Computer Security - CS 6531 – Hacking Exercise
Namana Y Tarikere – G21372717

1. Hacking Exercise (100 pts)
References:
https://nmap.org/book/man.html
https://www.kali.org/tools/hydra/
https://www.gnupg.org/documentation/
https://www.openwall.com/john/

Note: You may be able to bypass the local authentication of the VM via the bootloader. Such an attack will not result in any credit for this assignment.

Download the target VM and import it into VirtualBox or VMWare fusion. Ensure you download the correct target for your hardware: one is for x86 (64-bit) and one is for Apple M1/M2/M3. Place both machines on the same host-only network (as we went over in class). https://gwu.box.com/s/n2fnfflk0becj3d8jc7jxbwz3skhmy7e

Your goal is to obtain access to at least five user accounts on the VM.

- Gaining access to one account is 25 pts total.
- Gaining access to two accounts is 45 pts total.
- Gaining access to three accounts is 70 pts total.
- Gaining access to four accounts is 95 pts total.
- Gaining access to five accounts is 100 pts total.
- Getting access to root is 110 pts total.

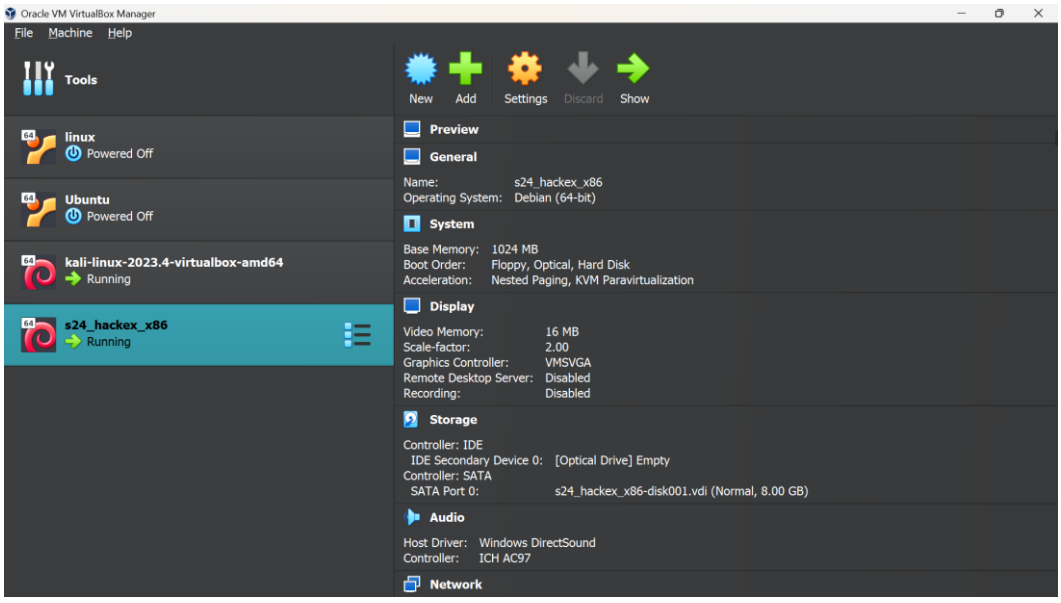
Thoroughly document how you go about gaining access, including your failures. If you write any code, include it in your submission as a separate file. If you use any password-cracking tools, include the logs for your password-cracking attempts. Explain your reasoning behind the actions you take. This assignment is nonlinear and will require you to be both curious and patient. There is more than one way to get full credit, and it is possible to get full credit using techniques we have used together in class. There is only one way to get no credit: *not trying.* **I strongly recommend that you start this early!**

Good luck. Ask questions. Do not give up.

Solution:

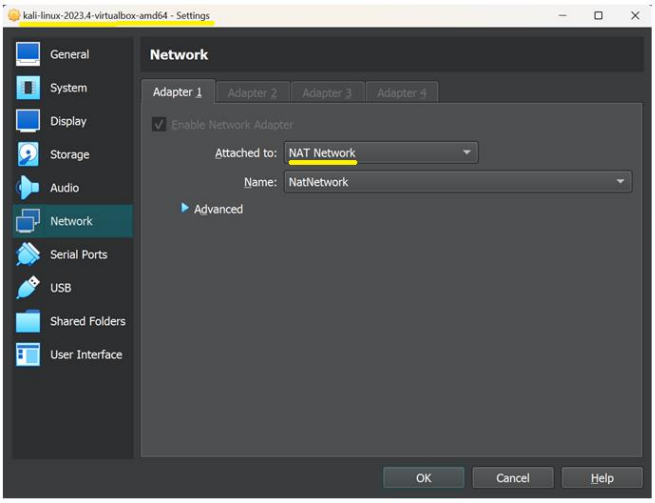
I have a Windows laptop with Oracle VirtualBox installed. For this hacking exercise, I downloaded the required files from the given link, unzipped the contents and installed the given s24_hackex_x86 virtual machine in my Oracle VirtualBox.

I am parallelly running both my Kali Linux (Host VM) and hackex_x86 VM (attacking VM) on Virtual Box as shown below:

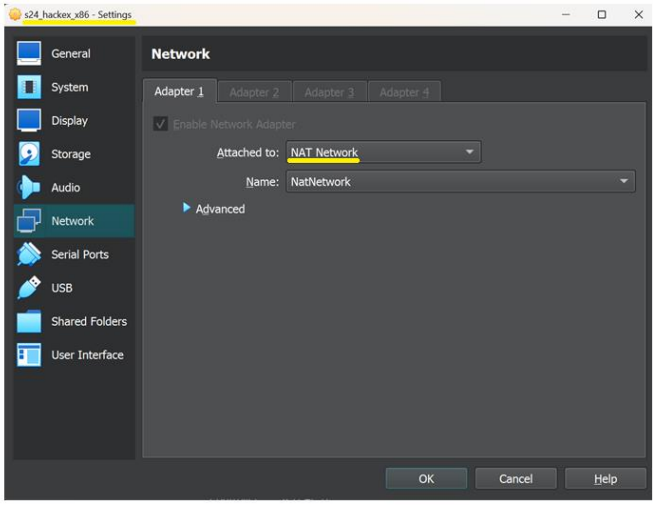


I configured the network settings of both Kali and hackex_x86 VM to NAT Network as shown below:

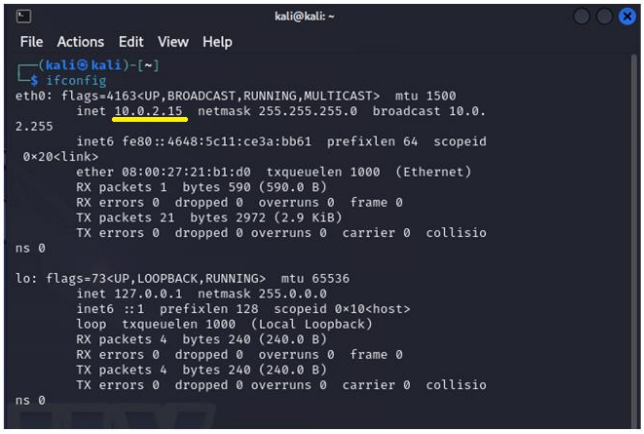
Kali Linux in NAT Network:

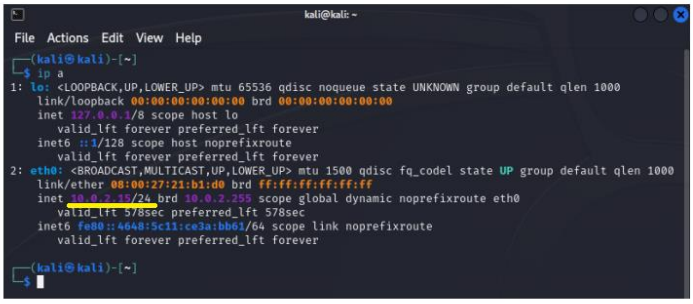


Hackex_x86 VM in NAT Network:

