Lab 10- Backup

Dawn M Inman

CSC-432 Computer and Network Security

Professor Nick Merante

May 8, 2020

Abstract

This lab sets up a backup for the routerVM and the WebServerVM. Options for the rsync program were investigated and studied. The program, rsync, needed installation on both CentOS 7 routers (router and WebServerVM) so that the backups for the labs could be performed. The rsync program is an excellent backup source and cron pairs well with it when scheduling the backup tasks. The cron program was also installed and studied. One issue that prevented the installation of rsync and sshpass onto the WebServerVM was when the wrong external interface name for the router had to be found and corrected. Once working properly, a password was also required to log into the Kali Xfce VM through SSH. This lab uses sshpass to perform this function but other programs like ssh keys or SSH Passwordless authentication could be better for real world use. This was a very successful lab as the backups went well. The importance of backups cannot be emphasized enough as they are vital to keep systems from being down for long periods when problems or disasters arise.

Keywords: CentOS 7, FirewallD, NAT, Kali Xfce, Backup, Proxmox, KVM, QEMU, PuTTY, SSH, Rsync, Cron, SSHPass

Lab 10- Backup

Backing up the information contained on digital devices is necessary to be able to have security in case of anything that could happen to make that device not run or work properly. Examples could be a building fire, unsuccessful update, or a virus that infects a machine, even if it does not hurt anything, it could cause time delays in production because of annoying pop-ups. In either instance, if a good back-up is available, the system can be reset or a new system can be created with the existing recent back-up. This is one of the quickest ways to reset a computer or system back to a good working state. This lab uses a virtual network that contains one virtual router, a web server VM, a Kali Xfce VM and a Metasploitable machine. These are set up in a virtual network with the router facing the external traffic, the firewall in place and the other virtual machines behind the firewall and the router, creating an internal network. The virtual router being used is a CentOS 7 router which works with FirewallD, the virtual computer being run has Kali Xfce and the web server VM uses Ubuntu and Apache. These are managed via Proxmox.

CentOS 7 is so named because it stands for Community ENTerprise Operating System. It is based on the Linux kernel, free and has been available since 2004. Red Hat Enterprise Linux is the origination of CentOS 7 so it is a compatible option when requiring Linux software. It is very popular with almost 30% of Linux web servers using it in 2011 and has been one of the most popular in hosting history. (CentOS Blog, 2020)

FirewallD uses zones and services to manage and control the traffic that goes to and from the system (network). It manages by using trust levels for interfaces and network connections. The zones and services take the place of iptables that were previously used, making it more user friendly. These can be configured to create control to and from flow of traffic, whether it will be allowed or disallowed according to trust level, according to “How to set up a firewall with FirewallD on CentOS7”. (November 11, 2019)

NAT stands for Network Address Translation. It allows an internal network (private network) to have one internet gateway. This gateway is the CentOS 7 router. The machines on the internal network can have different IP addresses inside the network but when going outside of the router it will appear as if there is only one IP address being used. (Bischoff, 2019)

Kali Xfce is a newer Kali release. It is on the same line of Kali environments that have been created for Penetration Testing. There is a new feature called “Kali Undercover” which can make the display of Kali look like Windows 10. This can happen quickly so it is a type of stealth feature meant for blending in when in public areas. (Abrams, 2019) Other new features include KaliNetHunter KeX for Android which can install a full Kali desktop via Android, upgrading the kernel, Git powered documentation and adding PowerShell. (Elwood, 2019)

