Automated Router and Switch Backup

Pierre Bjurdelius, Andreas Bjurdelius, Alexander Blomqvist

Network project, 7.5 credits

Halmstad 2014-02-06
Automated Router and Switch Backup

Network Project

2013

Author: Pierre Bjurdelius
Author: Andreas Bjurdelius
Author: Alexander Blomqvist
Supervisor: Olga Torstensson and Malin Bornhager
Examiner: Olga Torstensson and Malin Bornhager

School of Information Science, Computer and Electrical Engineering
Halmstad University
PO Box 823, SE-301 18 HALMSTAD
Sweden
Preface

With the help of this thesis we have gained a greater understanding about Linux systems, the backup and handling of cisco configuration files, which will help us in our professional future.
When the lab environment was set up and all tests were successful, the same setup was used in a company with worldwide routers and switches. The setup was done with this documentation as reference.
We want to mention and send our thanks for all tips and help to our supervisors and tutors, Olga Torstensson and Malin Bornhager.
Abstract

Today's companies are growing in a steady pace, with more and more network devices added to the network it is very important to keep track of and monitor the status of devices. Even though the wireless evolution has come, it all depends on the wired connections to supply a continuous connection to the rest of the world.

This thesis explores, tests and informs about creating a functional system that automatically creates backups of configuration files from network devices and how to troubleshoot networking problems and maintain a network to keep it in good shape.

Even though many companies have manual backups of router and switch configurations, the possibility to have this part automated should be desired by most companies. It can open up for the administrators in the company to have more time over to help the employees that are experiencing problems at the same time as the automated system eliminates the possible errors that a human can cause. Of course one can see it the other way, that it takes away manual labor for the employees, but it is just a small part of the job yet it is so very important that making this service automated is a good choice for a company.

Integrity is proven by the means of backups and by the option to see the difference between the previous backups and the most recent.

The three of us have worked as a group to do all tests and to write the documentation. After working with a couple of companies it is clear that well functional backup systems of network devices are not as common as it should be. Companies that do take backups of the network devices often do this manually. When seeing this it makes sense to use a reliable system that uses revision handling so it is easy to see the recent changes made to the devices.

The results ended up in a working automated backup system for routers and switches. The automated system is running Debian and connects to all the routers and switches in the network to collect the configuration files with the help of rancid. The thesis also explains the functions of concepts such as disaster recovery and different maintenance models.
Contents

Preface i
Abstract iii

1 Introduction 1
  1.1 Goals ......................................................................................................................... 1
  1.2 Objectives .................................................................................................................... 1

2 Method 2

3 Troubleshooting and maintenance of a network 3
  3.1 Maintenance models....................................................................................................... 3
    3.1.1 Interrupt-driven maintenance ................................................................................. 3
    3.1.2 Structured network maintenance ............................................................................. 3
  3.2 Maintenance Tasks ...................................................................................................... 4
  3.3 Disaster recovery plan ............................................................................................... 4
  3.4 Network documentation ............................................................................................. 5
  3.5 Troubleshooting a network environment .................................................................... 5
    3.5.1 Top-Down............................................................................................................... 6
    3.5.2 Bottom-Up .............................................................................................................. 7
    3.5.3 Divide and conquer ............................................................................................. 7
    3.5.4 Move the problem ............................................................................................... 7
    3.5.5 Follow the path ................................................................................................... 8
    3.5.6 Spot the differences ............................................................................................ 8
  3.6 OSI model .................................................................................................................. 8
  3.7 Helpful troubleshooting tools .................................................................................... 9
    3.7.1 Telnet .................................................................................................................... 9
    3.7.2 Debug .................................................................................................................. 10
    3.7.3 SNMP .................................................................................................................. 10
    3.7.4 SPAN & RSPAN ................................................................................................. 10
    3.7.5 Wireshark ............................................................................................................ 11

4 Software research 12
  4.1 VMware and Debian ................................................................................................. 12
  4.2 Rancid ....................................................................................................................... 12
  4.3 CVSweb ..................................................................................................................... 13
  4.4 Crontab ...................................................................................................................... 13

5 Implementation and results 14
  5.1 Installation of VMware and Debian ............................................................................ 14
  5.2 Implementation of Rancid and CVSweb ................................................................. 14
  5.3 Crontab implementation .......................................................................................... 16
  5.4 Testing and troubleshooting .................................................................................... 16

6 Conclusions 20

References 22
Chapter 1

1 Introduction

A company always lives with the risk of a possible router or switch failure. Therefore the ability to create backups of configurations is a vital part in any sort of industry. However the way that backups of configurations are created today involves manual backing up. This is not a very efficient way and there should be better ways of doing it.

With the modern technology that is available in the society today, it should be a standard available to have an automated system setup for greater efficiency and a more secure network.

