Report

Act I: The University Server

Date: 14/03/2021

Author: Anas Kwefati
Email: ak223wd@student.lnu.se
Semester: Spring 2021
Area: Digital Forensics
Course code: 2DV704

Contents

1 Executive Summary

The University has contacted us for an investigation that would confirm if the professor Bob lab's server was infected with a worm or not. This issue has been notified, and suspected by the Network Operations Center (NOC), saying that his lab's server was infected with a worm, due to a huge spike in Internet traffic at 4 in the morning. This professor seems to not think it was infected, however, because the University was eager to obtain an independent confirmation, they had to contact us. According to the professor, there were no many files except on his own account called "bob". Other accounts might have files, such as "eric", "kevin", "peter", and "takeda".

2 Purpose of the Investigation

The reason for conducting the investigation can be read in Section 1. But to sum it up, the university has contacted us, due to a notification from NOC saying that a university server has been compromised due to a huge spike in Internet traffic at 4 a.m. Hence, in that case, we will probably check, if there are any malware in the hard disk, then verify the logs section of the image to see if anything happened. We have managed to obtain the server image, and we are investing it using Kali Linux 2020.3 in a Virtual Machine environment.

3 Methodology

In this Forensic examination, we have decided to download **act1.img**, in our local computer using the command tool rsync. Then, we mounted that image into our Kali Linux VM using VirtualBox. We decided to use Kali Linux as it regroups many tools that can be used for digital forensics. Before actually mounting the image, we made sure to keep a copy of the image, and create a hash value of that image, in order to make sure it is the same (Figure 1). When accessing the image, we first check the system logs, then each user bash history.

4 Electronic Media Analyzed

The file **act1.img** has been examined. An .img extension means that the file is storing raw disk images of either a floppy disks, hard drives, and optical discs. Therefore, there should be no data beyond what the content of the disk. The investigator has obtained the file on March 11, 2021 in the morning. The total size of the file is 2,15Gb.

5 Report Findings

An analysis was conducted on the given image file of the hard disk. This hard disk represents one of the university lab's server, furthermore, the given file was imaged by the professor Bob.

We received this image on March 11 2021 at 11:00 AM, named as **act1.img**. After receiving this image, we decided to create a copy of it and generate a cryptographic hash of the copy and the original file. Doing that allows us to ensure that the two images are bit-for-bit the same, and represent exactly what is on the file. This result can be seen in Figure 1.

Now, we know for certain that the copy image has not been changed, therefore, all future work will be done on the copy image file.

The first thing we had to do, is to verify the first hypothesis of the Network Operations Center, and check if the hard disk was indeed infected with a malware. Therefore, we used the anti-virus **ClamAV**. We scanned the file, and Figure 2 shows that no malware were detected.

The second step was to check the passwd file. This file will help to make sure that no suspicious account was created, and only the given users by the professor have an account. The Figure 3, shows the result, it is located in **/etc/passwd**. We can confirm that no unwanted user has been found. While at it, we also checked the group information, **/etc/group**, Figure 4 shows that the student Takeda has admin privilege, which seems to be quite suspicious that a student has admin level.

From that point, we decide to look further in the system, so the first thing we want to know is the logs. Thus, we go check **/var/log** location, which is where the log files are typically stored in a UNIX system like Linux. We firstly examined the file that contains system operations (**var/log/syslog**), as seen in Figure 9, most of the data sound like normal logs. We notice the last few logs the shutdown of the server happening. This confirms on what the professor Bob has said. And at the top of the file, we can see that a user called **"kevin"** was using the cron service on January 4 at 08:56:07, he seems to have listed the crontab and replaced something. The basic usage of cron is to execute a job at a specific time. But this information alone is not enough to suspect anything. So, we decide to examine the authentication logs (**var/log/auth.log**), there were normal logs, but we notice that the user Kevin connected with SSH to the server and it happened somewhat at the same time as the crontab list and replace in the syslog. In Figure 10 we filtered with only cron and kevin in the auth.log file to see if there would be some useful information. Afterwards, we decide to check bash history of each users.