Proxmox is also being used by the systems but it is not used extensively in this lab. Proxmox VE hypervisor is based on GNU/Linux (Debian) and is open source. It has a central web-based management that does not require more installation. (Cheng, 2014) Version 5.4 is built specifically on Debian 9.8 with a “specially modified Linux Kernel 4.15”. (Proxmox, 2019) Proxmox is capable of two types of virtualization: OpenVZ and KVM. OpenVZ needs a patched Linux kernel so Linux guests are the only operating system type that can be created. In OpenVZ, the guests are called containers because they share the same architecture and kernel as the host operating system. (Cheng, 2014) KVM (Kernel-based Virtual Machine) is a modified Linux kernel built with the KVM module so that it can give hardware-assisted virtualization. Virtualization is performed by a software-based emulator (QEMU) which simulates the virtualized environment while KVM only exposes the /dev/kvm interface. (Cheng, 2014) “This converts Linux into a Type 1 (bare-metal) hypervisor.” (What is KVM?, 2020) Then QEMU or the software-based emulator will create the virtual machines on top of KVM. (What is KVM?, 2020) Proxmox VE is relatively simple to start working with but can be very in depth as Simon M.C. Cheng has authored a book called Proxmox High Availability which goes into more detail when setting up a high availability virtual cluster. (Cheng, 2014)

PuTTY is an SSH client for Windows, Mac and Linux. It has a terminal window for access to the server used in this lab, the GNU/Linux server named chewy. (How to use PuTTY on Windows, 2020) SSH is a software package and means Secure Shell. It secures system administration and file transfers even though the networks are insecure. Tatu Ylonen is the inventor of SSH and OpenSSH which is an open source SSH program is based off his free versions. (SSH(Secure Shell), 2020) Sshpass is a password authentication tool used with SSH allowing the password to be input to the command line. Sshpass is not a secure feature so care needs to be taken when using it because the password for the backup is listed inside its’ command. SSH keys or SSH Passwordless authentication are recommended, instead, for long term backup scripts.

Rsync is an opensource file copying tool. It can work remotely through SSH or locally. It is versatile as there are numerous options controlling all aspects of the copying specifications. It works quickly because it uses a “quick check” algorithm looking for files to be transferred by looking for changes in file sizes and modification times. The delta-transfer algorithm is used find the differences between source files and the destinations existing files, then sends only the difference over the network making the data sent as small as possible and making for faster copying/updating times. (Davidson, Mackerras and Tridgell, 2018)

Cron is an opensource scheduler than can start programs and run them at a specific time. This can be done once or on a schedule like by the year, month, week, day, hour, or minute. Job reports can be set up to be sent to an administrator email. Cron is used to schedule backups, delete old files, running system updates, system maintenance or to monitor disk space. Listing out the current cron jobs can be accomplished at the command line with:

```
crontab -l
```

To edit these jobs type in:

```
crontab -e
```

To run a job each server reboot use:

```
@reboot <command>
```

Remove all jobs with:

```
crontab -r
```

Any other information needed can be found typing into the command line:

```
man crontab
```

The schedule for cron is kept inside of a text editor file and can be added to and subtracted from using the commands above, some which open the text editor. At the beginning of the cron job to be made, the time needs to be input. Every minute looks like

```
* * * * * <command to be run>
```

These stars change and have a variety of possibilities so using the man page for accurate results is essential.

Objective

This labs purpose is to study the backup capabilities of rsync and cron beginning with making backups of the router VM using a local back-up. The next task is adding a web server back-up of the directory /var/www/ in the WebServerVM to a remote machine, in this instance the KaliVM, using the SSH feature and a bash script. Lastly, a cron job is set up that sets up a nightly back-up using the same script as for the WebServer VM back up.

The computer that is being used is a 2011 HP Pavillion dv7, i7 quad core processor and 16GB RAM with Windows10Pro operating system. Google Chrome is the internet browser being used for connecting to ProxMox including the Router and VM consoles. The Kali VM runs Fire Fox internet browser. Kali VM can also run Putty for SSH or the terminal.

Results and Analysis[1]

The following command lines needs analysis:

```
rsync -avz –delete /home/  /var/backups/home
rsync -avz –delete /home/ 192.168.1.10:/var/backups/home
```

The first line will back-up to the local device, copying the directory /home/ and making a backup copy of it inside of the /var/backups/ directory. The second makes a copy of the local /home/ directory and sends it to the device at 192.168.1.10 in the /var/backups/home file. According to the man page for rsync, -avz means to archive a verbose, compressed version. The option of delete means for the program to delete extraneous files from destination directories. To be able to try out more flags, a backup directory was created on the router VM.