As a company’s network continue to grow, it gets more difficult to manage every router and switch. The manual way to make sure that there is an up to date backup of the configuration available for a fast recovery of the network device is a time consuming task, especially when you have a large system of network devices to manage.

In the event of a network failure or a crash of a network device, it’s vital to the majority of all companies to get the network devices up and running with the correct configuration as soon as possible.

1.1 Goals

The project goal is to implement a system that can automatically generate backups of switch and router configurations. The configuration will be saved on a virtual server with a set time interval of when to generate backups. When this part is functioning properly it is planned to set up so that there is a program on the server that can compare the configuration files, to find any possible changes in the configuration. The main issue of this project is to see if it is possible to make this work seamlessly, while considering the possible risks and problems that might arise from this kind of setup.

1.2 Objectives

The main objective of this project were to figure out a way to with the help of a virtual server running Debian, automatically create backups of multiple routers and switches, without the need of accessing each device manually, making a company network more efficient.
Chapter 2

2 Method

The main purpose of the project is to create a system that automatically accesses the configuration of routers and switches, creating a backup and saving it on a remote server. When this is working well, the next step in the project is to compare configurations, to easily see what has changed between the backups. To achieve this, the server that is going to be used will be a virtual server running Linux, Debian.

The reasoning behind the usage of a virtual server is as a security measure, if one of the hosts in the virtual cluster would go down/crash, the virtual server will automatically move between the hosts in the cluster. This way the server will always be up and running, even in the event of hardware failure.

The reason for using a Linux server is that the operating system is free and all members of the group has been using Linux before, and that makes things easier in the work with the server.

There was a decision to choose between the two distributions Debian and Ubuntu. The group decided to use Debian as there was a minimal installation image available for us to use that did fit very well with the project. Of course there are other systems available on the market but Debian seemed to be the best fit for the project and it was the operating system that everyone in the group agreed on would work best for a project like this. One of the reasons for it to fit great with the project was that the operating system could be a small image due to us only needing the text based operating system and no graphical interface.

Our group plan to solve this by using software called VMware player, which is an application that can simulate different operating systems, as long as you provide an installation file of the desired operating system. The software will be used to create a virtual Debian server; Debian is a Unix operating system, very similar to Linux. The server will use a minimalistic installation image, this to be able to keep the installation under our control and avoid installing unnecessary applications that would increase load time. To gather the configuration information from switches and routers, the group plans to take advantage of the rancid repository. To run rancid automated in the middle of the night when it does not interfere with the business hours, crontab will be used within the Debian server.
Chapter 3

3 Troubleshooting and maintenance of a network

To be able to troubleshoot and maintain a network is essential to a company. There is always a possibility of network failure, but you might be able to minimize the risk of failure with a good maintenance model. Maintaining a network includes tasks such as making backups of the network and upgrading devices or software[1]. Having a well thought out and structured maintenance plan will increase the uptime of a network and reduce the amount of outages.

3.1 Maintenance models

A network maintenance model is a pre planned way of how to approach the network in a maintenance point of view. There are of course different models out there and two much known models are interrupt-driven and structured network maintenance.

3.1.1 Interrupt-driven maintenance

Interrupt-driven maintenance is a model that works well in small networks but is not recommended to be used in larger networks. The way this model is designed is that the network is maintained when something has happened or is reported to the network engineer. So it is first when the network experiences problems that it will be handled. Even though it can be good for small networks, to save money, it is also the reason why not to use interrupt-driven maintenance in a larger network. A lot of problems can occur at the same time and it can be too much to handle, resulting in a lot of downtime for the network. With the use of this model, important tasks might end up being delayed, due to the model not following any priority of what to maintain first[1].

3.1.2 Structured network maintenance

Structured maintenance is the other well known model and is quite different from interrupt-driven maintenance. Structured maintenance is a lot more smooth and planned out, resulting in not as many small problems occurring. There are a lot of benefits of using the structured maintenance model and one benefit is that it is a more cost effective way to maintain your network in the long run for a larger company. Having small errors that can crash the network occurring a lot can result in a lot of lost revenue for the company.
This model helps to counteract problems like that with the help of performance monitoring and planning of the network capacity[1]. Capacity planning is a way to measure the loads on the network. It will let you measure the peaks and average loads which can help to find links that can be in need of an upgrade.

Keeping an eye on the network results in a more secure network, network monitoring is therefore a huge part in having a well maintained network. To help with monitoring the network there are different tools to use and a good tool is the logging services. The logging services will notice events that happen on the device and it can be linked to store the logs on a server for easier access[1]. When using the logging services there are different levels that can be set, deciding how sensitive the logging will be. There are eight levels, stretching from zero to seven, where zero will only log emergencies and seven will log pretty much anything that happens on the device. Usually a good recommendation can be to set the logging services to level 4 which is equal to warnings, resulting that any warnings or anything worse than that will be logged. If the administrator decides to not set a specific level for the logging, it will log everything as default.