- **Bob:** He was the first one, and Figure 5, shows the only command he has entered. This history confirms on what he said previously.
- **Eric:** As seen in Figure 6, he has entered only 2 commands **mutt** and **logout**. The former command is used to send and read email from command line. Whereas the latter is used to logout from the system. This history does not give us much context, and on what is happening, but it is still something that could be suspect. Especially the use of the command mutt, should not be allowed for a student on university's server, if this one did not receive an authorization.
- **Kevin:** Figure 8, we can see the entered command. The first thing we notice is again the use of **crontab -l**, which lists all current cronjob. Just after it, we can see the use of an rsync with a cronjob command. The latter command, helps to schedule and execute tasks at a fixed period of time. On the other hand, the former command, is a network-enabled syncing tool, so it can pretty much synchronize files and folders from one location to another. Moreover, we can see that he has **0 4 * * ***, and according to cron time string formatting, this means, that it will execute a

command at 4 a.m. every day, of every week, of every month. Therefore, we might think it is him who created that spike. Furthermore, we can see other folders such as Music and Links which seem to contain music files.

- **Peter:** He did not seem to have any command history as the **.bash_history** was not found.
- **Takeda:** In Figure 7, does seem to have entered many commands such as, fgrep which is used to search for a fixed character strings. This person was trying to obtain account information from /etc/passwd file. There is also an ifconfig command which helps to check network interface configuration such as IP address. Then, we can also see that there is another folder called **eggdrop**, after looking online, it seems to be like an IRC bot. We do not think, it is a harmful program, but still it is probably not allowed to install and configure such program on university's server.

As said previously, the basic usage of cron is to execute a job at a specific time. Thus, when looking to the command history, Figure 8, we are suspecting that the user **"kevin"**, has run such command with rsync to transfer music from his PC to the university server. In this rsync command, he has specified many options. For instance, -q is used to suppress errors, and he also specified the use of ssh.

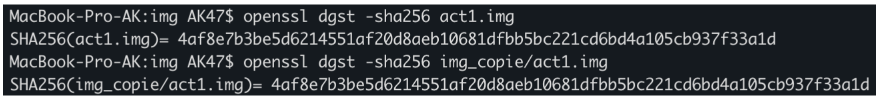
"rsync -aq –del –rsh="ssh" -e "ssh -l kevin" "kevin.dynip.com:My_Music/" " /music"

To continue, he seems to be transferring files from his own PC, **kevin.dynip.com:My_Music/** to **/music** which is his directory on his account in this server. Also, after checking on the Internet for the name **dynip.com**, it seems that it is a service that allows to register a personalized name that can be used to connect to your computer online. Then, when looking at the music directory, we can see in Figure 12, that it contains 280 Megabyte (280M) of music. This sounds quite a lot for this server, as it is supposed to be nearly empty with no files in it. But, before concluding anything, we need to inspect the file where the user's cron job has been defined. This file, is located in **/var/spool/cron/crontabs**. Figure 11 shows clearly that the user **"kevin"**, is the one who has created a crontab file and defined the cronjob. Opening this file shows the command used by Kevin.

6 Conclusion

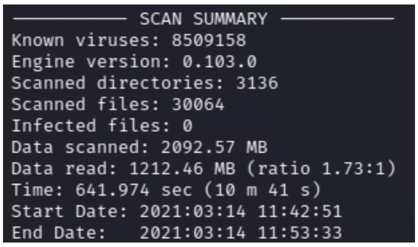
To conclude, according to the data, we believe that the server was not compromised, but happened because of a user, named as **"kevin"**, who has transferred data from one PC to another using rsync and cron service at 4 in the morning. Indeed, he has added a cronjob to transfer music files to the university server. We also believe that no attacker managed to access sensitive information. Therefore, before returning the system to production, there is a need to delete the cronjob, created by kevin, in order to stop this command to happen again. Then, student accounts should be restricted to only what they need for their lab assignments. For instance, they should not be able to run a chatbot using university's server, uploading music files into the server, using cron service, or even using other commands such as mutt. Many of these commands should be limited to only the system administrator and not normal users. Hence, all accounts that are not admin should be restricted to the maximum. Finally, there should be a sort of monitoring system, in order to know better who is doing what on the system.