First, rsync is install using:

```
yum install rsync
```

Then the man pages can be obtained at

```
man rsync.
```

The backup directory should be created next. If the directory does not exist the backup will fail as it does not know where to put the backup, giving a failed message.

A backup directory can be created using:

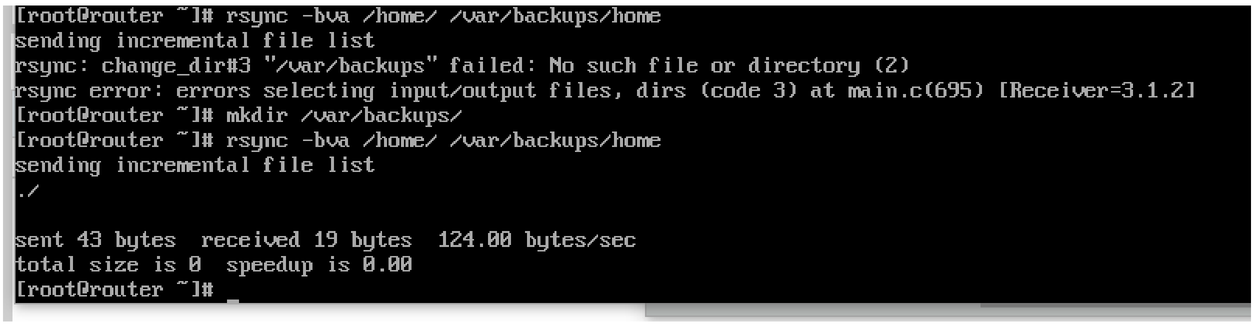
mkdir /var/backups/

A backup will be run with rsync in the home file with:

rsync -bva /home/ /var/backups/home/

This will make backups that are verbose and use archive mode. There will not be any hard links, ACL's or extended attributes in the backup, according to archive mode. It backs up from the /home/ directory to the /var/backups/home directory. The slash behind the word home makes sure that the copy is only of the files inside the home directory, not the directory itself. Other options could be

rsync -bva /home /var/backups/ and this would give the same result if the home directory is not already made, otherwise it may be best to use the first option so extra directory files are not created.



```
[root@router ~]# rsync -bva /home/ /var/backups/home
sending incremental file list
rsync: change_dir#3 "/var/backups" failed: No such file or directory (2)
rsync error: errors selecting input/output files, dirs (code 3) at main.c(695) [Receiver=3.1.2]
[root@router ~]# mkdir /var/backups/
[root@router ~]# rsync -bva /home/ /var/backups/home
sending incremental file list
./

sent 43 bytes  received 19 bytes  124.00 bytes/sec
total size is 0  speedup is 0.00
[root@router ~]# _
```

Figure 1 Backup from /home/ to /var/backups/home

Another set of backup options that were shown in the man pages of rsync is:

rsync -av -e "ssh -l ssh-root" rsync-root@router::module /var/backups/home

This script uses the rsync daemon from a remote SSH connection. The -av means to archive in verbose mode, -e specifies the remote shell to use. This command line specifies ssh -l and ssh-root for changing the remote user at log in. The part, rsync-root@router::module

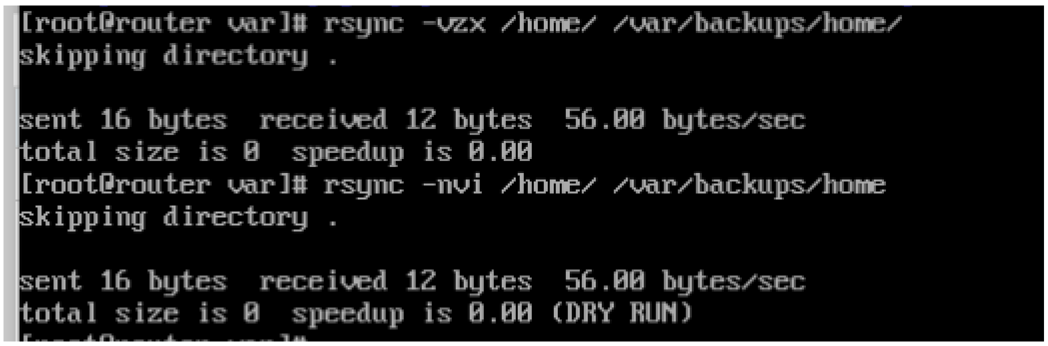
/var/backups/home is showing the owner where the files will be stored, the module and the backup location.

