP tables and TC/HTB.

2.3 Providing stable SSH connections

The programmers need to use SSH connections for their daily missions and it has to be made sure they works all the time. Enough bandwidth should be given to the SSH connection so it will work all the time regardless of downloading or/and uploading. SSH connection will always need to go through only one ISP at time, this is a better way to make sure that the connection is not disturbed because two different ISPs.

2.4 Security issues

Most of the security measures to do were to configure them inside Ubuntu Server and use tools which are already embedded in Ubuntu.

SSH connections come only from inside and certain MAC addresses from the internet; therefore, it should be secured that no one can sniff the MAC addresses. DOS attacks and brute forcing through SSH connection need to be taken care of.

Wireless connection for visitors should be secured well, the password needs to be secure enough and connection to the internet will be only allowed. Email alert to administrators should be sent if there is any kind of fishy business going on, such as password hacking attempt or IP/MAC sniffing.

2.5 Monitoring

With proper monitoring it can be seen how the server machine is doing, how the network is behaving and if there are any security issues. The monitoring system needs to alert administrators if something odd happens and the system needs to make a weekly report of network usage so that the administrators can improve and secure the network even better.
Issues to be monitored:

•Hardware: Any problems with electronic devices, issues about overheating.

•Bandwidth: Is the bandwidth going smoothly, and is there enough bandwidth to important connections.

•Software: Monitoring that there are only trusted programs going on and none of the programs is doing anything weird.

3 PROBLEMS BEFORE WORKING AND PLANS HOW TO DO THE WORK

3.1 What are the difficulties

3.1.1 Hardware

Before starting embedding the author of the thesis was afraid of the compatibility problems between hardware and operation system. There are some problems with drivers when using Ubuntu, because many companies are usually making drivers for Windows only. Difficulties can appear if the components are not good quality; they can overheat easily or just make the OS crash with some misbehavior.

3.1.2 Software

With software there might be problems when many programs are doing quite the same thing at the same time (for example, shaping bandwidth). Some security software can prevent other software from working normally. Memory consumption can get high if the software is doing the same task at the same time, making them go to an infinite loop.

Testing proper bandwidth limit may be time consuming and it can be hard to test because changes in the bandwidth can be huge depending on other employees and ISPs. Testing security might also be hard to carry out because it will be like playing game against oneself, one already thinks that one knows the weaknesses and strengths of the server and tries to break and hack those. Making sure that basic attacks will not work is possible, however, trying to prevent real smart and planned attacks can be hard, due to possibility and capacity of attack types.

3.2 Plans how to do work and how to solve problems

The main custom to handle problems is to search for an answer on the internet. Basically sites focused on Ubuntu help with all problems with software and most of hardware problems also. Coworkers and teachers from the university were also available to help, not to forget help from fellow students who could also help with problems.

Good practice for noticing problems is to write and read Log files. These files will give more information about where the problems are and what is causing them.

.

4 PROGRAMS AND PROTOCOLS

4.1 DNS

”Domain Name System (or Service or Server), an Internet service that translates domain names into IP addresses”.
(DNS, Webopedia, n.d.)

The internet is based on IP addresses, but it is easier to remember names than random numbers, therefore, DNS is there to translate the names into the IP address for the user.

4.2 PPPoE

“Acronym for Point-to-Point Protocol over Ethernet. PPPoE relies on two widely accepted standards: PPP and Ethernet. PPPoE is a specification for connecting the users on an Ethernet to the Internet through a common broadband medium, such as a single DSL line, wireless device or cable modem. All the users over the Ethernet share a common connection, so the Ethernet principles supporting multiple users in a LAN combine with the principles of PPP, which apply to serial connections.”
(PPPoE, Webopedia, n.d.)

In this thesis PPPoE setting to the server were needed, so that the server can try to make a connection call, not the router as usually. This is described more in detail in chapter 7.

4.3 Iptables

Iptables is program in Linux that guides networks behavior with shaping and filtering rules. Iptables is Linux kernel's firewall's (netfilter) user interface. (iptables(8) - Linux man page, die.net, n.d.) Nowadays iptables is a standard in modern Linux distributions.