### 3.2 Maintenance Tasks

A network administrator should create a collection of maintenance tasks that should be looked through and carried out on a regular basis to prevent major issues. In the collection of tasks there should be sets of:

- Show commands to be able to verify current status and settings of the system. This is probably the most important part of the collection.
- Quick configuration information to refer to when fast changes are needed.

### 3.3 Disaster recovery plan

Something a larger company is recommended to have is a disaster recovery plan. This plan will help the network to get back up and running in the shortest time possible after a device failure. Of course the network should never go down without it being planned but sometimes it can happen. If the company has some money to spend, redundant links to network devices can help keeping it up and running. But what if it still goes down, then a disaster recovery plan can be great to have. A plan like this involves everything from whole new hardware to replace the broken one, to just knowledge of how to install software and configure devices[1]. When creating a disaster recovery plan the company has to decide what is required to have a recovery in a short enough time to not lose revenue.

Recovering the network usually means that you either have to create a new configuration file from scratch or have an old one backed up that you can quickly load. Although it can be hard to always have the latest configuration from every router and switch backed up in a larger network. This is where an automated system for backing up devices can help the company a lot, making sure that there are always the latest configuration update of every router and switch saved on a server.
3.4 Network documentation

It is very important to document the network. If you keep the network well updated in your documents, it will be a lot easier to find errors or just to see how the network connected over all links. There are a lot of recommended items that should be included in the network documentation, and some of them are extra beneficial to have in the documentation.

A network topology is always good to have included, for an easy and fast overview of the network. With the topology available there can be a lot of saved time when planning of where to connect new devices. Having the configurations of files saved in the documentation is always a good idea. Having configuration files available to be looked at can help find errors in a network that can be corrected much faster than without having the files saved in the documentation. The documentation can be saved in binders on a desk or even collected together in a wiki page. Using a wiki page might be a lot easier, a wiki page is created to be easy to edit by the users.

3.5 Troubleshooting a network environment

Even though your goal is to never have any problems with the company network, it can end up happening at any given time. So when the maintenance is not enough and the network goes down the administrator needs to be able to find the faulty device or cable or what it can be that is blocking the network. A person with good knowledge in troubleshooting principles and methods is more likely to solve a problem faster so it is always good to have personnel educated in this area of expertise. There is a good flow chart that you can follow when troubleshooting a network to maximize the efficiency and make it more structured.

The so called flow chart consists of a lot of steps that the network administrators can follow to create a smoother troubleshooting approach. Of course it starts with defining the problem and ends with solving the problem, but the more interesting parts of the approach is in between those two steps.

At first the administrator starts by gathering information of what has happened. There might be a ticket sent to the administrator to say what is wrong with the device or network but usually that is not enough information to pinpoint the error. So to gather more information the administrator must decide which devices to collect information from and how to do it. While gathering information it can be good to decide which method to use while troubleshooting the problem. The administrator can issue tests on the network to find faulty areas. If the problem is within an application the administrator can decide to implement the top down troubleshooting method or if it is assumed that there is a problem with cables the bottom up method can be a good decision[1].
When the administrator has gathered information, it is time to analyze what has been gathered. While analyzing the information the administrator can get a deeper understanding of the problem that has occurred.

If the analysis went well there might be a lot of theories that can be eliminated from the list of possible causes for the problem. When the administrator has more of an understanding of the problem, he or she can try to formulate a hypothesis of what the problem actually is. If there are many hypotheses created, the administrator should rate all the hypotheses with a probability rate of how likely they are to have happened. This rating should be created just to make it easier to start with the most likely problem and possibly saving time.

When you have created a list of hypotheses you can finally start testing them by creating a solution to the first hypothesis and trying it out to see if it fixes the problem with the network[1]. It is important to note that the hypothesis might indicate for example that the problem is on the side of the internet service provider. When the problem is on the outside of the responsibilities of the company, it can be given to whoever is responsible of the area[1]. If the first hypothesis does not solve it, then try with the next one in the list. In the end, if not a single hypothesis worked, start over from the beginning and try to come up with new hypotheses. However it is recommended to have eliminated most of the hypotheses to save time during the testing of the solution, but it is not always possible to eliminate all but one hypothesis.

To accommodate this flow chart style of troubleshooting there are a lot of different methods, these methods sets up the troubleshooting process in different ways. One method might be to start from the top of the network and another might be to start from the bottom. There are a lot of different methods and they all have good qualities that can help out in different scenarios.