rsync -vzx /home/ /var/backups/home/

The option z compresses the backup into a smaller space and the x tells rsync to avoid crossing a filesystem boundary when recursing.

rsync -nvi /home /var/backups/home/

The options on this line will give a dry run of a back-up that does not make any changes. The verbose option will give the most output and the i option tells the command to itemize the changes that will be made so the changes that rsync is going to make can be viewed before performing the action.

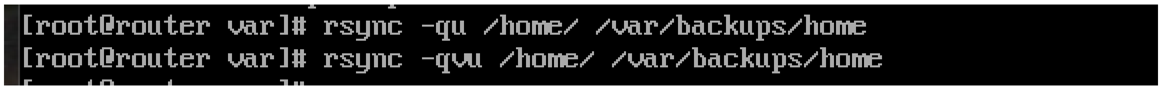


```
[root@router var]# rsync -vzx /home/ /var/backups/home/
skipping directory .

sent 16 bytes  received 12 bytes  56.00 bytes/sec
total size is 0  speedup is 0.00
[root@router var]# rsync -nvi /home/ /var/backups/home
skipping directory .

sent 16 bytes  received 12 bytes  56.00 bytes/sec
total size is 0  speedup is 0.00 (DRY RUN)
```

Figure 2rsync options -vzx and -nvi

rsync -qu /home/ /var/backups/home

This set of options means quiet update. The quiet mode will suppress non-error messages and the update will skip files that are newer on the receiver. The quiet mode means there is no output from the return unless there is an error message. Otherwise, it just returns to the next line like this:



```
[root@router var]# rsync -qu /home/ /var/backups/home
[root@router var]# rsync -qvu /home/ /var/backups/home
```

Figure 3 rsync in quiet mode

rsync -aH /home/ /var/backups/home

The option -a is for archive and H is to preserve hard links in the source or target directories. This command is used on the contents of two directories to sync them so that at the end the target directory is identical to the source.

The next portion of the lab creates a basic backup script that copies the contents of the /var/www/ directory on the WebServerVM. It needs to remotely perform the backup via SSH from the WebServerVM to be backed up to the Kali Xfce machine. The bash scripting language is used for the task. A copy of the script is below.

```
---------------------------------------------------------------------------------------------------
#!/bin/bash
# Author: Dawn Inman
# Date Written: 5/7/2020
# Lab 10: Backups
# Purpose: This script is used to copy the contents of the WebServerVM /var/www/
# directory to the /var/backups/www on the Kali Xfce machine. This is done from the
# web server.
sshpass -p “toor” rsync -bva -e ‘ssh -p22’ /var/www/ root@192.168.11.10:/var/backups/www
---------------------------------------------------------------------------------------------------
```

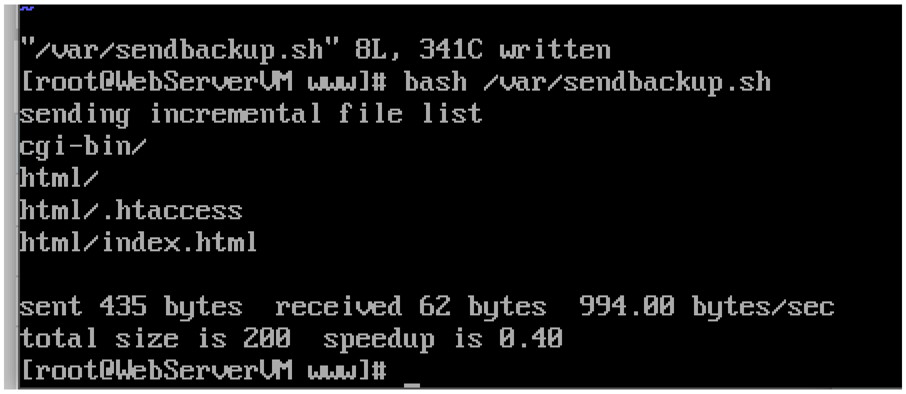
When trying out the commands to make sure the script worked, there were issues as rsync had not been installed on the WebServerVM. When trying to load the updates and rsync, there was an error in the mirrors. This was due to an incorrect setting on the external router. The name of the internet adapter was wrong so the internet could not be accessed properly from inside the network. Once the router was corrected, the WebServerVM began performing properly and installation of the needed programs was easy to complete.

```
yum install update
yum install rsync                    (when prompted):y
yum install sshpass                  (when prompted): y
yum install cronie
```