Iptables contains tables and chains which will decide the order of the commands and rules. Iptables has four main tables built-in it: filter, NAT, Mangle and Raw tables. The main tables have some built-in chains inside them.

The Filter Table has INPUT, OUTPUT and FORWARD chains built-in and the filter table is a default table and if the user defines their own tables those will use the filter table.
NAT table has three built-ins: PREROUTING, OUTPUT and POSTROUTING chains and the table is heard when a packet with a new connection is encountered.
The raw table is for configuration exemptions and it has PREROUTING and OUTPUT chains by default
For specialized use there is Mangle table with two chains, PREROUTING and OUTPUT. Also, the chains INPUT, FORWARD and POSTROUTING are supported since kernel version 2.4.18.

4.4 Nagios

Nagios is a monitoring tool for networks and it can follow multiple servers at the same time. Nagios has a Web interface and with Nagios users can monitor their server's local resources (e.g. hard drive, processor, memory,) and network's services (HTTP, SMTP, SSH, etc). Nagios can also make reports and alert the administrators about the problems in the server.

4.5 **SELinux – Apparmor**

SELinux is developed by The United States National Agency, and it is a support feature to some Linux distributions. With SELinux users can administrate user rights, policies and security contexts more accurately
AppArmor is a framework that protects applications and the operation system

from external or internal threats. AppArmor defines where programs can have access and using which privileges.
Compared to better known SELinux, AppArmor is easier to use and does almost all the same things as SELinux.

5 HARDWARE

5.1 Requirements for hardware

The server machine for a small office can be almost anything, only issues which most of the time matter are capacity and speed of memory and the speed of CPU. With low bandwidth (<20Mbit/s) internet connections, the Network interface cards do not need to be anything special. Network interface cards should have drivers to the Linux system; otherwise there might be many difficulties. Good server components should be quiet and resistant and not produce heat. Low electric consumption is also a good feature on components.

5.2 Keeping computer power on all the time

Computer should stay switched on all the time and if for some reason it I turns off it should automatically turn on again. BIOS has a setting which will automatically start the computer, and this is tried to be set to On-position.