### 3.5.1 Top-Down

The top down troubleshooting method is created to start troubleshoot a network from the “top”, meaning that you work through the OSI model. The OSI model has seven layers and the top layer is the application layer, and the bottom layer is the physical layer. With the top down method you start by troubleshooting the application layer. A normal start when using this method is to troubleshoot problems with the user application. If you do not find any problem in this step then you keep moving through the OSI model. Usually this method is recommended only be used when the administrator is certain that the occurring problem is close to the top of the OSI model[2].
3.5.2 Bottom-Up

Bottom up is working the same way as the top down method, but instead the troubleshooting starts at the physical layer. Troubleshooting the physical layer first can solve problems that are not inside of the devices or applications, it might just be a loose or broken cable in the network. If you are lucky this method can be a big timesaver, if the problem ends up being just a loose cable for example. So if the network administrator suspects the problem to be physical he or she should use the bottom up method. A downside with the bottom up method is that since it starts with the physical layer, the administrator will have to check every device for failure, that might take a lot of time in a large network[3]. A good reason to use the bottom up method is that most often the network problems are hardware related, making this approach perfect a lot of times unless the administrator knows the problem is elsewhere[1].

3.5.3 Divide and conquer

When using a divide and conquer method the administrator begins by investigating one of the middle layers, usually the network layer. This method can be used if the administrator is not sure of where the problem is in the network. The administrator can for an example issue the ping command and see if it works to reach the other devices in the network. If the ping is successful the administrator can assume that the layers below the network layer are working correctly. This method can help deciding where to troubleshoot in the next step. If the ping is successful, you can use the bottom up method and starting from the network layer to continue the search to find the problem and if the ping does not work you can use the top down method and start from the network layer as well[1].

3.5.4 Move the problem

Move the problem is a troubleshooting method that can be used to find the problem in the network. The administrator replaces a component with another to see if the problem disappears or if it remains. If the problem remains it is clear that the device was not the faulty device. Instead of moving a whole device, the administrator can also switch places of cables to see if the cables are broken. With the use of this method it is possible to isolate the problem and if not solve it completely, be able to change to another method fitting the results that the administrator have found. The drawback with this method is that it is assumed that the problem is in a single device and not spread onto more, making it hard to find all the problems[1].
3.5.5 Follow the path

Follow the path is a troubleshooting method that literally follows the path of the traffic in the network. This is not often used by itself but used to complement one of the other methods[1]. To use this method, traceroute can be a good help. Traceroute shows how far the ping reaches and thus making it possible for the administrator to pinpoint where the faulty device might be[4].

3.5.6 Spot the differences

The spot the differences method is as the name implies a method to find an error by just looking. The administrator compares configuration files or devices with each other to find the problem. The nonworking configuration file can be compared with a working one to look for something in the file that might be wrongly configured. If an error or a missing command is found it can be corrected and then tested to see if it works, if it doesn’t help the administrator can keep comparing the files to see if any other error can be found. A problem with this method is that even if the device ends up working after the changes, there is no real explanation of what was wrong, or how it occurred in the first place[1].

3.6 OSI model

The Open System Interconnection model, most often referred to as the OSI model which defines a framework for the network. There are seven different layers in the OSI model, all which involves different tasks in the network. From layer one in the bottom to layer seven in the top, each layer is dependent on the layers below to work and makes it possible for the layer above to work. Due to the layers being dependant on each other, it can be used to troubleshoot a network environment. By testing if a layer works, the administrator can then exclude the layers below if the tested layer works.

The seven different layers that build the OSI model are the following from the top down; the top layer is known as the application layer and is noted as layer 7. As the name indicates, it provides the application services for the network.

Layer 6 is the presentation layer which translates the data between application and network formats so that the application layer can understand what is being sent and vice versa.

Layer 5 is the session layer which handles the connections between applications.

Layer 4 is the transport layer; this layer provides a reliable way of transferring data in the network, making sure that no data is lost in the transportation. A well known transport layer protocol is the TCP (transport control protocol).

Layer 3 is the network layer; this is the layer that provides switching and routing technologies. Every device in the network has an address (Internet Protocol) and the network layer helps with the forwarding of information in the network environment.
Layer 2 is the data link layer which is divided into two different sub layers, the Media Access Control (MAC) layer and the Logical Link Control (LLC) layer. At this layer the data is encoded and decoded into bits to be sent over the network. Layer 1 is the physical layer and in short, it involves the hardware of the network such as the cables and network cards in the network environment[5]. With the help of the OSI model, an administrator can get a better understanding of how the network is built and during a troubleshooting procedure be able to exclude certain parts of the network for better efficiency. Below you can see a picture of the OSI model, with the application layer (layer 7) in the top and the physical layer (layer 1) in the bottom.

![OSI Model](image)

Picture 1: The OSI Model

### 3.7 Helpful troubleshooting tools

There are many different tools available to the administrator to troubleshoot the network. Some tools are built in into the device and others are used as a separate application. Of course there are a lot of tools available but some of the tools might be used more than others. This chapter brings up some tools that are more well-known, tools such as telnet, debug and SNMP.