The above script was written into the file “/var/sendbackup.sh”. To run the script type into the command line:

```
bash /var/sendbackup.sh
```

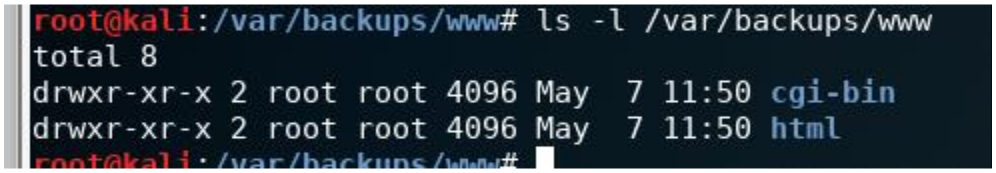
This script uses sshpass to take care of the password prompt so that all that is needed is to run the script and the backup begins and finishes. It was interesting to note that the -a was taken off from the options and the backup worked but did not send anything to the KaliVM as the backup had zero bytes. Once the -a (archive) option was added back as originally intended, the backup worked perfectly. From the WebServerVM:



```
"/var/sendbackup.sh" 8L, 341C written
[root@WebServerVM www]# bash /var/sendbackup.sh
sending incremental file list
cgi-bin/
html/
html/.htaccess
html/index.html

sent 435 bytes  received 62 bytes  994.00 bytes/sec
total size is 200  speedup is 0.40
[root@WebServerVM www]# _
```

Figure 4 Successful backup from WebServerVM

And from KaliXfceVM:



```
root@kali:/var/backups/www# ls -l /var/backups/www
total 8
drwxr-xr-x 2 root root 4096 May  7 11:50 cgi-bin
drwxr-xr-x 2 root root 4096 May  7 11:50 html
root@kali:/var/backups/www#
```

Figure 5 Successful backup from KaliVM

Now that the backup can be run, a nightly schedule for it will be created with cron. For purposes of making sure the backup is working, the schedule will be set to every minute at first, then reset to once daily at midnight. This is done by changing the cron timing:

* * * * * means schedule the job every minute in cron.