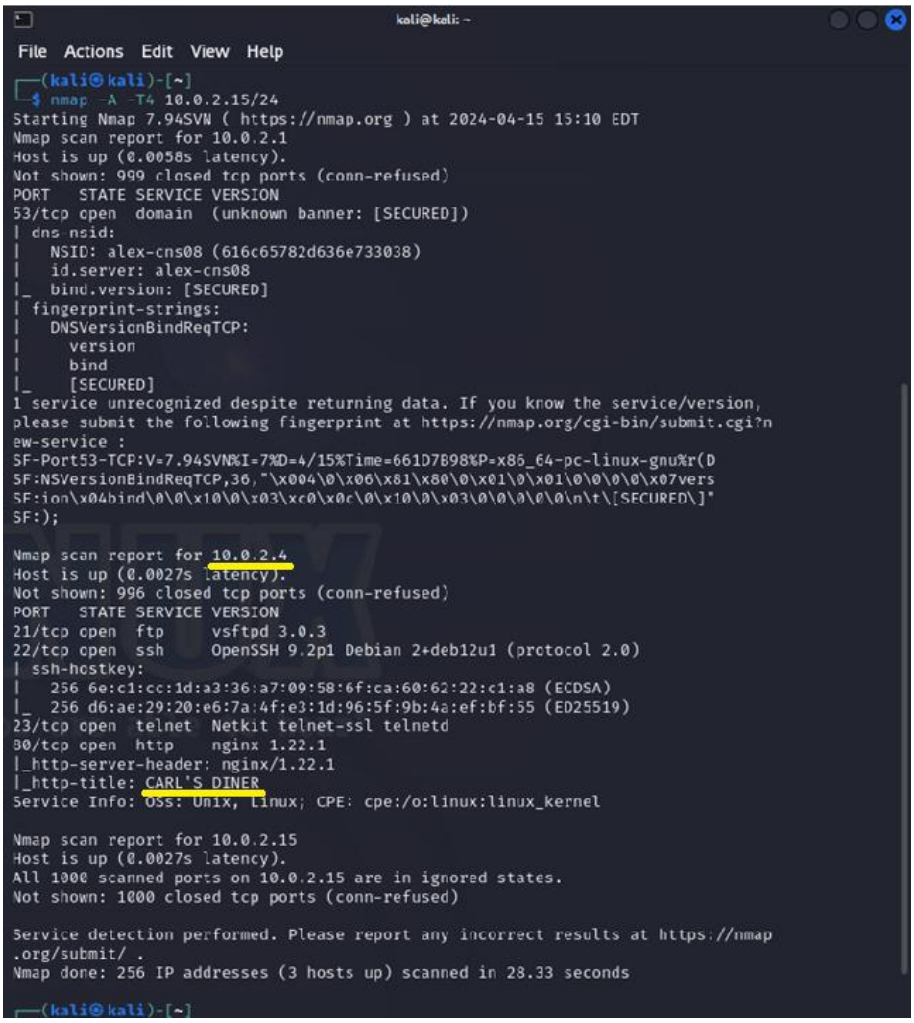


I ran **ifconfig** and **ip a** command [1] to check the IP address of NAT network to be **10.0.2.15** as shown:

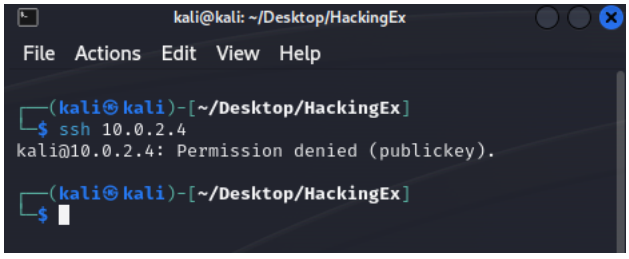




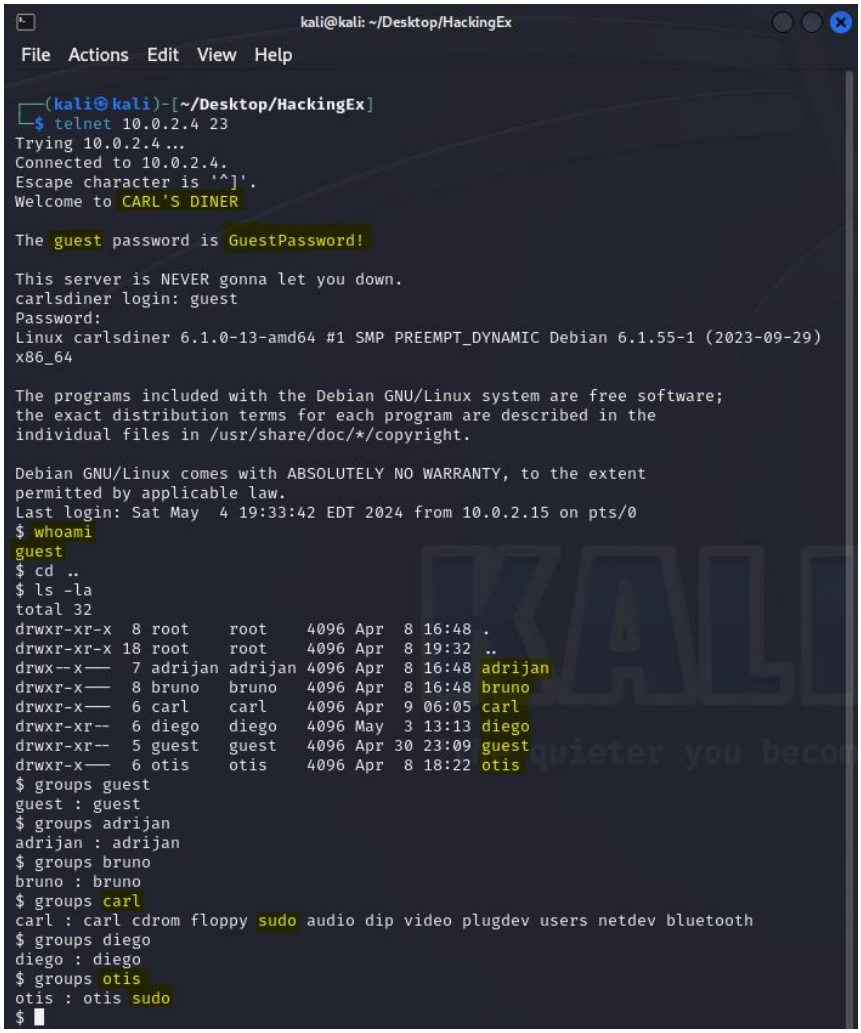
Next, I ran the **nmap -A -T4 10.0.2.15/24** command [1] to check the available networks, open services and ports, and target VM on the fetched IP address. I found the IP address of target VM **Carl's Diner Website** to be **10.0.2.4** as shown below:



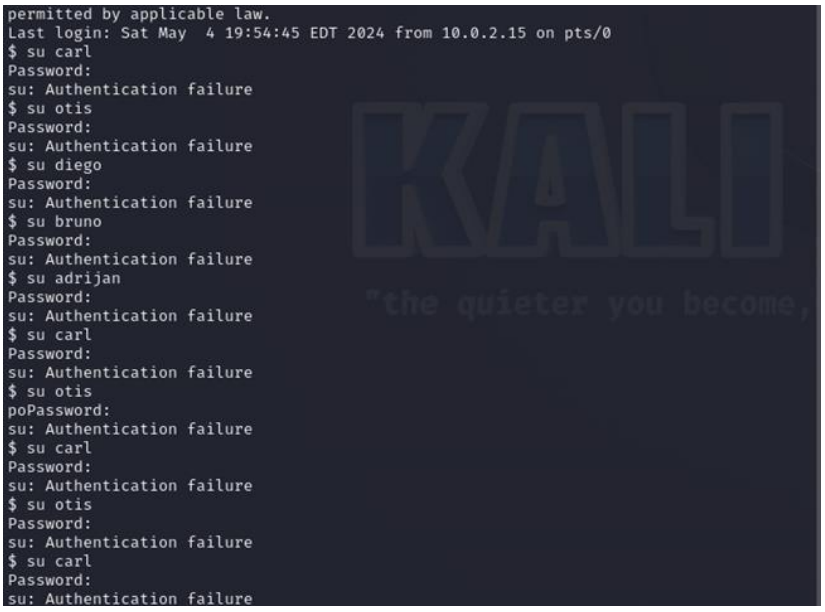
I tried to use **ssh 10.0.2.4** command to connect to the server and failed:



I used **telnet 10.0.2.4 23** command [1] to connect to the server and was able to get the password as **GuestPassword!** for user **guest**. I successfully logged into Carl's diner as guest, found out that there are **5 users Adrijan, Bruno, Carl, Diego, and Otis** and found that users **Carl and Otis have sudo access** permissions as shown:

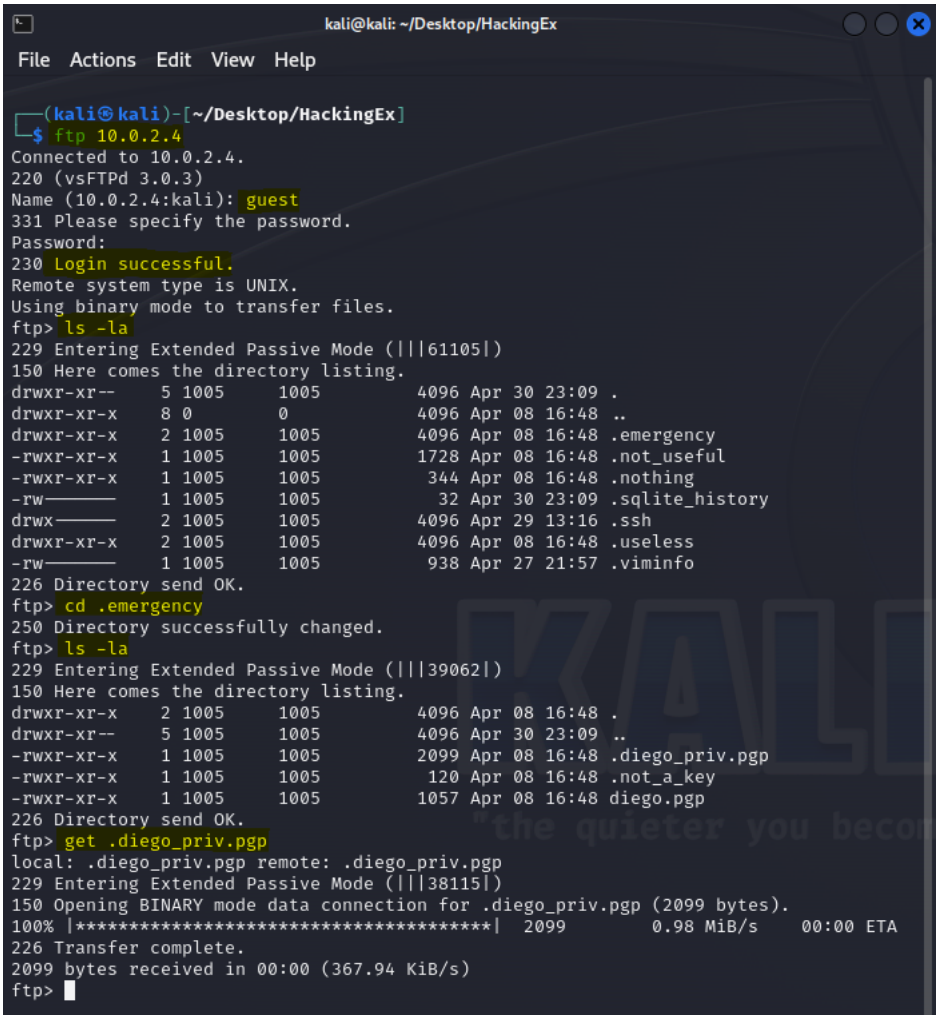


Next, I connected to the server using telnet and tried various passwords from Homework 2 exercise for all users in trial-and-error format. For example, password1, password123, password! for Otis and other users but was not successful.



I used **ftp 10.0.2.4** command [1] to connect to the server, logged in as guest and checked the contents of the account. I was able to get **Diego's private key** in the **.emergency** hidden file. I followed the following commands to fetch and store Diego's private key in host kali Linux as shown below:

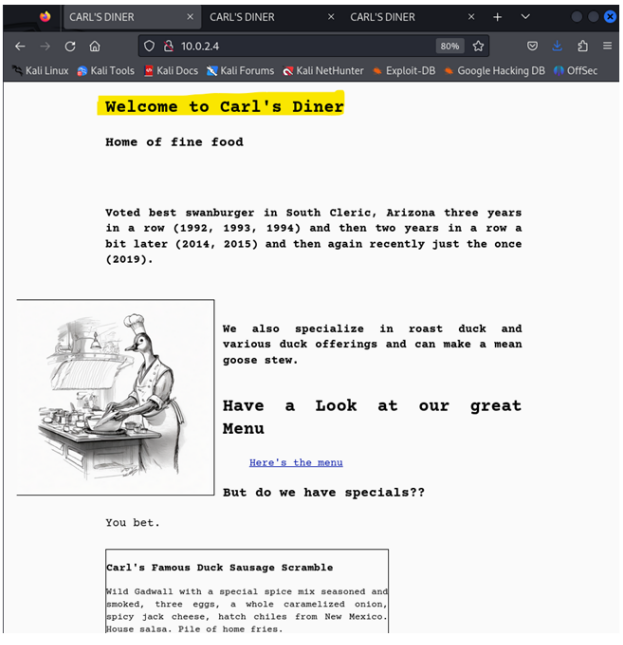
1. ftp 10.0.2.4
2. logged in as guest
3. ls -la
4. cd .emergency
5. ls -la
6. get .diego_priv.pgp

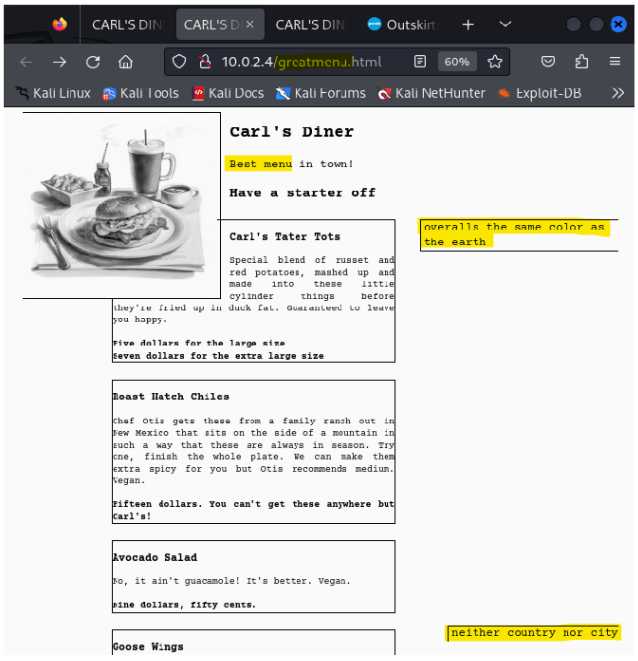


Next, I exited from ftp connection and checked the contents of my HackingEx directory in kali using the command **ls -la** and found **.diego_priv.pgp** hidden file. I converted the pgp file into a file that can be cracked by john the ripper by running the **gpg2john .diego_priv.pgp > diego_privkey.hash** command [2] and created a **diego_privkey.hash** file. I executed **cat diego_privkey.hash** command to check the content of the file which was Diego's private key details as shown:

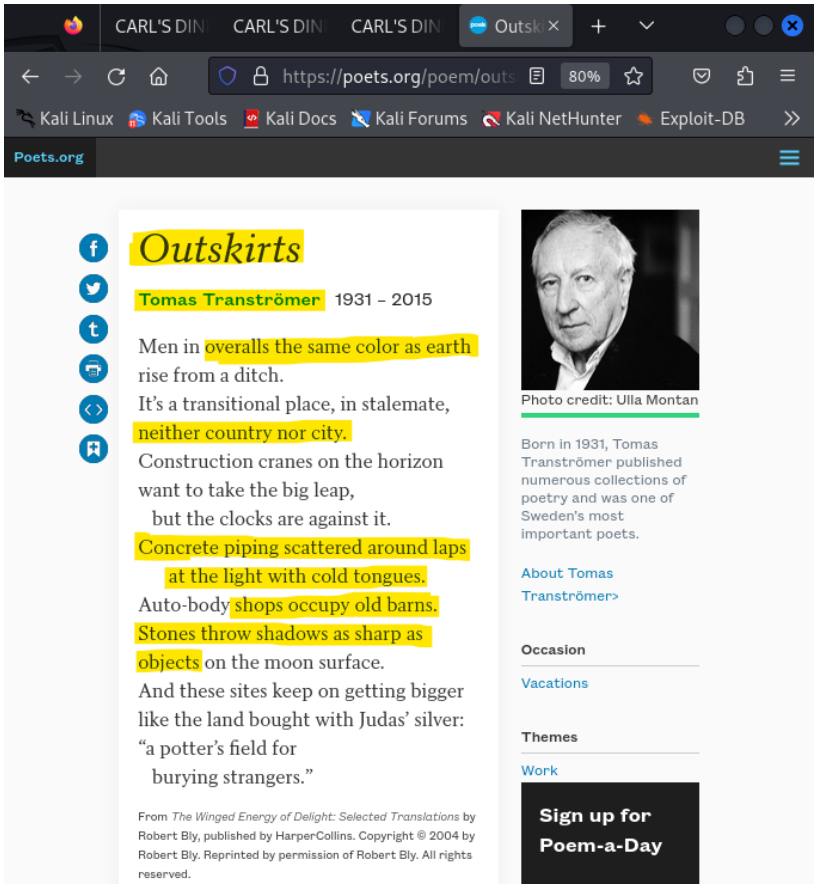
```
kali@kali: ~/Desktop/HackingEx
File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~/Desktop/HackingEx]
└─$ ls -la
total 150704
drwxr-xr-x 2 kali kali      4096 May  4 17:04 .
drwxr-xr-x 5 kali kali      4096 May  4 16:58 ..
-rw-r--r-- 1 kali kali      2635 May  2 18:42 BrunoPrivKey
-rw-r--r-- 1 kali kali      1057 Apr  8 12:48 diego.pgp
-rw-r--r-- 1 kali kali      2099 Apr  8 12:48 .diego_priv.pgp
-rw-r--r-- 1 kali kali      2381 May  1 21:47 HackExPwd.txt
-rw-r--r-- 1 kali kali 154275867 May  4 17:07 hydra.restore
-rw------- 1 kali kali     12288 May  3 13:17 .jounnalclue.swp
-rw-r--r-- 1 kali kali        33 May  3 13:21 journalclue

┌──(kali㉿kali)-[~/Desktop/HackingEx]
└─$ gpg2john .diego_priv.pgp > diego_privkey.hash

File .diego_priv.pgp

┌──(kali㉿kali)-[~/Desktop/HackingEx]
└─$ cat diego_priv.hash
cat: diego_priv.hash: No such file or directory

┌──(kali㉿kali)-[~/Desktop/HackingEx]
└─$ cat diego_privkey.hash
Diego:$gpg$*1*348*1024*bcec8207561358f82cb5b00351302871ee51bac3572e242f8f9f80f28ee797b65abb6b2a4c773
16b3f65d37cb83b826893a205c15f2fe85094f9e89365090bb8d011dd7111e442b4c35a89d0364332f1baa934e2dc51e29c0
2ff381983fa82d288ea6fc474548d1fd45749967fe6bee5faed4fba2ef7a8cb2a021c941524f14334a93d6a0126664a0e4a2
6f6bda2b9d1b45058c25f2ac581fd76c091a02ba14bc7afb3dac4759d2f1c022db8966d0b9ebf7ad6af3e7fbaf7c60a7c5fb
f79889832471388009997739873340 3f532ba257d58c78805d2d569d76034f69b41952983826357b73d52a04c25b3981cc1d
07664552d6b1b2a1d1a971fa7106f8f41bc0b61f3aebc25589b0b52be5d926c084f8f560ee463359da2a40fc9e860599be16
b9c45b703b73d7e78961fad761ea020b4f96f0217696703316b968633947ed3516e608f8235cfb3d3fe961707861f2b1cb04
1ee32bfc5bc3fc05777*3*254*2*7*16*87ebec3730ade263e9b22e46967fa8d1*65011712*fb7393d1a1d160c3:::Diego
<diego@carlsdiner.net>:::.diego_priv.pgp
```

I created my own wordlist consisting of words from Carl's diner website, some commonly found words, possible combination of passwords for all users with their names and Homework 2 clues into a **HackExPwd.txt** file. Some poetic or riddled sentences on the right-side boxes of Carl's website menu page caught my attention. I searched those phrases online and found that they are sentences from a poem called **Outskirts by Tomas Transtromer** [3]. I added all the words and phrases from that poem into my HackExPwd.txt file which is attached as part of this submission.



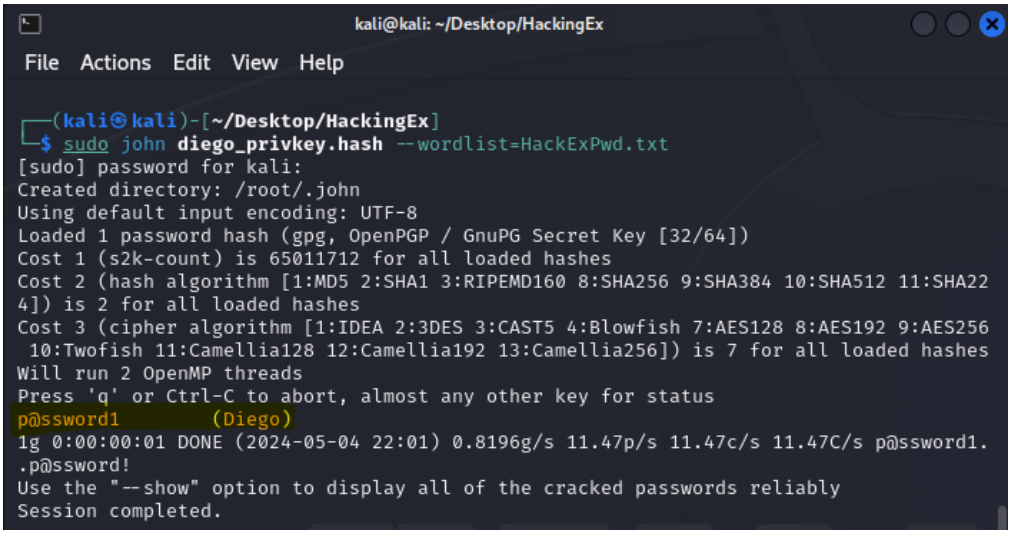


Poem – Outskirts by Tomas Transtromer:

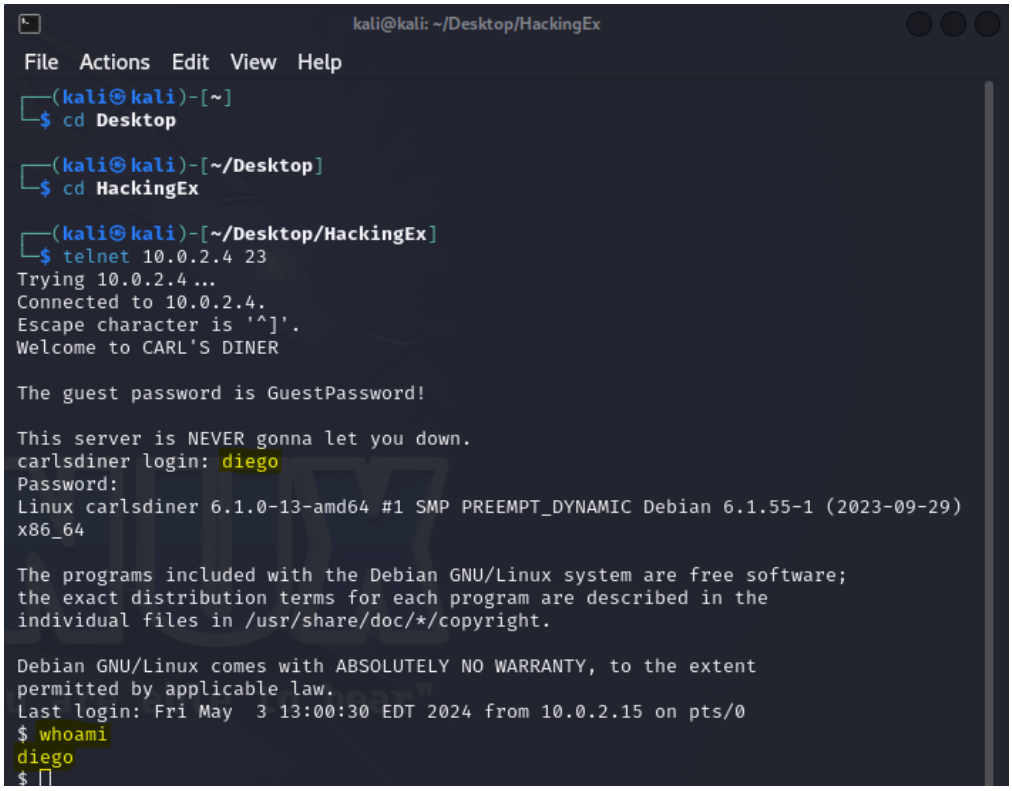


I ran john the ripper on Diego's private key using the custom-made wordlist to do a brute force attack and cracked the password of **Diego** to be **p@ssword1** as shown below:

Command: sudo john diego_privkey.hash –wordlist=HackExPwd.txt



I connected to the server using telnet and successfully logged in using the credentials I fetched for Diego as shown:



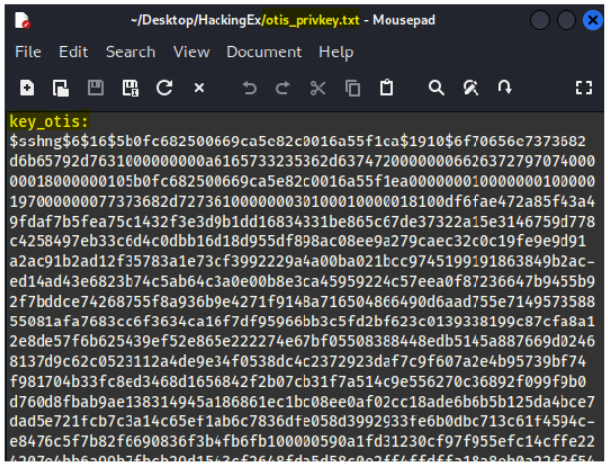
In the hackex_x86 Virtual Machine, I was successfully able to login both as guest and diego as shown:

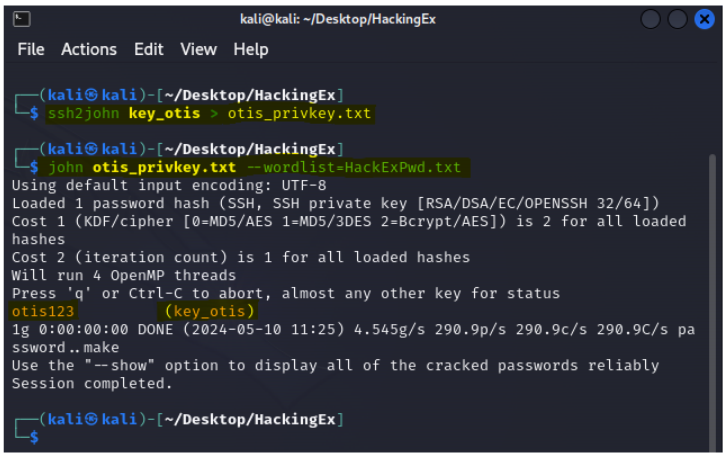
Guest: GuestPassword!

Diego: p@ssword1

Since the professor mentioned that in-class exercise involving **key_otis** was designed to prepare us for the hacking exercise, I suspected if there are any clues in the files with SSN and Email content and the public-private key pairs given during class. I found these files on Misc section of the Computer Security course on Blackboard, saved the key_otis file into my host kali.

I converted the file to be recognized by john the ripper using **ssh2john key_otis > otis_privkey.txt** command [2], ran John on the converted file **otis_privkey.txt** using my custom HackExPwd wordlist. I cracked the password of Otis's private key to be **otis123**, used this password to login to all the user accounts including Otis but was unsuccessful.