#### 3.7.1 Telnet

Telnet is a text based protocol and application that makes it possible to connect a device to another device, such as from one router to another. Telnet can be useful when working with more than one router or switch at the same time, using telnet in between them to not need to move the cable. Telnet can also be used to try out connections to different ports in the network. By adding the port number in the end of the telnet the administrator can test if ports are open or closed[1], making it a good and simple way to troubleshoot ports and also the transport layer of the network. For connection to devices SSH is recommended to use nowadays due to that telnet does not encrypt any traffic, making it vulnerable to attacks.
3.7.2 Debug

Debug is a built-in troubleshooting tool in the routers and switches and can help finding errors in the configuration of the network. Debugging can be turned on and off on different protocols that the device is using, such as if the administrator turns on IP packet debugging the terminal will display the messages travelling between the hosts over the network[1].

Basically the debug tool displays information about the selected protocol that the administrator turns on with the help of the “debug …” command. However, debugging large protocols such as IP packet, the network has a possibility to hang due to the massive amounts of packets travelling over the network. A way to try and avoid this is to create an access-list that only allows certain matching packets to be shown in the debugging[6].

The administrator can for example issue the command “debug ip ospf events” which will trigger messages about the OSPF network, possibly showing mismatches in the configuration on different sides of the network[7].

3.7.3 SNMP

Simple Network Management Protocol is a network management TCP/IP protocol. With the help of SNMP an administrator can monitor the network easily and get notifications about the devices[8]. SNMP collects information about the network devices and stores it in the management information base and makes all the information available to a main device called the network management station (NMS), such as the administrators PC, and the network devices that are being used to collect information is called SNMP agents. With the use of Simple Network Management Protocol the administrator can keep an eye on the network from his or her own computer[9]. Using SNMP will help the administrators with a fast and easy overview of the network making it possible to see problems easier than moving to every device.

3.7.4 SPAN & RSPAN

Switch port analyzer and remote switch port analyzer are traffic monitoring systems that duplicates the network traffic on an interface of the device such as a switch to another interface. The difference between SPAN and RSPAN is that RSPAN can have its destination on another device, so it does not have to be local like it has to be when SPAN is in use, where the destination port has to be on the same device as the source that the administrator want mirrored[10].

SPAN and RSPAN are good ways of monitoring a single or multiple ports on a device, maybe one port is more important on the switch than the others because it leads to a server. Then setting up SPAN or RSPAN on that link can provide great
insight of what is sent over the connection and possibly find errors that occurs and therefore correcting the problems before the server crashes.

### 3.7.5 Wireshark

Wireshark is one of the most used traffic-capturing tools and it works in the way that it analyzes protocols that are sent over the network. Looking at traffic with a tool such as wireshark is called sniffing. When the administrator uses wireshark to analyze the network, he or she will see everything that is being sent over the network in the graphical interface. A sniffing tool captures a lot of information and can be hard to use without filtering\[1\]. Filtering for different protocols can be great while troubleshooting why different applications are not working correctly or even filtering for a special IP address to see just that traffic being sent. Even though this is a good way for an administrator to analyze the network, it can also be used in wireless attacks because wireless information is not encrypted. Keeping the wireless network safe from sniffing attacks is something that an administrator must keep in mind if there is a wireless network in use in larger networks so that not important information can be stolen.
Chapter 4

4 Software research

To understand what the different software is needed for the project to work, and how to be able to use them in the best way possible, the group decided to do some more research about them. Maybe even find out if a virtual or a real machine is best for a project like this. There were already some knowledge of what software should be used and in the subsections you will find out more about them.

4.1 VMware and Debian

VMware is a so called virtualization software, making it possible to run a program without first having to obtain expensive hardware products, which is a great benefit. Having a larger company, using virtual machines, can save a lot of money because each machine can simulate more than one application and operating system. VMware creates a fault tolerance environment, meaning that if a server goes down, it will start all of the applications that it was running on another machine[11].

Debian is an operating system based on the Unix family and is a free software, it can be installed and run in just a text based system as might be desired for companies, seeing how it can save a lot of space compared to installing the operating system with a graphical interface as well.

4.2 Rancid

Rancid stands for Really Awesome New Cisco config Differ and is an application that monitors a device’s configuration and keeps track of any changes made to it. It stores the collected data in CVS. Rancid’s support of login scripts and its capability to connect to a device via telnet or SSH makes it very appealing for the purpose at hand. Rancid supports most of the current switches and routers, such as Cisco, Foundry and Juniper devices[12].
4.3 CVSweb

CVSweb is a WWW interface based on CVS, which stands for “Concurrent Version System”. CVS works in the way that it compares versions of a file to find changes in it. The files are stored in a central repository folder on the server with CVS installed where they easily can be compared. This can be great for use when making an automated router and switch configuration backup. Seeing how it can directly compare two file versions to find changes between an old version and a new one[13]. On top of tracking changes made to the file, it also saves older versions of the file to ensure backup potential so they can be restored if needed. CVSweb adds an interface for a website, listing saved files from the central repository so that you can open them and see changes[14].