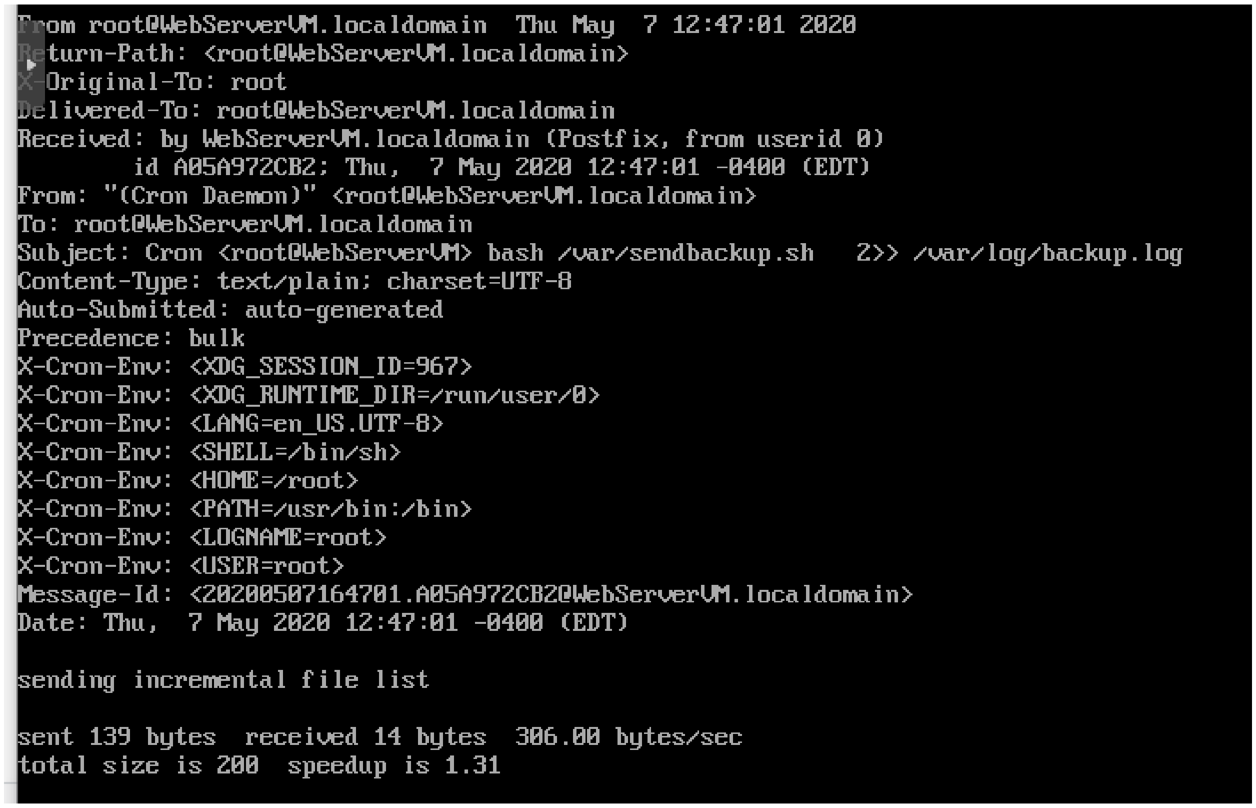
0 0 * * * means schedule the job once per day at midnight in cron.

To schedule the job, open crontab using the -e option:

```
crontab -e
```

This opens a text file for cron that could have other jobs scheduled into it. If so, just add the job at the end of the list of scheduled jobs and it will perform as it is programmed.

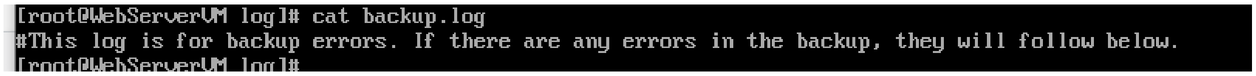
This web server has never had jobs scheduled into it before so a short heading behind a # (comment) is nice to make sure the person working in the file knows for sure it is the crontab file. The cron job goes beneath any comments and begins with the timing, followed by the job and any redirection of errors. For this job, any errors will redirect to a different log file called /var/log/backup.log. Jobs that do not contain errors go to the standard out file which is /var/spool/mail/root. Here is a job that completed without errors in the /var/spool/mail/root location:



```
From root@WebServerVM.localdomain  Thu May  7 12:47:01 2020
Return-Path: <root@WebServerVM.localdomain>
X-Original-To: root
Delivered-To: root@WebServerVM.localdomain
Received: by WebServerVM.localdomain (Postfix, from userid 0)
        id A05A972CB2; Thu,  7 May 2020 12:47:01 -0400 (EDT)
From: "(Cron Daemon)" <root@WebServerVM.localdomain>
To: root@WebServerVM.localdomain
Subject: Cron <root@WebServerVM> bash /var/sendbackup.sh   2>> /var/log/backup.log
Content-Type: text/plain; charset=UTF-8
Auto-Submitted: auto-generated
Precedence: bulk
X-Cron-Env: <XDG_SESSION_ID=967>
X-Cron-Env: <XDG_RUNTIME_DIR=/run/user/0>
X-Cron-Env: <LANG=en_US.UTF-8>
X-Cron-Env: <SHELL=/bin/sh>
X-Cron-Env: <HOME=/root>
X-Cron-Env: <PATH=/usr/bin:/bin>
X-Cron-Env: <LOGNAME=root>
X-Cron-Env: <USER=root>
Message-Id: <20200507164701.A05A972CB2@WebServerVM.localdomain>
Date: Thu,  7 May 2020 12:47:01 -0400 (EDT)

sending incremental file list

sent 139 bytes  received 14 bytes  306.00 bytes/sec
total size is 200  speedup is 1.31
```