7 Exhibits/Appendices



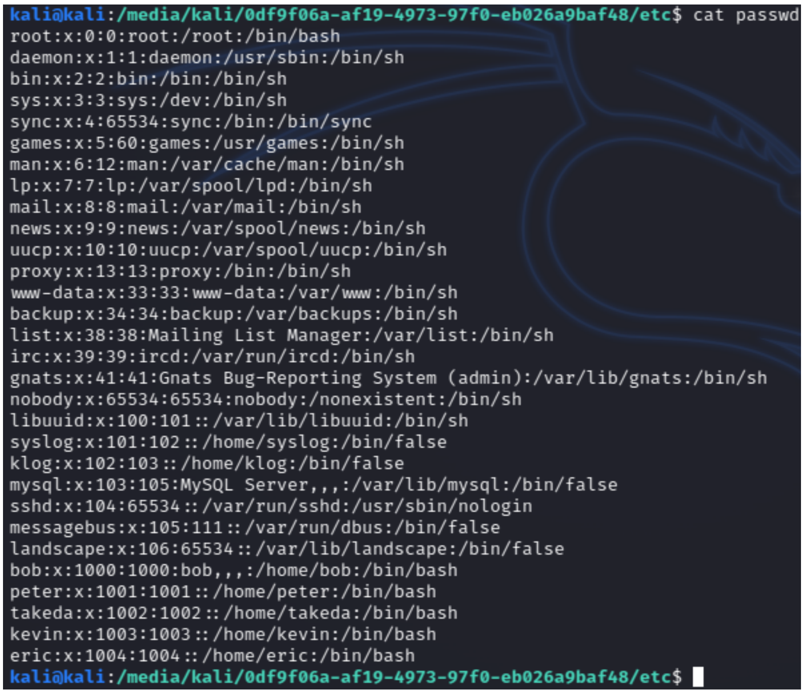
```
MacBook-Pro-AK:img AK47$ openssl dgst -sha256 act1.img
SHA256(act1.img)= 4af8e7b3be5d6214551af20d8aeb10681dfbb5bc221cd6bd4a105cb937f33a1d
MacBook-Pro-AK:img AK47$ openssl dgst -sha256 img_copie/act1.img
SHA256(img_copie/act1.img)= 4af8e7b3be5d6214551af20d8aeb10681dfbb5bc221cd6bd4a105cb937f33a1d
```

Figure 1: Cryptographic hash value of act1.img



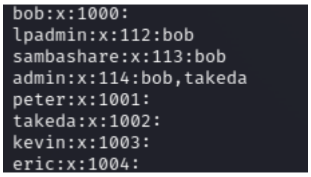
```
----------- SCAN SUMMARY -----------
Known viruses: 8509158
Engine version: 0.103.0
Scanned directories: 3136
Scanned files: 30064
Infected files: 0
Data scanned: 2092.57 MB
Data read: 1212.46 MB (ratio 1.73:1)
Time: 641.974 sec (10 m 41 s)
Start Date: 2021:03:14 11:42:51
End Date:   2021:03:14 11:53:33
```

Figure 2: ClamAV result on the act1.img



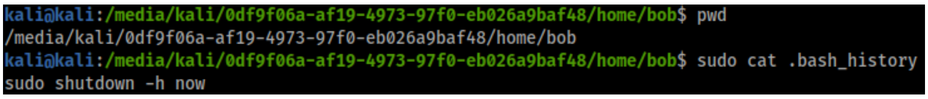
```
kali@kali:/media/kali/0df9f06a-af19-4973-97f0-eb026a9baf48/etc$ cat passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:102::/home/syslog:/bin/false
klog:x:102:103::/home/klog:/bin/false
mysql:x:103:105:MySQL Server,,,:/var/lib/mysql:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
messagebus:x:105:111::/var/run/dbus:/bin/false
landscape:x:106:65534::/var/lib/landscape:/bin/false
bob:x:1000:1000:bob,,,:/home/bob:/bin/bash
peter:x:1001:1001::/home/peter:/bin/bash
takeda:x:1002:1002::/home/takeda:/bin/bash
kevin:x:1003:1003::/home/kevin:/bin/bash
eric:x:1004:1004::/home/eric:/bin/bash
kali@kali:/media/kali/0df9f06a-af19-4973-97f0-eb026a9baf48/etc$
```