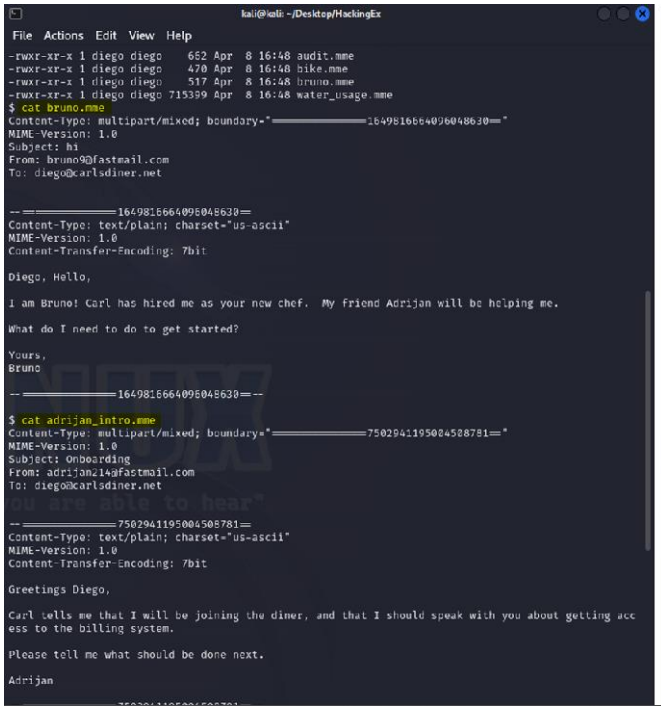




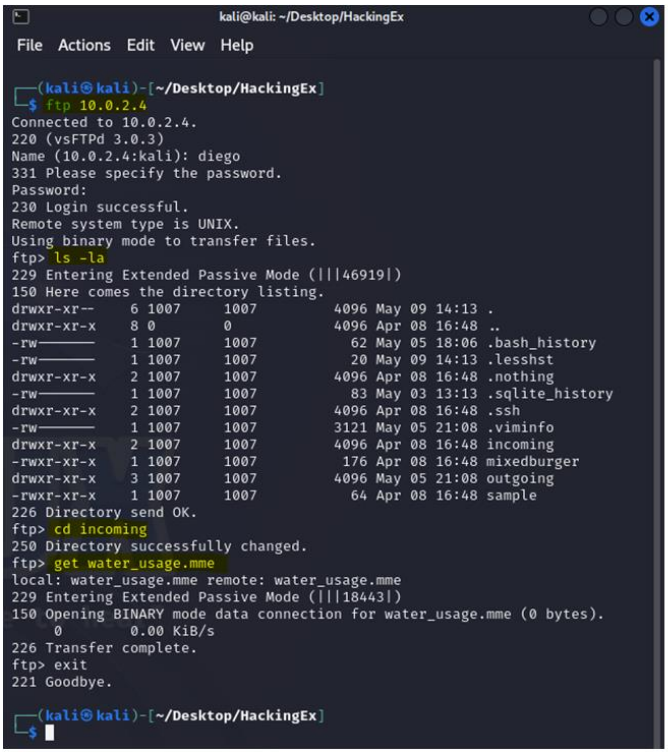
Next, I **logged in as Diego using p@ssword1** and checked the folder/file contents as follows:

A. Incoming folder: Here I found some **mme files** that provided me information that Diego got a new bike, and I found the roles/responsibility of each user in Carl's Diner as follows:

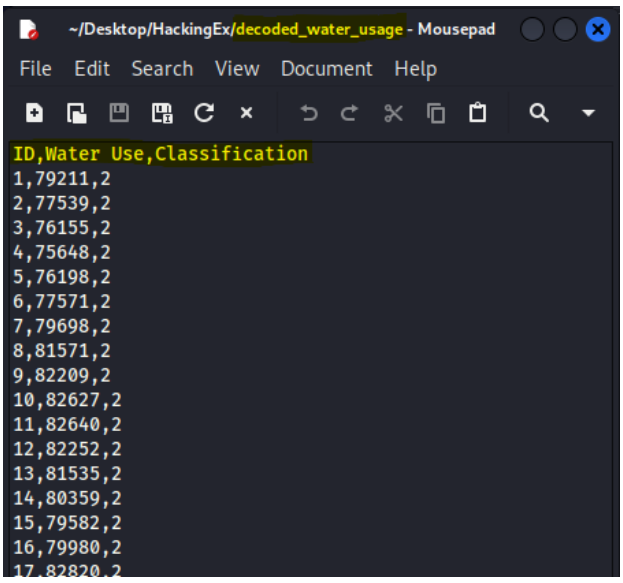
Diego – Security, Bruno – New chef, Adrijan – Billing, Otis – Audits, Carl - Owner



I found **water_usage.mme** file in the same folder which had some information about water bills and usage in csv format. I fetched the **water_usage.mme** file to my host kali by connecting to **ftp** and using the **get water_usage.mme** command as shown below:



Next, I copy pasted the water_usage.mme file content in Base64 decoding website [4] and stored the text in **decoded_water_usage** file. But this information was not useful in fetching any user passwords.



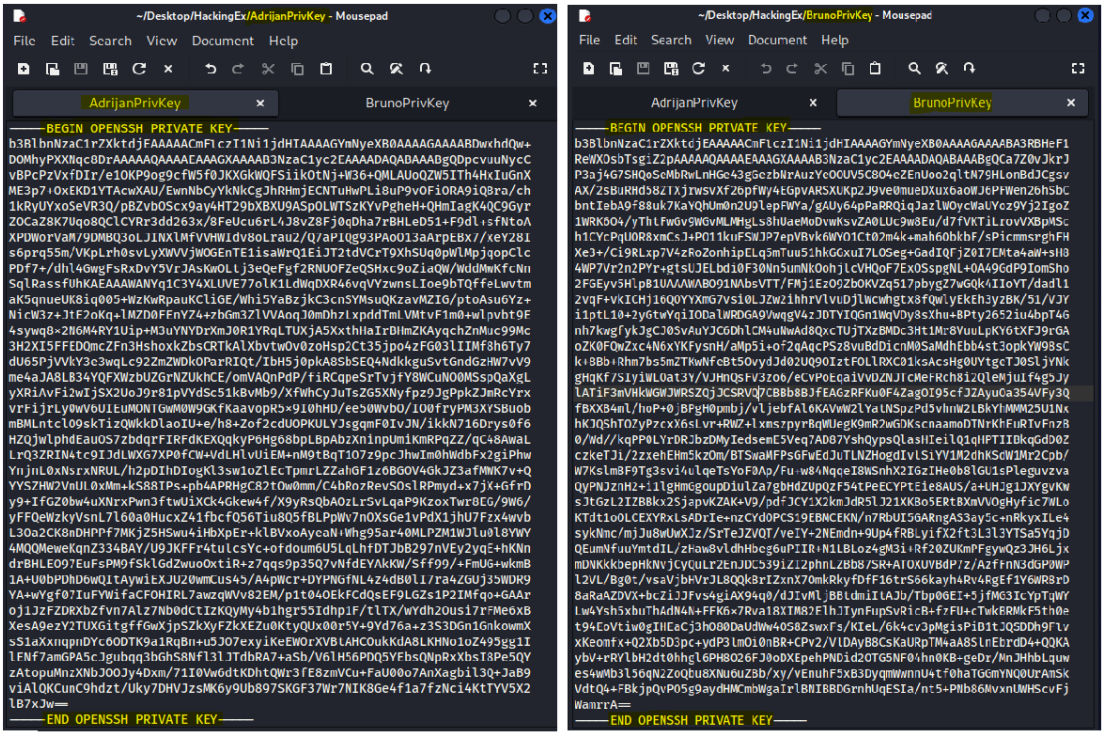
B. Mixedburger file: In this file, I found some information which was not useful to crack any passwords.

```
┌──(kali㉿kali)-[~/Desktop/HackingEx]
└─$ telnet 10.0.2.4 23
Trying 10.0.2.4 ...
Connected to 10.0.2.4.
Escape character is '^]'.
Welcome to CARL'S DINER

The guest password is GuestPassword!

This server is NEVER gonna let you down.
carlsdiner login: diego
Password:
Linux carlsdiner 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1 (2023-09-29) x8
6_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu May  9 14:32:37 EDT 2024 from 10.0.2.15 on pts/0
$ ls
incoming  mixedburger  outgoing  sample
$ cat mixedburger
5 lbs of ground goose
5 lbs of ground duck
5 lbs of ground spicy sausages

- mix together??
- add an egg
- what egg?

best kind of bread


THIS IS WORK IN PROGRESS
$
```

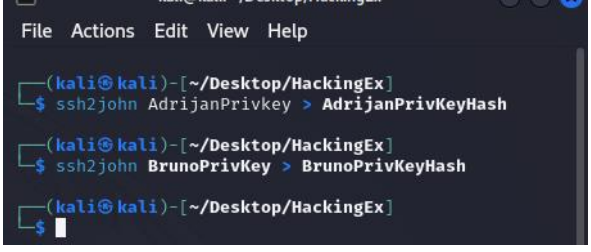
C. Outgoing folder: In this folder, I found **adrijan.mme**, **bruno.mme** and **locks.mme** files.