Figure 6 cron job without errors

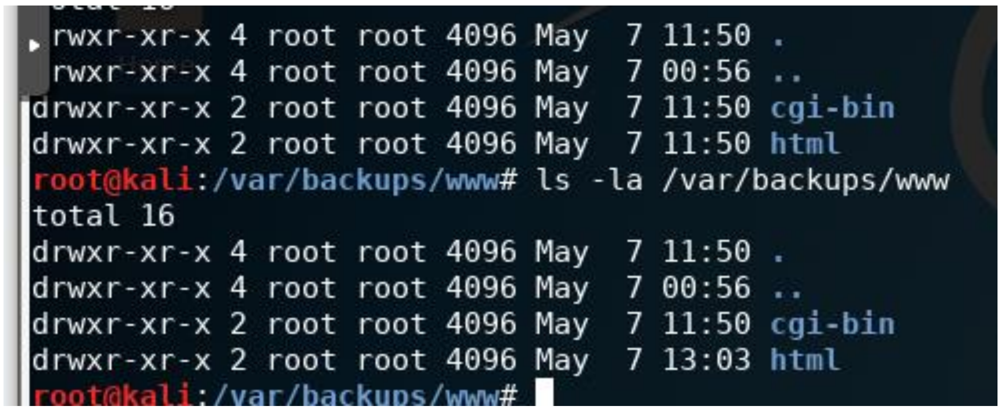
The error log at /var/log/back.ups does not have any errors in it as of yet.



```
[root@WebServerVM log]# cat backup.log
#This log is for backup errors. If there are any errors in the backup, they will follow below.
[root@WebServerVM log]#
```

Figure 7 empty error log for cron job

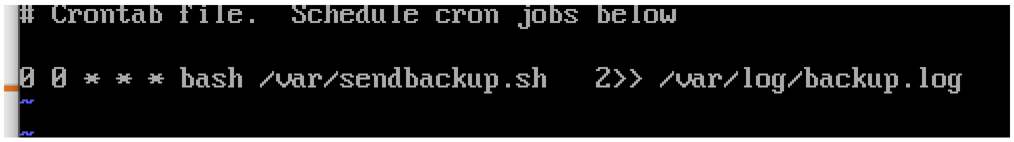
A change was made to the /var/www/html/index.html file to show that the backups are running successfully on KaliXfceVM. The backup is noted at time 13:03.



```
 rwxr-xr-x 4 root root 4096 May  7 11:50 .
 rwxr-xr-x 4 root root 4096 May  7 00:56 ..
drwxr-xr-x 2 root root 4096 May  7 11:50 cgi-bin
drwxr-xr-x 2 root root 4096 May  7 11:50 html
root@kali:/var/backups/www# ls -la /var/backups/www
total 16
drwxr-xr-x 4 root root 4096 May  7 11:50 .
drwxr-xr-x 4 root root 4096 May  7 00:56 ..
drwxr-xr-x 2 root root 4096 May  7 11:50 cgi-bin
drwxr-xr-x 2 root root 4096 May  7 13:03 html
root@kali:/var/backups/www#
```

Figure 8 change in backup for Kali to show the backups are performing successfully

Once all is working properly, the timing is changed so that the backups run every night at midnight. This is done by replacing the first two * with zeros (0). Below is the final finished crontab -e entry for midnight backups with the error logs going to a separate file called /var/log/backup.log. The double arrows will append the file when errors come in instead of overwriting what is on the file so that all error messages will be seen. Any errors already managed would need to be manually erased.



```
# Crontab file.  Schedule cron jobs below

0 0 * * * bash /var/sendbackup.sh   2>> /var/log/backup.log
~
~
```

Figure 9 finished crontab schedule for midnight with error redirects

Conclusion.