Figure 3: Passwd file



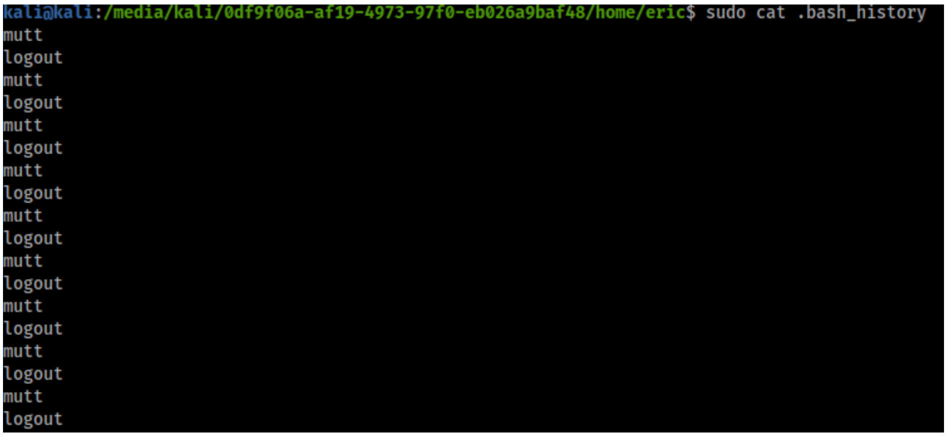
```
bob:x:1000:
lpadmin:x:112:bob
sambashare:x:113:bob
admin:x:114:bob,takeda
peter:x:1001:
takeda:x:1002:
kevin:x:1003:
eric:x:1004:
```

Figure 4: Group file



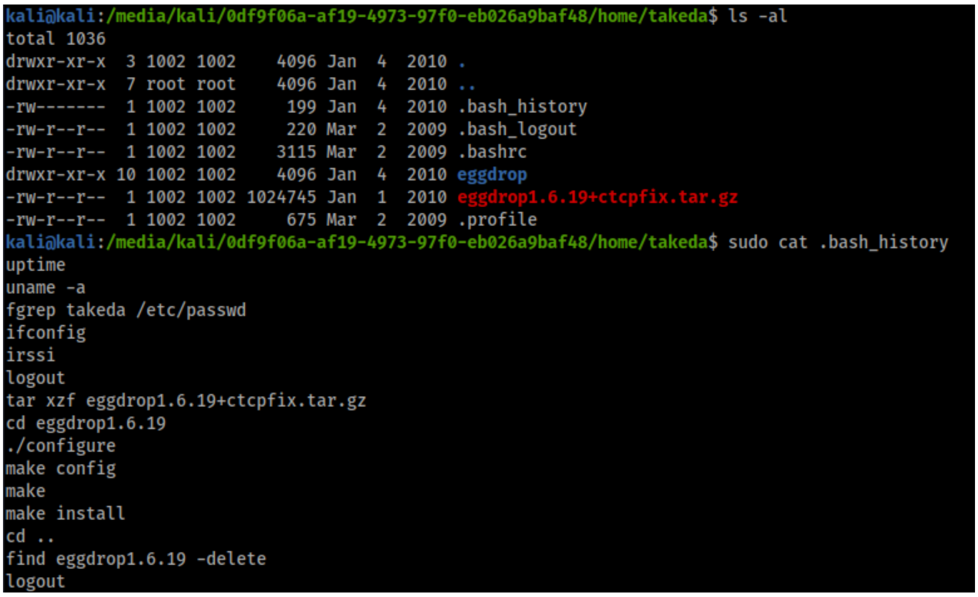
```
kali@kali:/media/kali/0df9f06a-af19-4973-97f0-eb026a9baf48/home/bob$ pwd
/media/kali/0df9f06a-af19-4973-97f0-eb026a9baf48/home/bob
kali@kali:/media/kali/0df9f06a-af19-4973-97f0-eb026a9baf48/home/bob$ sudo cat .bash_history
sudo shutdown -h now
```