4.4 Crontab

Crontab is an application that lets you do actions at a certain time, set by the user. With this you can set a program to update at noon everyday for example. In this project, crontab fits in to be used in the way that it will tell rancid to make copies of the running configuration on the switches and routers.
Chapter 5

5 Implementation and results

Already by the start of the project the group had some knowledge of what software and operating system to use, but not all that was needed to finish the project. When talking it over, the group was most interested in using the operating system called Debian, since all the kinds of software that was planned to be needed was available there. At first it was thought about installing Debian directly on a computer, but there might be issues with uptime and just more work if something goes very wrong. Therefore it was decided to install the operating system on a virtual machine.

5.1 Installation of VMware and Debian

Together the team started the project by installing VMware player, it was decided to install a standard version of the player because the standard version was free. The VMware software can be found for download on their website. After this the group started installing the Debian image on the VMware player, the smallest image available was chosen as only the necessary items in the system was wanted to make it run properly. In the installation process the language that was chosen to use in the system was English language to ease the troubleshooting the group is expecting to do further in the project. The locales were set to Swedish to match our computers keyboards. Midway through the installation it is time to create accounts and passwords. The group decided to create one user account and password and one root password. That ends the installation of the operating system Debian. Now it's time to start the system in VMware player and login to the system. The first thing to do is to check that the system is up to date. To do this the team needed to sign in as root and run the command "apt-get update && apt-get upgrade". This will run through the system and check for updates and then install the updates if there is any.

5.2 Implementation of Rancid and CVSweb

Now that the virtual machine is up and it is running a working Debian installation, it was time to start installing rancid, which is the software program that is going to be used to connect to the routers and switches in the network to save configurations from the devices.
Chapter 5. Implementation and results

The command that was needed to be entered was “apt-get install rancid”, the standard way of installing software on a Linux based operating system. During the installation it might ask you if it is your first version of rancid that you are installing. You should answer yes here. You know that the installation is done when you are automatically returned to the normal prompt in the terminal.

Now it’s time to start configure the installation. To add a collection of routers or switches, uncomment the line list of groups in the rancid.conf file. This file can be found under /etc/rancid/. Here you can add some categories, like switches or routers. To make rancid create those categories configured in the rancid.conf file, run the following command as rancid user, /var/lib/rancid/bin/rancid-cvs. The command will create the categories as folders in /var/lib/rancid/.

Next step is to install a web server to present the data from the saved configuration files. To do this, the group decided to use the software called CVSweb in this project. If you want to install CVSweb, you can issue the command "apt-get install cvsweb" as root user.

When the installation is done you can test the web server by navigating to the server IP address in a web browser, however right now you will not see much here since nothing has been done with the web server yet.

For the server to be able to access a switch or a router by itself it needs to have user accounts and passwords to the devices. This is added to a hidden file that should be created in /var/lib/rancid/. The file should be named ".cloginrc" with the following configuration.

This configuration is based on dividing devices into routers and switches, by setting a default username/password and enable secret password together by using the service telnet.

# switches
add user cisco
add password {cisco} {cisco}
add method telnet

# routers
add user cisco
add password {cisco} {cisco}
add method telnet

To test the password configuration, the following command can be run as rancid, ./bin/clogin “IP address of a switch/router”. If this does not work, there might be problem with your configurations. The system should use the password file to log into the device.

Next up is to list the devices the system should access automatically and what fabric it is. This is done in the switch directory, there is a file called router.db which should be edited.

In to this file the name of the switch should be entered then a colon and the manufacturer of the device then another colon and the status of the device, up/down.
The status is used if you want to temporarily remove a device from the backup system. To be able to reach the new device through a name, the device must be added to the hosts file located in /etc/hosts. Here the IP address and the hostname should be entered. Now that there is a name added for an address, you can try to ping the name instead of the address and it should work the same way.

5.3 Crontab implementation

To enter crontab you run the command "sudo crontab -e". You will now be in the crontab configuration file. To get crontab to run the rancid command every day at 00:00 you type the following line in the crontab configuration file.

```
00 00 * * * /var/lib/rancid/bin/rancid-run
```

5.4 Testing and troubleshooting

To be able to do some tests with the Debian server it was needed to connect the virtual machine’s host to a switch in our network. This sounds very easy but it actually proved to be a little more complicated than expected. First off the group started by creating a DHCP server/pool on the switch that will be used for testing, just for us to easily set the IP addresses to know what addresses are available for the devices in the network. Now the switch could ping the host but for some reason it could not reach the virtual machine, which was thought to be weird since they pretty much are the same computer. To solve this problem it was decided to create a bridge between the network on the host and the virtual network card on the virtual machine.