This lab was successful at teaching how to use rysyc to set up a backup on both a router and a web server, including using SSH for remote backups and cron for scheduling the jobs in advance. It also taught how to use scripts to do jobs remotely and with scheduling. The program, rsync, is an exceptionally easy program for backups once the user is familiarized with it. The scheduler, cron, is strait forward and easy to use, as well. The script used in this lab is simple, but other scripts can be made that would address using ssh keys for added security measures and eliminating the need for an open password inside of the backup script. Having the password in the open without being encrypted is a serious security risk so using ssh keys would be the better option when using this process in the real world.

Another issue that had to be overcome was that the web server could not ping a domain name. This was because the router had a typo in the interface name. This caused many issues but once it was found, it was effectively changed using Firewalld. The external interface was the faulty one so, since the internal interface is the default, every time anything was added or subtracted, --interface=external had to be added to the command line, as well as –permanent.

```
Firewall-cmd –remove-interface etho0 –zone=external –permanent
Firewall-cmd –add-interface eth0 –zone=external –permanent
Firewall-cmd –complete reload
```

Although there were a couple glitches, this was essentially the main route needed to be taken to fix the issue on the router. Once that was done, all updates were available so that rsync and sshpass could be loaded in.

Overall, this was an excellent lab and taught the student a lot about using backups and schedulers for many different types of jobs and uses. This will be useful in a future career, teaching younger students as well as personally on computers on the students' private network. Special thanks to Professor Merante for finding the router interface issue so progress on the lab could continue.

Lab Network Topology

```
        Kali2020VM--VMWare ethernet adapter--Student HP----(internal –router—external)
        192.168.22.136        192.168.22.1          10.0.0.17     10.0.0.1        192.168.104.161
                                                                                      /
                                                                              /
                                              WWW
                                               |
                                        Gateway 10.42.0.1
                                               |
                                       chewy   10.42.0.31/16
                                               |
                                       darknet  172.16.0.3/16
                                               |
                                      External 172.16.242.32
                                             router
Metasploitable VM  192.168.11.111 -----    Internal  192.168.11.1
                                         /              \
                   Kali VM   192.168.11.10            WebServerVM   192.168.11.15
```

References

Abrams, Lawrence. (November 29, 2019). Kali Linux adds 'undercover' mode to impersonate Windows 10. BleepingComputer. Retrieved from https://www.bleepingcomputer.com/news/security/kali-linux-adds-undercover-mode-to-impersonate-windows-10/

Bischoff, Paul. (March 28, 2019). What is a NAT firewall and how does it work? Comparitech. Retrieved from https://www.comparitech.com/blog/vpn-privacy/nat-firewall/

Centos Blog (2020). What is Centos? Retrieved from https://www.centosblog.com/what-is-centos/

Cheng, Simon M.C. (October 27, 2014). Basic concept of ProxMox Virtual Environment. Packt. Retrieved from https://hub.packtpub.com/basic-concepts-proxmox-virtual-environment/

Davidson, Wayne., Mackerras, Paul. & Tridgell, Andrew. (January 28, 2018). rsync. Retrieved from https://download.samba.org/pub/rsync/rsync.html

Elwood. (November 26, 2019). Kali Linux 2019.4 release. Kali Linux News. Retrieved from https://www.kali.org/news/kali-linux-2019-4-release/

How to set up a firewall with FirewallD on CentOS7. (November 11, 2019). Linuxize. Retrieved from https://linuxize.com/post/how-to-setup-a-firewall-with-firewalld-on-centos-7/

ProxMox. (April 11, 2019). ProxMox.com Retrieved from https://www.proxmox.com/en/news/press-releases/proxmox-ve-5-4

RobetRSeattle. (March 13, 2017). Start ssh automatically on boot. AskUbuntu. Retrieved from https://askubuntu.com/questions/892447/start-ssh-automatically-on-boot

SSH(Secure Shell). (2020). SSH.com. Retrieved from https://www.ssh.com/ssh