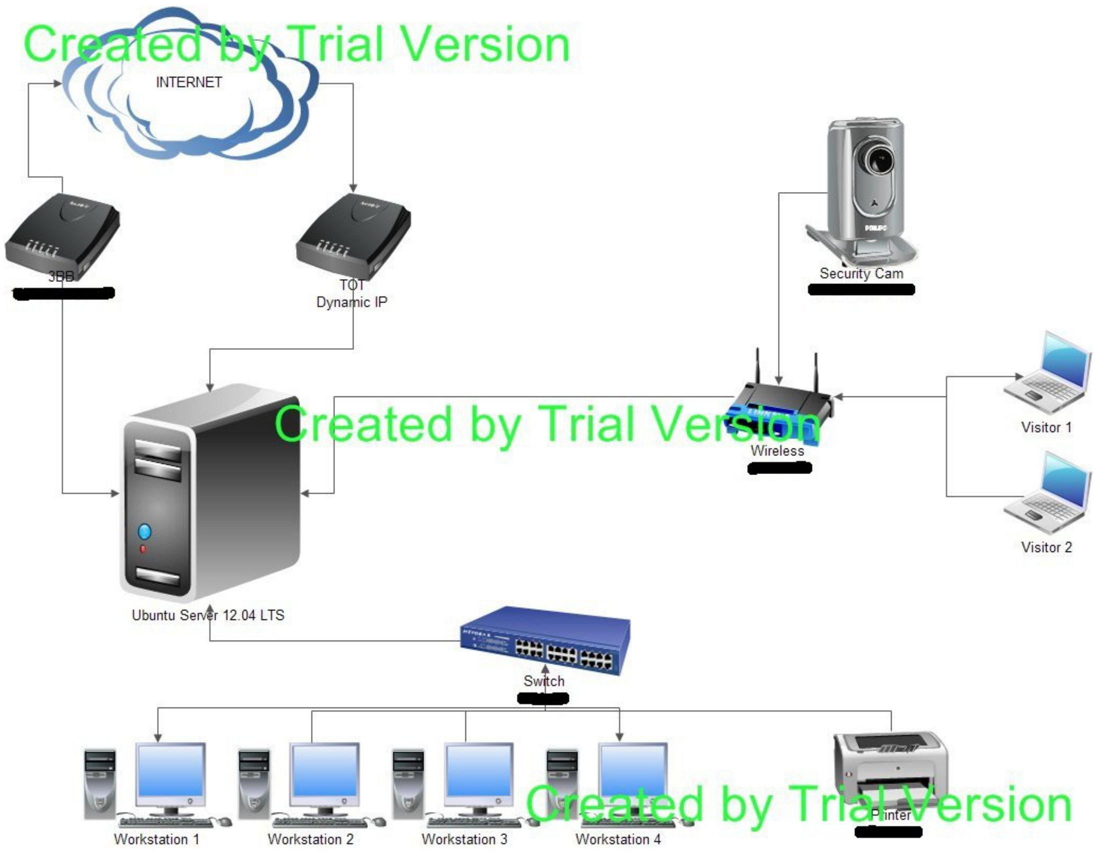


Figure 1: Topology of network

6 UBUNTU SERVER

6.1 What is Ubuntu Server 12.04 LTS

Ubuntu Server 12.04 LTS is an operation system sponsored by Canonical Ltd.. Ubuntu is based on Debian Linux distribution and it is under GNU and GPL licenses. Ubuntu is the most popular Linux distribution on personal computers and it is also often used for servers and cloud services. (Katherine Noyes, Which Linux Distro Is Fairest of Them All? Ubuntu, Survey Says, PCWorld, 2012)

6.2 Why Ubuntu Server 12.04 LTS?

Ubuntu Server 12.04 LTS was chosen because it can do everything that was expected of it now and in future. Ubuntu was also already familiar to use and there is a great deal of information about it on the internet. In the author's opinion the best features in Ubuntu are:

- It is fast
- It has long support (up to year 2016)
- Free to use
- If there are problems there is much information available
- Security is updated

6.3 What makes Ubuntu Server safe?

Because Ubuntu is the most popular Linux distribution it most likely has more security threads than other distributions; however, it also has the largest support and development group behind it. Ubuntu has many different kinds of security programs, such as scanners, firewalls, analyzers and user right regulators.

7 SETTING PPPoE CONNECTIONS

7.1 Configuring PPPoE connections

DSL router's settings were first changed to bridge mode, so the server will be the one making a call to ISP. Because there should be different ISPs the configuration was made manually for both services. For making a connection with PPPoE user name and password provided by the ISP were needed. /etc/ppp/chap-secrets file was edited adding the user names and passwords of both ISPs there. Peer file is the file which contains all the settings needed to connect to the ISP, the files are saved under /etc/ppp/peers. Two files, dsl-provider-3bb and dsl-provider-tot were made, which are named after providers.

These are really basic PPPoE configurations and the difference between these two is that they have different plug-in where to bind, user name, unit number and link name.

```
noipdefault
#defaultroute
#replacedefaultroute
hide-password
lcp-echo-interval 30
lcp-echo-failure 4
noauth
persist
#mtu 1492
maxfail 0
#holdoff 20
plugin rp-pppoe.so eth3
#usepeerdns
user "username@isp"
unit 1
linkname TOT
```

```
noipdefault
#defaultroute
#replacedefaultroute
hide-password
lcp-echo-interval 30
lcp-echo-failure 4
noauth
persist
#mtu 1492
#persist
maxfail 0
#holdoff 20
plugin rp-pppoe.so eth1
#usepeerdns
user "username@isp"
unit 0
linkname 3BB
```