When the bridge was created over the network cards, the ip address was automatically set to 192.168.127.1, which was not an address in the same range as the DHCP pool was set to, so the group decided to change the DHCP pool on the switch to that network as well to make it as easy as possible. However, later, after looking around on for some extra information over the internet, a member of the group actually discovered that the IP address that was automatically set could be removed on one of the other virtual network cards listed on the hosts list of network connections, making it so that the host was receiving its IP address from the DHCP pool again. Since it was receiving an IP address from the DHCP now, the team decided to keep the new pool and not create our old pool again. After all, this is just a test run of our server. In picture 1 you can see the setup for the first test that was issued.

![Topography for our first test](image-url)
Now when this is done, the switch can finally ping the virtual machine. To test that the server and Rancid software can reach the switch you can run the following command:

```
/var/lib/rancid/bin/rancid-run
```

This command tells rancid to run the files that were created by the group, looking for devices has been added by the group, that it will try to connect to.

The test that was issued created a new configuration file in the switch directory on the virtual machine. Unfortunately it is only an error message in the configuration file. The error message states that the file should not have world read write permission. Luckily this was an easy error to correct, it is just to change who can use the file. By changing the owner to rancid with the help of the command “chown” on the switch folder and everything below that folder as well and remove the possibility for anyone else than the owner to read or write in the file with the “chmod” command. Now it was time to start another test and the group did get yet another error message. This time it says that there is no password. The password file used by cisco devices is located in /var/lib/rancid and is called .cloginrc. After a quick look in the file it was noticed that there is one thing missing, and that is an * sign after the text user, password and method. The “star” sign is interpreted by the file as “all”. That means all devices that the server will try to connect to will use this credentials.

The list now looks like the following:

```
# switches
add * user cisco
add * password {cisco} {cisco}
add * method telnet

# routers
add * user cisco
add * password {cisco} {cisco}
add * method telnet
```

After this it was time to run the test again and now it is working, the server has created a complete backup of the switch configuration.

With the backup function running as intended next step is to get the web server up and running and presenting the contents of the backups.

During an attempt to access the website, the group just got an error. After some hours of troubleshooting the error was finally found in the configurations file for cvsweb, the file with the error in it was /etc/cvsweb/cvsweb.conf. There was a missing row with links, which is used for cvsweb to point what information was wanted for cvsweb to show on the web server. In the configuration file, under “@CVSrepositories = (“, the following was added “‘rancid’ => [‘rancid devices’, ‘/var/lib/rancid/CVS’]”, this tells cvsweb what should be shown on the website, in this case the group want to show their rancid devices, which will list all devices that rancid connects to and collects information from.
Now when an attempt to connect to the website was issued, through the IP address, address/cgi-bin/cvsweb/CVSROOT/, you can see the correct information shown on the website.

The group notices that there still is a problem with the layout, actually the icons are missing. There seems to not be any icons in the www folder, which is where such things should be saved. There is also a missing CSS file, which makes sense, seeing how strange the website looked it could be easy to figure out that there was no CSS file available. If you do not know, the CSS file is a style sheet, used to define the layout for a website.

Go get the layout working it was decided to start creating our own style sheet and kept working on it for a while, but it never got to the point that it looked the right way because of a problem that occurred with symbolic links. Though after a while of working with this, a member found the icons and the actual CSS style sheet in another folder in the system, they were hidden under /usr/share/cvsweb. Now it was just needed to move the icons and the CSS file to the right folder, /var/www/cvsweb. After this the server was restarted again and now it actually looked pretty nice, showing icons and a nice layout.

<table>
<thead>
<tr>
<th>File</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVSROOT/</td>
</tr>
<tr>
<td>seget/</td>
</tr>
<tr>
<td>segot/</td>
</tr>
<tr>
<td>sevxo/</td>
</tr>
</tbody>
</table>

![Picture 3: The root structure of the website]

This was mainly the purpose of making it more user friendly, seeing how otherwise it could take a lot more time to use the website to check for changes in the configuration files.

<table>
<thead>
<tr>
<th>Current directory: [rancid devices] / seget / configs</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
</tr>
<tr>
<td>![Parent Directory]</td>
</tr>
<tr>
<td>![Attic/ [show]]</td>
</tr>
<tr>
<td>seget-1gb-konstruktion</td>
</tr>
<tr>
<td>seget-2970_data2_reserv-datahall</td>
</tr>
</tbody>
</table>

![Picture 4: This is the structure of how the config files are shown]
Chapter 5. Implementation and results

Now it is easy to navigate through the configuration files and check for changes in the different configuration files.

As shown above, it is easy to pick out the changes in the configuration files with the style sheet in place.