Figure 5: Bob's Bash History



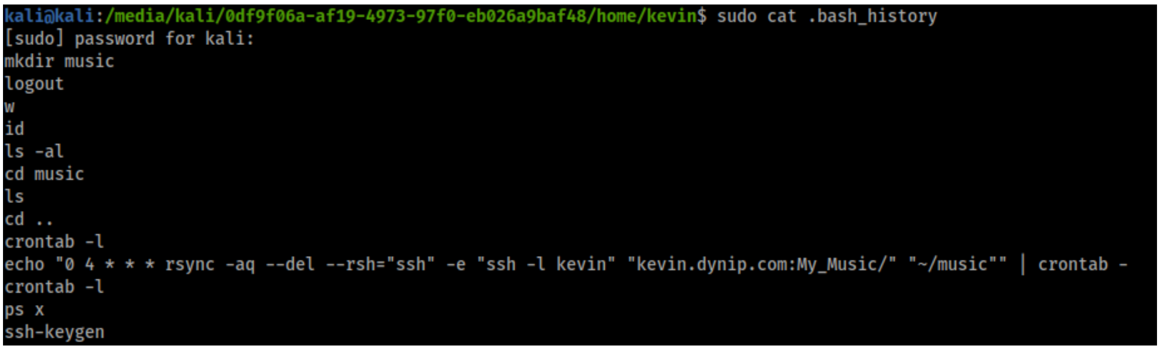
```
kali@kali:/media/kali/0df9f06a-af19-4973-97f0-eb026a9baf48/home/eric$ sudo cat .bash_history
mutt
logout
mutt
logout
mutt
logout
mutt
logout
mutt
logout
mutt
logout
mutt
logout
mutt
logout
mutt
logout
```

Figure 6: Eric's Bash History



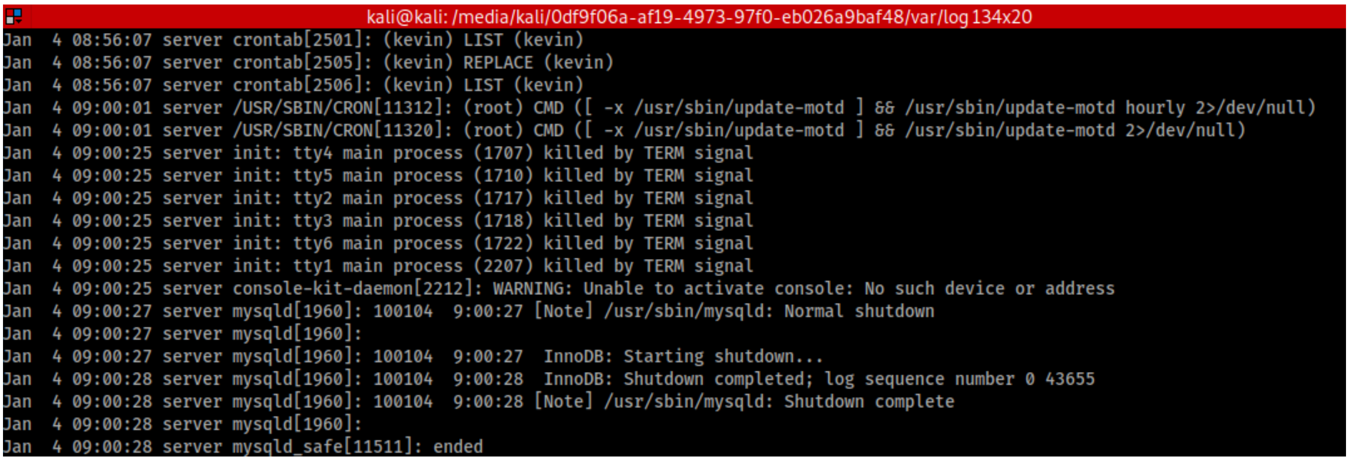
```
kali@kali:/media/kali/0df9f06a-af19-4973-97f0-eb026a9baf48/home/takeda$ ls -al
total 1036
drwxr-xr-x  3 1002 1002    4096 Jan  4  2010 .
drwxr-xr-x  7 root root    4096 Jan  4  2010 ..
-rw-------  1 1002 1002     199 Jan  4  2010 .bash_history
-rw-r--r--  1 1002 1002     220 Mar  2  2009 .bash_logout
-rw-r--r--  1 1002 1002    3115 Mar  2  2009 .bashrc
drwxr-xr-x 10 1002 1002    4096 Jan  4  2010 eggdrop
-rw-r--r--  1 1002 1002 1024745 Jan  1  2010 eggdrop1.6.19+ctcpfix.tar.gz
-rw-r--r--  1 1002 1002     675 Mar  2  2009 .profile
kali@kali:/media/kali/0df9f06a-af19-4973-97f0-eb026a9baf48/home/takeda$ sudo cat .bash_history
uptime
uname -a
fgrep takeda /etc/passwd
ifconfig
irssi
logout
tar xzf eggdrop1.6.19+ctcpfix.tar.gz
cd eggdrop1.6.19
./configure
make config
make
make install
cd ..
find eggdrop1.6.19 -delete
logout
```