7.2 Managing interfaces

Changes made to / /network/interfaces will control how Ethernet cards will behave. There it was set that eth0 and eth2 are PPPoE connections and eth1 and eth3 will be the LAN-network and wireless connection to the server. The LAN and wireless IP's were given to their users and DHCP servers were made for the Ethernet cards.

```
#3bb pppoe
auto ppp0
iface ppp0 inet ppp
pre-up ip link set eth1 up
provider dsl-provider eth1

#TOT pppoe
auto ppp1
iface ppp1 inet ppp
pre-up ip link set eth3 up
provider dsl-provider-tot eth3
```

7.3 Configuring DHCP servers

DHCP servers were set to wireless pointer and to the switch where the LAN is located.

```
sudo nano /etc/dhcp/dhcpd.conf
```

```
ddns-update-style none;
log-facility local7;
authoritative;

# Unsecured WLAN
subnet 10.0.60.0 netmask 255.255.255.0 {
        interface                       eth2;
        option routers                  10.0.60.1;
        option subnet-mask              255.255.255.0;
        option broadcast-address        10.0.60.255;
        option domain-name-servers      8.8.8.8, 8.8.4.4;
        default-lease-time              7200;
        max-lease-time                  86400;
        range                           10.0.60.2
10.0.60.100;
}

# Secure LAN
subnet 10.0.0.0 netmask 255.255.255.0 {
        interface                       eth0;
        option routers                  10.0.0.1;
        option subnet-mask              255.255.255.0;
        option broadcast-address        10.0.0.255;
        option domain-name-servers      8.8.8.8, 8.8.4.4;
        default-lease-time              7200;
        max-lease-time                  86400;
        range                           10.0.0.2
10.0.0.100;

        host printer {
                hardware ethernet xx:15:99:a4:96:a2;
                fixed-address   10.0.0.100;
        }
}
```

7.4 Sharing connection

A file was made where small script to setting IP addresses and default gateways was written and weights to devices were given. This script will also check how many connections there are, and it sets the right gateway depending on that.

`/etc/network/load_balancing`

```
#!/bin/bash
# Set devices:
DEV1=${1-ppp0}   # default eth0
DEV2=${2-ppp1}   # default ppp0

# Get IP addresses of our devices:
ip1=`ifconfig $DEV1 | grep inet | awk '{ print $2 }' |
awk -F: '{ print $2 }'`
ip2=`ifconfig $DEV2 | grep inet | awk '{ print $2 }' |
awk -F: '{ print $2 }'`

# Get default gateway for our devices:
```

```
gw1=`route -n | grep $DEV1 | grep -v '^0.0.0.0' | awk
'{ print $1 }'`
gw2=`route -n | grep $DEV2 | grep -v '^0.0.0.0' | awk
'{ print $1 }'`

echo "$DEV1: IP=$ip1 GW=$gw1"
echo "$DEV2: IP=$ip2 GW=$gw2"

ip route del default
ip route del default
ip rule del fwmark 1 lookup 3BB

/etc/network/bandwidth-shape.sh

if  [ "$gw1" == "" ] && [ "$gw2" == "" ]; then
        service networking restart
        echo "ei loytynyt mitaan"
elif [ "${gw1}" == "" ]; then
        route add default gw $gw2
        echo "loyty gw2"
elif [ "${gw2}" == "" ]; then
        route add default gw $gw1
        echo "loyty gw1"
else
        ip route add default via $ip1 dev $DEV1 table 3BB
        ip route add default via $ip2 dev $DEV2 table TOT
        ip route add default scope global nexthop via
$gw1 dev $DEV1 weight 1 nexthop via $gw2 dev $DEV2 weight
2
        ip rule add fwmark 1 lookup 3BB
        echo "loytyy"
fi
```