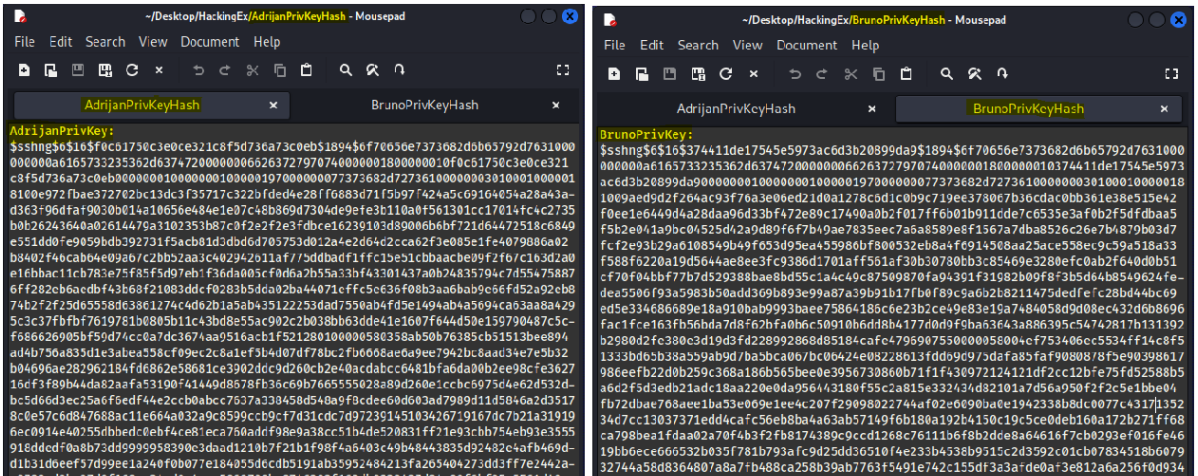
I opened **adrijan.mme** and **bruno.mme** files, copied the base64 encoded content from both the files.

I decoded the content of both adrijan.mme and bruno.mme files using Base64 decoder website [4] and stored the OpenSSH private keys in **AdrijanPrivKey** and **BrunoPrivKey** files respectively as shown:

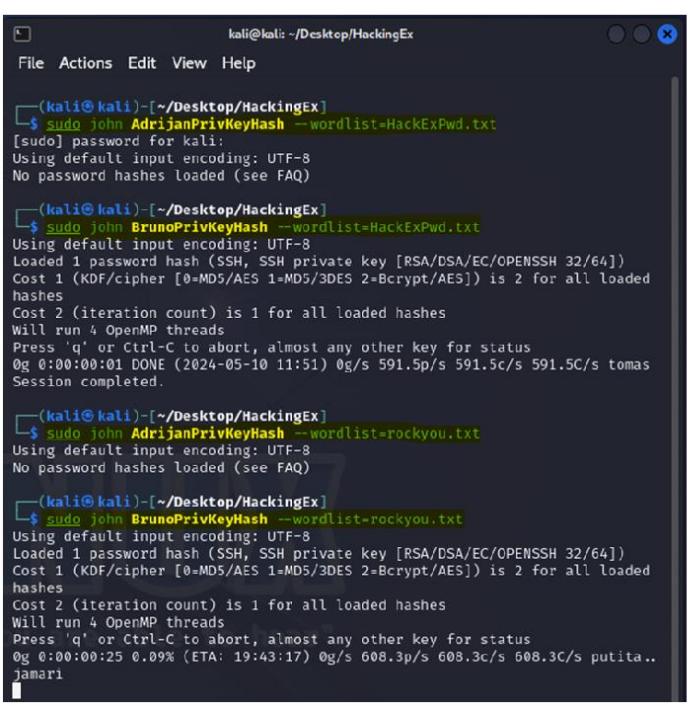


I converted the OpenSSH private keys of Adrijan and Bruno into files that can be cracked with john using **ssh2john** command [2] as shown and created **AdrijanPrivKeyHash** and **BrunoPrivKeyHash** files.



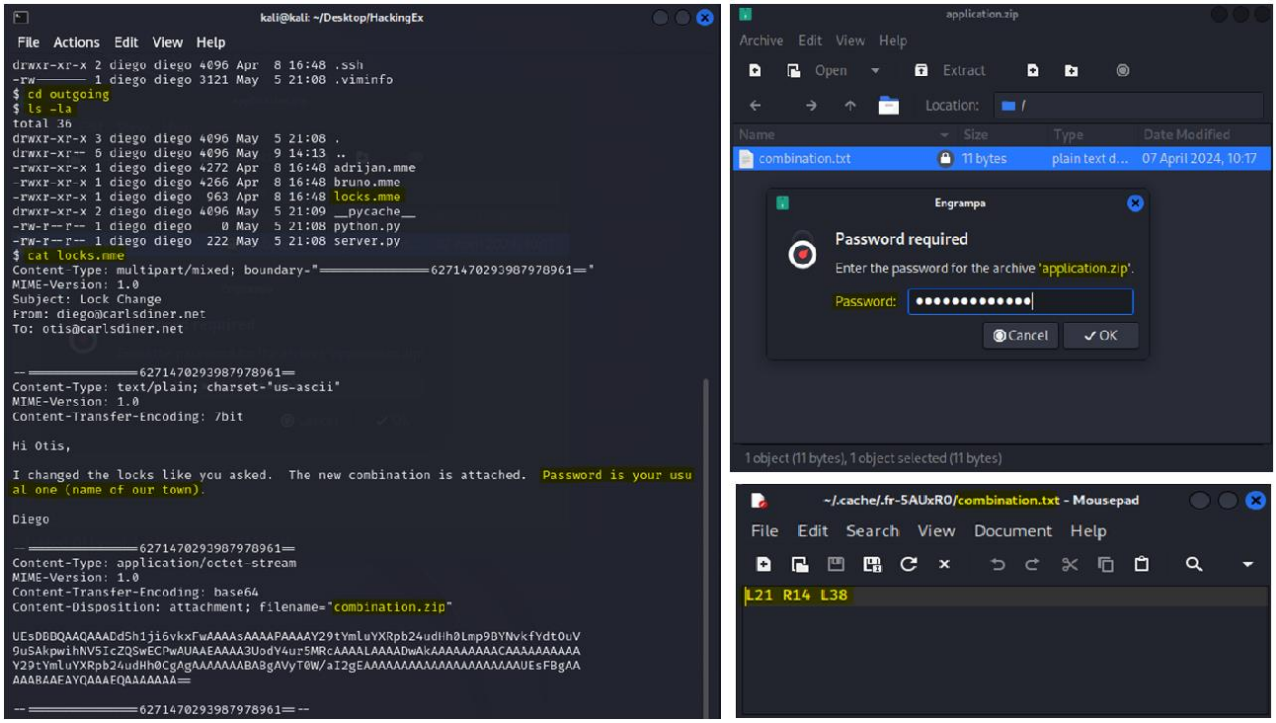


After obtaining the private keys, I ran John the ripper and hydra on both AdrijanPrivKeyHash and BrunoPrivKeyHash with my custom wordlist HackExPwd.txt and rockyou.txt for 3-4 days but I didn't get any results.