Figure 7: Takeda's Bash History + files



```
kali@kali:/media/kali/0df9f06a-af19-4973-97f0-eb026a9baf48/home/kevin$ sudo cat .bash_history
[sudo] password for kali:
mkdir music
logout
w
id
ls -al
cd music
ls
cd ..
crontab -l
echo "0 4 * * * rsync -aq --del --rsh="ssh" -e "ssh -l kevin" "kevin.dynip.com:My_Music/" "~/music"" | crontab -
crontab -l
ps x
ssh-keygen
```

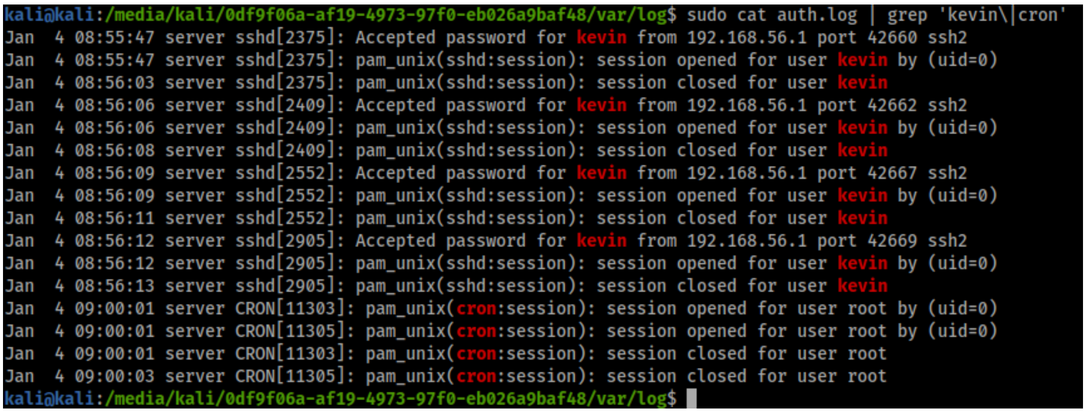
Figure 8: Kevin's Bash History



kali@kali: /media/kali/0df9f06a-af19-4973-97f0-eb026a9baf48/var/log 134x20

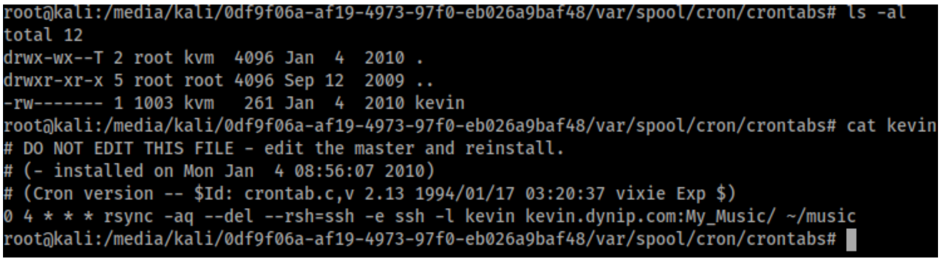
```
Jan  4 08:56:07 server crontab[2501]: (kevin) LIST (kevin)
Jan  4 08:56:07 server crontab[2505]: (kevin) REPLACE (kevin)
Jan  4 08:56:07 server crontab[2506]: (kevin) LIST (kevin)
Jan  4 09:00:01 server /USR/SBIN/CRON[11312]: (root) CMD ([ -x /usr/sbin/update-motd ] && /usr/sbin/update-motd hourly 2>/dev/null)
Jan  4 09:00:01 server /USR/SBIN/CRON[11320]: (root) CMD ([ -x /usr/sbin/update-motd ] && /usr/sbin/update-motd 2>/dev/null)
Jan  4 09:00:25 server init: tty4 main process (1707) killed by TERM signal
Jan  4 09:00:25 server init: tty5 main process (1710) killed by TERM signal
Jan  4 09:00:25 server init: tty2 main process (1717) killed by TERM signal
Jan  4 09:00:25 server init: tty3 main process (1718) killed by TERM signal
Jan  4 09:00:25 server init: tty6 main process (1722) killed by TERM signal
Jan  4 09:00:25 server init: tty1 main process (2207) killed by TERM signal
Jan  4 09:00:25 server console-kit-daemon[2212]: WARNING: Unable to activate console: No such device or address
Jan  4 09:00:27 server mysqld[1960]: 100104  9:00:27 [Note] /usr/sbin/mysqld: Normal shutdown
Jan  4 09:00:27 server mysqld[1960]:
Jan  4 09:00:27 server mysqld[1960]: 100104  9:00:27  InnoDB: Starting shutdown...
Jan  4 09:00:28 server mysqld[1960]: 100104  9:00:28  InnoDB: Shutdown completed; log sequence number 0 43655
Jan  4 09:00:28 server mysqld[1960]: 100104  9:00:28 [Note] /usr/sbin/mysqld: Shutdown complete
Jan  4 09:00:28 server mysqld[1960]:
Jan  4 09:00:28 server mysqld_safe[11511]: ended
```