7.5 Keeping connections up

Another script was written which will be executed every minute by crontab. This script will check if the internet connection is still on or not. If the connection is lost it will run the Load balancing script.

```
/etc/network/check_internet_connection
```

```
#!/bin/bash -l

# Check and restart connections
```

```
wget --spider http://google.com
STR="$?"

if  ifconfig ppp0 | grep -q "inet addr:" && ifconfig ppp1 | grep -q "inet addr:" && [ $STR == 0 ]   ; then
    echo "all ok!"
    exit
fi

if test `find "/etc/network/last_connection" -mmin +60` ; then
        if ifconfig ppp0 | grep -q "inet addr:" ; then
           off dsl-provider-tot
           pon dsl-provider-tot
           echo "restarted TOT"
        else
           poff dsl-provider
           pon dsl-provider
           echo "restarted 3BB"
        fi
        sleep 30
        /etc/network/load_balancing
   exit
fi

touch /etc/network/last_connection
```

8 CONFIGURING IPTABLES

8.1 What is done?

Configuring iptables will give the right permissions to connections and also will shape the bandwidth. Most of the basic security settings are made here. Configuring the iptables forwards the SSH connection from the internet and the right MAC addresses to the local network where the SSH port is open. It also forwards upcoming HTTP calls to the sec camera. All the traffic from the wireless to the local network is dropped and all the SSH connection from local network to the internet is directed through only one ISP at a time.

8.2 How does it work?

Adding commands to the iptables is carried out as follows:

```
-nat PREROUTE 10.1 to 10.2 port 22
```

The next snippet will forward the SSH to the local network from the internet if the sender's MAC address is correct.

```
-nat PREROUTE 10.1 to 10.2 port 22 -m mac --mac-source
00:xx:xx:xx:xx:xx -j ACCEPT
```

Using iptables a rule to block the connection is added.

```
-A FORWARD -d 10.0.0.0/24 -i eth2 -j DROP
```

Ready iptables are illustrated below as follows:

```
# Generated by iptables-save v1.4.12 on Mon Mar 25
15:47:01 2013

*nat

:PREROUTING ACCEPT [0:0]

:INPUT ACCEPT [0:0]

:OUTPUT ACCEPT [0:0]

:POSTROUTING ACCEPT [0:0]

-A PREROUTING -d xxx.xxx.xxx.xxx/32 -p tcp -m tcp --dport
8000 -j DNAT --to-destination 10.0.0.99:8000

-A POSTROUTING -o ppp0 -j MASQUERADE

-A POSTROUTING -o ppp1 -j MASQUERADE

-A POSTROUTING -s 10.0.0.99/32 -p tcp -m tcp --sport 8000
```

```
-j SNAT --to-source xxx.xxx.xxx.xxx
COMMIT
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A FORWARD -d 10.0.0.99/32 -p tcp -m state --state
NEW,RELATED,ESTABLISHED -m tcp --dport 8000 -j ACCEPT
-A FORWARD -d 10.0.0.0/24 -i eth2 -j DROP
COMMIT
*mangle
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
-A PREROUTING -i eth0 -p tcp -m tcp --dport 22 -j MARK
--set-xmark 0x1/0xffffffff
-A FORWARD -o ppp1 -p tcp -m tcp --tcp-flags SYN,RST SYN
-m tcpmss --mss 1400:65495 -j TCPMSS --clamp-mss-to-pmtu
COMMIT
```

9 SECURITY SETTINGS

9.1 What was to be done

The installation of security software and configuration was mostly done according to list from the internet (How to secure Ubuntu 12.04 LTS server. The Fan Club. 2012). Visitors from wireless connection cannot get any connection to LAN network, with this improvement attacks from inside the company network can be reduced. SSH connections to the server should only be possible to be made from the inside LAN network and from few different MAC addresses using internet connections. Every logging attempt from non-trusted addresses should be written in logs and administrators should be informed about the situation. This is a way how administrators can improve security systems of their servers quickly when something weird is happening.

9.1.1 Securing shared memory

Adding line: tmpfs /dev/shm tmpfs defaults,noexec,nosuid 0 0
to the file: /etc/fstab
This will make attacking against running service using /dev/shm, harder to do.

9.1.2 Disabling root login

```
sudo nano /etc/ssh/sshd_config
```

Changing ”PermitRootLogin” to ”no”.