Now that all of this is fully working it was time to conduct a larger test with one router and three switches connected to our Debian virtual machine. To do this the group changed the .cloginrc file to contain the new devices of the network, and then it was time to try and run rancid again. The test was successful, our web interface now showed four new backed up configuration files. Below you can see the topology of our "large" test network.

<table>
<thead>
<tr>
<th>Rev</th>
<th>Age</th>
<th>Last log entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3</td>
<td>5 weeks</td>
<td>updates</td>
</tr>
<tr>
<td>1.6</td>
<td>2 weeks</td>
<td>updates</td>
</tr>
</tbody>
</table>

Picture 5: This is how the revision handler is showing the different revisions of the config files

Picture 6: This is how the diffs are shown from two different revisions of the current configuration file.

As shown above, it is easy to pick out the changes in the configuration files with the style sheet in place.

Picture 7: "Large" test topology
Chapter 6

6 Conclusions

The purpose of this thesis was to create an automatically backup system with revision handling. This thesis explains how to troubleshoot and maintain a network along with how to create an automated backup system for routers and switches. The different maintenance models are explained, interrupt-driven maintenance where the administrators only maintain the network when something actually goes down. And structured maintenance where the administrators work in the way to prevent interrupts in the network for a better uptime. The thesis also covers the different troubleshooting methods and gives the administrator more insight about the most efficient ways to troubleshoot their network, such as Top-Down and Divide and Conquer.

Different tools that can be useful while troubleshooting was explained in the next part of the thesis, of course there are a lot of tools available to use while troubleshooting but some tools are really good, such as telnet to test connection between devices and the debug feature on the devices to help find errors in the routers and switches.

The group created an automated router and switch backup system using basic programs such as VMware, available to anyone with internet access. VMware was used to create a virtual computer or server, just so that they could create a computer with the Debian operating system which is a Unix based system. After the virtual server was up and running, the group implemented rancid on the device. Rancid is a software that lets the user assign routers and switches in a list, when rancid later was executed the device would access every router and switch in the list and save their configurations.

To show the configuration files that rancid saved, the group implemented cvsweb to create a web interface where the configuration files could be shown in a browser locally with a nice layout, which made it easier than listing every configuration in Debian.

Since the group wanted the system to be automated, they decided to implement crontab. Crontab is a software program that lets the user set a program to be executed based on a set time interval, so that our backup system could create backups during the night.

A way to broaden the project even more and make it better than it already is, would be to implement Tacacs+ on the virtual machine. The Tacacs+ server would manage the accounts of the routers and switches, so that they would be gathered in a central point instead of having to create different accounts on every router and switch.
During our search on the internet a member stumbled upon an application for sale that works as an automated backup for different devices. The application’s name was CatTools and is created by the company called SolarWinds. In comparison, this system costs $630 and ours is completely free of charge. However, their application also works as a syslog tool, alerting when changes are made in the network which is something our tool does not do. Notable differences is that their application is programmed with a graphical user interface and works with Windows and our application is text-based during the configuration part and then the website lists all the configurations that have been backed up.
References


[2]. The Top-Down Troubleshooting Approach, Amir Ranjbar, cisco press (Last accessed 20/12-13)

[3]. The Bottom-Up Troubleshooting Approach, Amir Ranjbar, cisco press (Last accessed 20/12-13)

[4]. Follow the Path, Petri IT Knowledge (Last accessed 20/12-13)
http://www.petri.co.il/ccnp-tshoot-cisco-troubleshooting-techniques.htm

[5]. The 7 Layers of the OSI Model, Webopedia (Last accessed 27/1-14)

[6]. Using the debug ip packet Command, Cisco (Last accessed 28/12-13)

[7]. Debug ip ospf events, Cisco (Last accessed 28/12-13)

[8]. The Network Zone, kratosnetworks (Last accessed 28/12-13)

[9]. SNMP Basic Components, Cisco (Last accessed 28/12-13)

[10]. Understanding SPAN and RSPAN, Cisco (Last accessed 29/12-13)

[12]. RANCID - Really Awesome New Cisco config Differ (Last accessed 2/12-13)
http://www.shrubbery.net/rancid/

[13]. Advantages of CVS, CVShome (Last accessed 3/12-13)
http://cvshome.org/eng/vorteile.html

[14]. What is CVSweb?, The FreeDSB Project (Last accessed 5/12-13)
http://www.freebsd.org/projects/cvsweb.html#about
Pierre Bjurdelius:
Former network student, going back to finish the project. Working for the company that we implemented this project for.

Andreas Bjurdelius:
Network student, graduating in 2014. Excited about creating a very well functioning project that is being used in a company today.

Alexander Blomqvist:
Network student graduating in 2014. Pleased to know that his knowledge and abilities in networking can be of help in a real life scenario.