Figure 9: Syslog file content



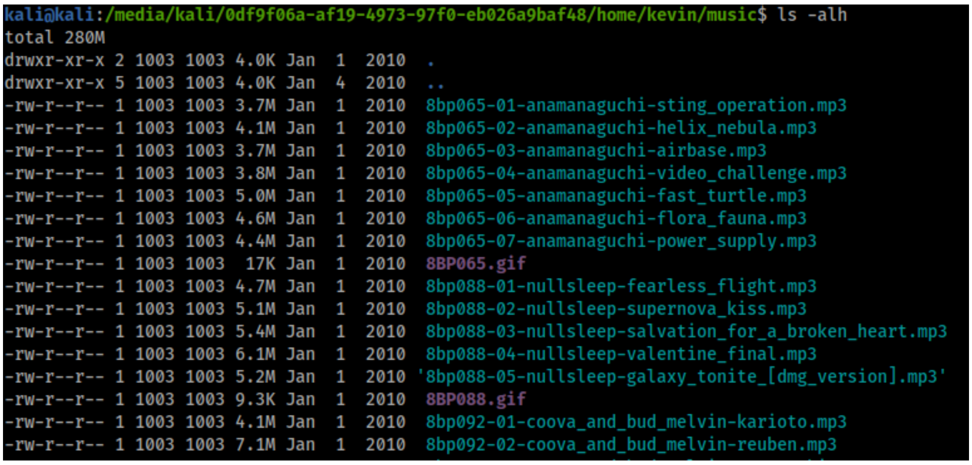
```
kali@kali:/media/kali/0df9f06a-af19-4973-97f0-eb026a9baf48/var/log$ sudo cat auth.log | grep 'kevin\|cron'
Jan  4 08:55:47 server sshd[2375]: Accepted password for kevin from 192.168.56.1 port 42660 ssh2
Jan  4 08:55:47 server sshd[2375]: pam_unix(sshd:session): session opened for user kevin by (uid=0)
Jan  4 08:56:03 server sshd[2375]: pam_unix(sshd:session): session closed for user kevin
Jan  4 08:56:06 server sshd[2409]: Accepted password for kevin from 192.168.56.1 port 42662 ssh2
Jan  4 08:56:06 server sshd[2409]: pam_unix(sshd:session): session opened for user kevin by (uid=0)
Jan  4 08:56:08 server sshd[2409]: pam_unix(sshd:session): session closed for user kevin
Jan  4 08:56:09 server sshd[2552]: Accepted password for kevin from 192.168.56.1 port 42667 ssh2
Jan  4 08:56:09 server sshd[2552]: pam_unix(sshd:session): session opened for user kevin by (uid=0)
Jan  4 08:56:11 server sshd[2552]: pam_unix(sshd:session): session closed for user kevin
Jan  4 08:56:12 server sshd[2905]: Accepted password for kevin from 192.168.56.1 port 42669 ssh2
Jan  4 08:56:12 server sshd[2905]: pam_unix(sshd:session): session opened for user kevin by (uid=0)
Jan  4 08:56:13 server sshd[2905]: pam_unix(sshd:session): session closed for user kevin
Jan  4 09:00:01 server CRON[11303]: pam_unix(cron:session): session opened for user root by (uid=0)
Jan  4 09:00:01 server CRON[11305]: pam_unix(cron:session): session opened for user root by (uid=0)
Jan  4 09:00:01 server CRON[11303]: pam_unix(cron:session): session closed for user root
Jan  4 09:00:03 server CRON[11305]: pam_unix(cron:session): session closed for user root
kali@kali:/media/kali/0df9f06a-af19-4973-97f0-eb026a9baf48/var/log$
```