With disabled root log in crackers cannot try to get inside the network straight by using the root password, this will make them to use more time to first to get inside of any other user's account and after that they can try to get “sudo” rights.

9.1.3 Sysctl settings

Editing /etc/sysctl.conf setting will prevent source routing of incoming packets and log malformed IP's.
For now ready made settings are used in this project:

```
# IP Spoofing protection
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1

# Ignore ICMP broadcast requests
```

```
net.ipv4.icmp_echo_ignore_broadcasts = 1

# Disable source packet routing
net.ipv4.conf.all.accept_source_route = 0
net.ipv6.conf.all.accept_source_route = 0
net.ipv4.conf.default.accept_source_route = 0
net.ipv6.conf.default.accept_source_route = 0

# Ignore send redirects
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0

# Block SYN attacks
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_max_syn_backlog = 2048
net.ipv4.tcp_synack_retries = 2
net.ipv4.tcp_syn_retries = 5

# Log Martians
net.ipv4.conf.all.log_martians = 1
net.ipv4.icmp_ignore_bogus_error_responses = 1

# Ignore ICMP redirects
net.ipv4.conf.all.accept_redirects = 0
net.ipv6.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv6.conf.default.accept_redirects = 0

# Ignore Directed pings
net.ipv4.icmp_echo_ignore_all = 1
```

Sysctl needs to be reloaded so the changes will come to use:

```
sudo sysctl -p
```

9.1.4 IP Spoofing

Adding these lines:

```
order bind,hosts

nospoof on
```

To the file:

```
sudo vi /etc/host.conf
```

Will prevent IP spoofing.

9.1.5 PHP

No website on the server is used, however, for future use some settings in the file are changed:

```
sudo nano /etc/php5/apache2/php.ini
```

Changing and adding the following lines will make the running of PHP programs much safer.

```


disable_functions = exec,system,shell_exec,passthru

register_globals = Off

expose_php = Off

magic_quotes_gpc = Off
```

9.1.6 Apache

Next thing that was done was was to restrict Apache information leakage with:

```
sudo nano /etc/apache2/conf.d/security
```

ServerTokens Prod

```
ServerSignature Off

TraceEnable Off

Header unset ETag

FileETag None

sudo /etc/init.d/apache2 restart
```

9.1.7 ModSecurity, ModEvasive & Fail2Ban

To protect from DOS attacks apache2 mod_security and mod_evasive mods were installed, which take care of weird behaving loggers. Ban2fail was also installed and configured to ban users who write the password wrong too many times and also send email to administrators.

9.1.8 PSAD

”psad is a collection of three lightweight system daemons (two main daemons and one helper daemon) that run on Linux machines and analyze iptables log messages to detect port scans and other suspicious traffic”
(PSAD. CipherDyne. n.d.)

Installation of psad:

```
mkdir /tmp/.psad

cd /tmp/.psad

wget http://cipherdyne.org/psad/download/psad-
2.2.1.tar.gz

tar -zxvf psad-2.2.1.tar.gz

cd psad-2.2.1

./install.pl

exit
```

After installation some main settings were changed to finish the install.
In file /etx/psad/psad.conf the author’s own email was changed where the program will sent notes, and also ENABLE_AUTO_IDS and ENABLE_AUTO_IDS_EMAILS were enabled.
In iptables four new rules were added:

```
iptables -A INPUT -j LOG

iptables -A FORWARD -j LOG
```

```
ip6tables -A INPUT -j LOG

ip6tables -A FORWARD -j LOG
```

For getting the signature file updated and reload PSAD these commands were run:

```
psad -R

psad --sig-update

psad -H
```

To check the status of PSAD:

```
psad --Status
```

9.1.9 SELinux -AppArmor

AppArmor is Linux network application security system, it protects the operating system and applications from attacks. More information about these two can be found earlier in Ubuntu Server Guide – AppArmor at chapter 3.7 installing AppArmor:

```
sudo apt-get install apparmor apparmor-profiles apparmor-
utils
```