Figure 10: Auth.log file content filtered with GREP for kevin and cron



```
root@kali:/media/kali/0df9f06a-af19-4973-97f0-eb026a9baf48/var/spool/cron/crontabs# ls -al
total 12
drwx-wx--T 2 root kvm  4096 Jan  4  2010 .
drwxr-xr-x 5 root root 4096 Sep 12  2009 ..
-rw------- 1 1003 kvm   261 Jan  4  2010 kevin
root@kali:/media/kali/0df9f06a-af19-4973-97f0-eb026a9baf48/var/spool/cron/crontabs# cat kevin
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (- installed on Mon Jan  4 08:56:07 2010)
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
0 4 * * * rsync -aq --del --rsh=ssh -e ssh -l kevin kevin.dynip.com:My_Music/ ~/music
root@kali:/media/kali/0df9f06a-af19-4973-97f0-eb026a9baf48/var/spool/cron/crontabs#
```

Figure 11: Crontab folder with cron file named Kevin



```
kali@kali:/media/kali/0df9f06a-af19-4973-97f0-eb026a9baf48/home/kevin/music$ ls -alh
total 280M
drwxr-xr-x 2 1003 1003 4.0K Jan  1  2010  .
drwxr-xr-x 5 1003 1003 4.0K Jan  4  2010  ..
-rw-r--r-- 1 1003 1003 3.7M Jan  1  2010  8bp065-01-anamanaguchi-sting_operation.mp3
-rw-r--r-- 1 1003 1003 4.1M Jan  1  2010  8bp065-02-anamanaguchi-helix_nebula.mp3
-rw-r--r-- 1 1003 1003 3.7M Jan  1  2010  8bp065-03-anamanaguchi-airbase.mp3
-rw-r--r-- 1 1003 1003 3.8M Jan  1  2010  8bp065-04-anamanaguchi-video_challenge.mp3
-rw-r--r-- 1 1003 1003 5.0M Jan  1  2010  8bp065-05-anamanaguchi-fast_turtle.mp3
-rw-r--r-- 1 1003 1003 4.6M Jan  1  2010  8bp065-06-anamanaguchi-flora_fauna.mp3
-rw-r--r-- 1 1003 1003 4.4M Jan  1  2010  8bp065-07-anamanaguchi-power_supply.mp3
-rw-r--r-- 1 1003 1003  17K Jan  1  2010  8BP065.gif
-rw-r--r-- 1 1003 1003 4.7M Jan  1  2010  8bp088-01-nullsleep-fearless_flight.mp3
-rw-r--r-- 1 1003 1003 5.1M Jan  1  2010  8bp088-02-nullsleep-supernova_kiss.mp3
-rw-r--r-- 1 1003 1003 5.4M Jan  1  2010  8bp088-03-nullsleep-salvation_for_a_broken_heart.mp3
-rw-r--r-- 1 1003 1003 6.1M Jan  1  2010  8bp088-04-nullsleep-valentine_final.mp3
-rw-r--r-- 1 1003 1003 5.2M Jan  1  2010 '8bp088-05-nullsleep-galaxy_tonite_[dmg_version].mp3'
-rw-r--r-- 1 1003 1003 9.3K Jan  1  2010  8BP088.gif
-rw-r--r-- 1 1003 1003 4.1M Jan  1  2010  8bp092-01-coova_and_bud_melvin-karioto.mp3
-rw-r--r-- 1 1003 1003 7.1M Jan  1  2010  8bp092-02-coova_and_bud_melvin-reuben.mp3
```

Figure 12: Kevin’s music folder + total folder’s size