Checking status:

```
sudo apparmor_status
```

AppArmor directory is kind of configuration directory and place where the all of the profiles are stored.

```
/etc/apparmor.d
```

Setting profile into the enforcing mode is carried out with:

```
sudo enforce <application_name>
```

and into the complain mode:

```
sudo complain <application_name>
```

In order to get profiles updated AppArmor had to be restarted:

```
sudo /etc/init.d/apparmor restart
```

10 SCANNERS AND ANALYZERS

10.1 Rootkits

RKHunter and CHKRootKit are doing the same thing, checking if the system has any known rootkits. Although they do the same thing it is good to have them both, and because the installation is so easy it does not take much more time than installing just another one.

```
sudo apt-get install rkhunter chkrootkit
```

To run chkrootkit:

```
sudo chkrootkit
```

To update and run RKHunter:

```
sudo rkhunter --update

sudo rkhunter --propupd

sudo rkhunter --check
```

10.2 LogWatch

Logwatch is a customizable log analysis system. Logwatch parses through the system's logs and creates a report analyzing areas that are specified.

LogWatch is a program which can make reports about log files in a server. With LogWatch administrators can easily see what programs have been installed, reinstalled, removed or updated, how the mailing system is working, how the “sudo” rights have been used and who has been using them a great deal of other useful information.

Installing and running LogWatch:

```
sudo apt-get install logwatch libdate-manip-perl

sudo logwatch | less
```

Way to send report straight to email, with this syntax the report will be written from last 7 days.

```
sudo logwatch --mailto mail@domain.com --output mail
--format html --range 'between -7 days and today'
```

10.3 Nmap

Network Mapper (Nmap) is a security scanner which has many different features (OS and version detection, port scanning, host discovery). Administrators can use Nmap to check which ports are open and how much information other people can get from the server and network that it is serving.

Installing Nmap and few different scans:

```
sudo apt-get install nmap
```

Scan your system for open ports with:

```
nmap -v -sT localhost
```

SYN scanning with the following:

```
sudo nmap -v -sS localhost
```

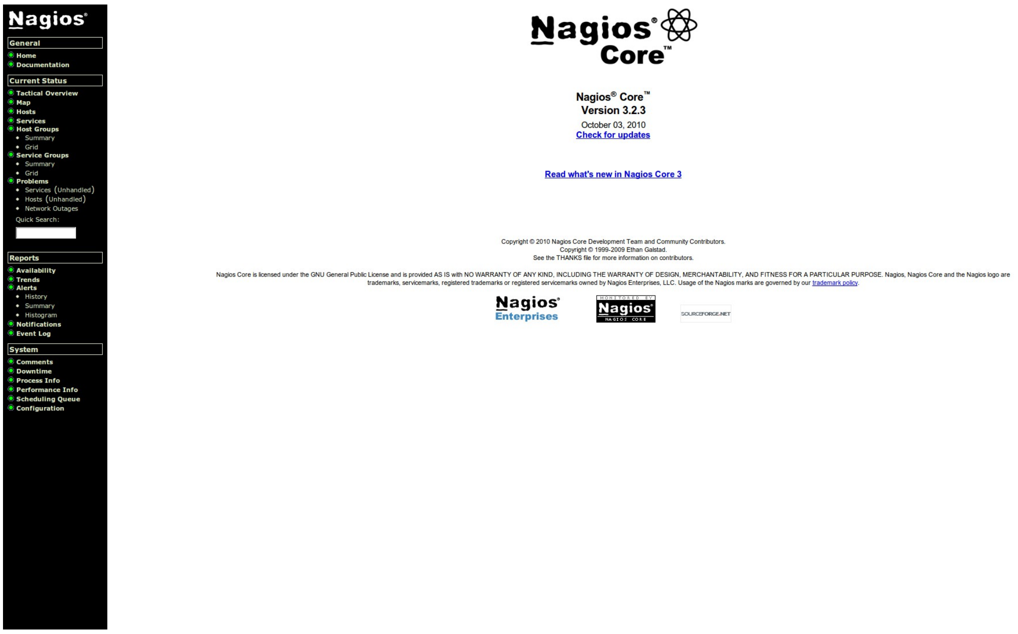
11 INSTALLING AND SETTING MONITORING TOOL NAGIOS

Nagios is a powerful monitoring tool which helps administrators to easily get information about server and their users.

Installing Nagios:

```
sudo apt-get install -y nagios3
```

After selecting and setting email and password for email what was to be used, the installation was ready. The next step was to open a web browser and go to localhost/nagios3 and log in. Nagios will automatically add information from localhost and does load, current users, disk space, HTTP and SSH checks.



Nagios® Core™
Version 3.2.3
October 03, 2010
Check for updates

Read what's new in Nagios Core 3

Copyright © 2010 Nagios Core Development Team and Community Contributors.
Copyright © 1999-2009 Ethan Galstad.
See the THANKS file for more information on contributors.

Nagios Core is licensed under the GNU General Public License and is provided AS IS with NO WARRANTY OF ANY KIND, INCLUDING THE WARRANTY OF DESIGN, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Nagios, Nagios Core and the Nagios logo are trademarks, servicemarks, registered trademarks or registered servicemarks owned by Nagios Enterprises, LLC. Usage of the Nagios marks are governed by our trademark policy.

Figure 2. Nagios main page

Nagios was needed to accept external commands for example to recognize problems and add some comments.

The line was changed as follows:

```
check_external_commands=0 into to
check_external_commands=1.
```

In file `/etc/nagios3/nagios.cfg`

Getting Nagios work properly a line in a file needed to be changed `/etc/group`

```
line: nagios:x:114 to nagios:x:114:www-data
```

Some rights for the Nagios file were changed next:

```
sudo chmod g+x /var/lib/nagios3/rw
sudo chmod g+x /var/lib/nagios3
```

After that Nagios was restarted:

```
sudo service nagios3 restart
```

12 RESULTS AND CONCLUSIONS

12.1 Results

The bandwidth limit works fine, and even if there is downloading or uploading going on there is always enough bandwidth to normal internet use. Keeping the internet on all the time is still not so reliable, because other ISP is reconnecting and changing IP address multiple times a day. There should be enough security to prevent basic attacks and threads from single users or small groups, larger attacks cannot be tested at this time.

12.2 Problems

Setting and testing during a work day proved to be impossible for some settings, because internet connection or server needed restarting and that would make other employees stop their work. Scripting language was much more difficult than first excepted.

12.3 Conclusion

This setup took much more time and results than I first thought as far as the amount of work is concerned. I think I can say that most of the goals of this thesis were almost finished. A great deal of programs was installed and configured; however, proper testing will take place in the future when there might be some real cyber-attacks and many employees are making weight to network. Now I can only say that the server ”should be working as planned”, but I cannot be sure.

REFERENCES

DNS. Referenced 08.05.2013
http://www.webopedia.com/TERM/D/DNS.html

How to secure an Ubuntu 12.04 LTS server. Referenced 06.05.2013
http://www.thefanclub.co.za/how-to/how-secure-ubuntu-1204-lts-server-part-1-basics

Iptables(8) - Linux man page. Referenced 08.05.2013
http://linux.die.net/man/8/iptables

Network. Referenced 06.05.2013.
http://www.thefreedictionary.com/network

PPPoE. Referenced 08.05.2013
http://www.webopedia.com/TERM/P/PPPoE.html

PSAD. Referenced 06.05.2013
http://www.cipherdyne.org/psad/

Katherine Noyes, Which Linux Distro Is Fairest of Them All? Ubuntu, Survey Says. Referenced 09.05.2103
http://www.pcworld.com/article/254516/which_linux_distro_is_fairest_of_them_all_ubuntu_survey_says.html

Zero Day Exploit. Referenced 06.05.2013
http://www.webopedia.com/TERM/Z/Zero_Day_exploit.html