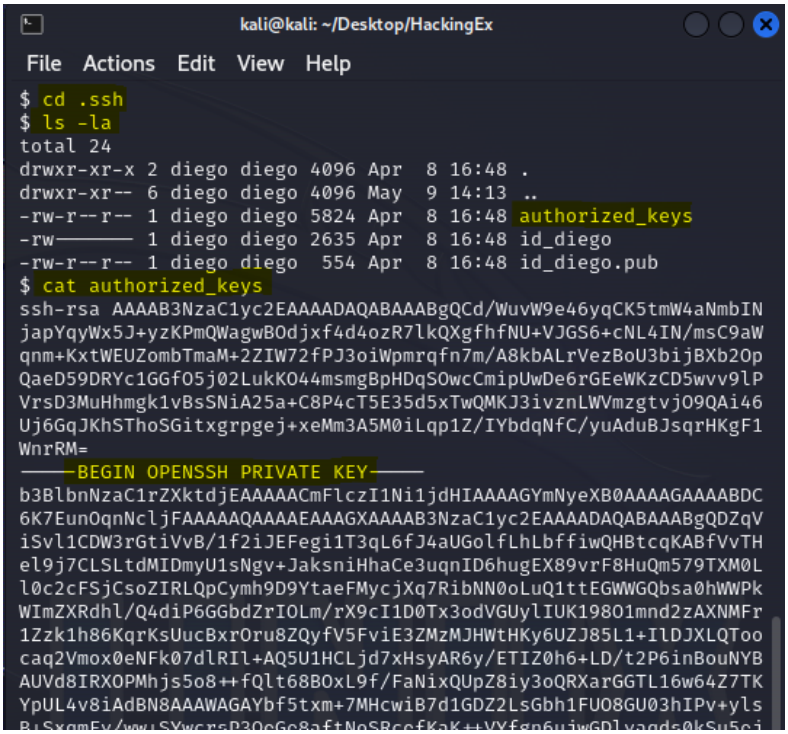


Next, I checked **locks.mme** file and found an email sent by Otis to Diego with a password protected combinations.zip file. I copied the encrypted Base64 code, decrypted it to a file using an online tool [5] and downloaded the file **application.zip** on my host.

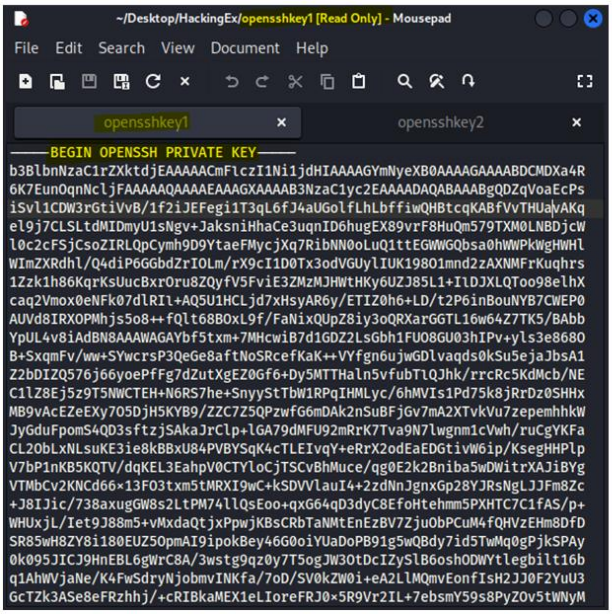
Using the clues given by Otis in the mail, I tried variations of town SouthCleric, Arizona from Carl's diner website, opened the file with **southclericaz** as password and found some content which was not useful for cracking any password as shown:



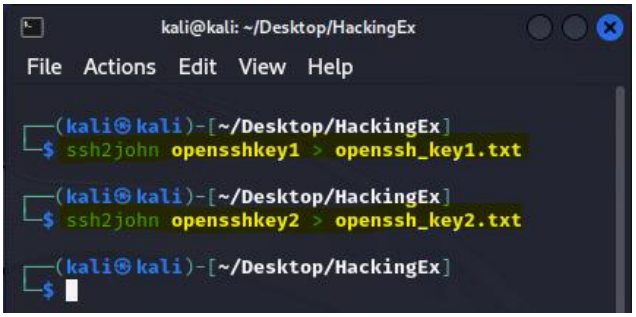
D. Hidden folder .ssh: In this folder, the authorized_keys file caught my eyes, I opened it using the command **cat authorized_keys** and found **two OpenSSH private keys** in it.

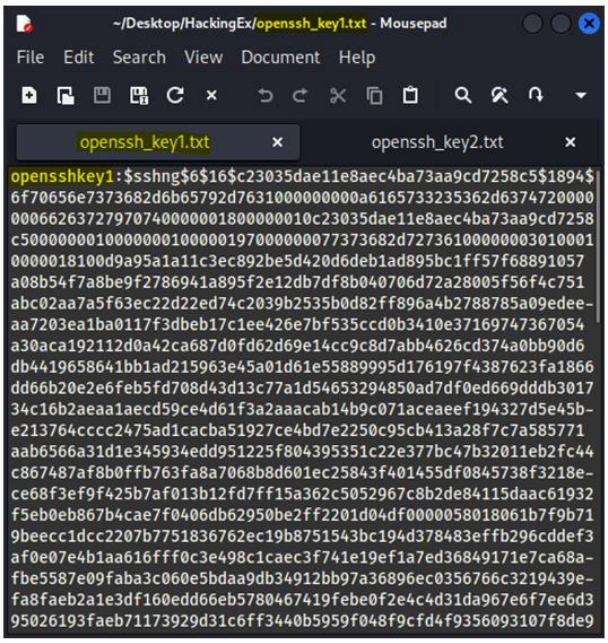


I copy pasted each OpenSSH key from the authorized_keys file into separate files called **opensshkey1** and **opensshkey2** as shown below:

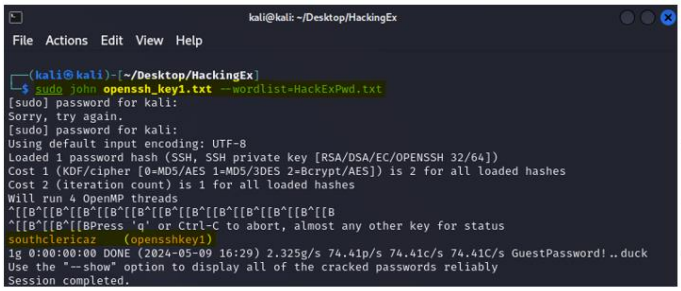


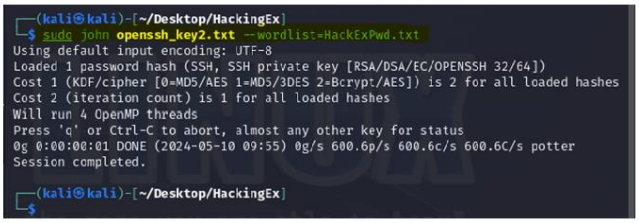
Since these private keys are not recognizable by John, I converted the private keys into files that can be cracked with john the ripper by using **ssh2john** command [2] as shown below and created **opensshkey_1.txt** and **openssh_key2.txt** files.





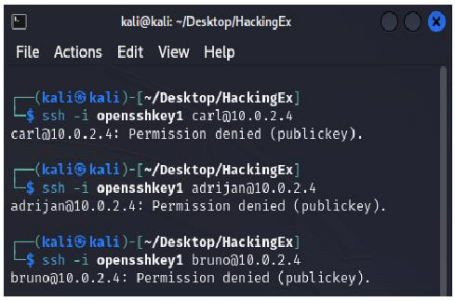
I ran John the ripper on both openssh keys with my custom wordlist **HackExPwd.txt** and was able to crack password for **openssh_key1.txt** as **southclericaz** as shown below:

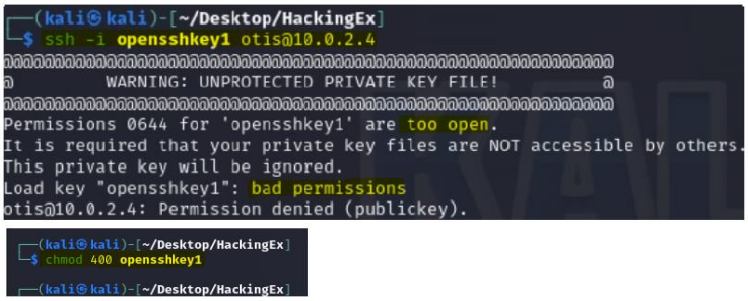




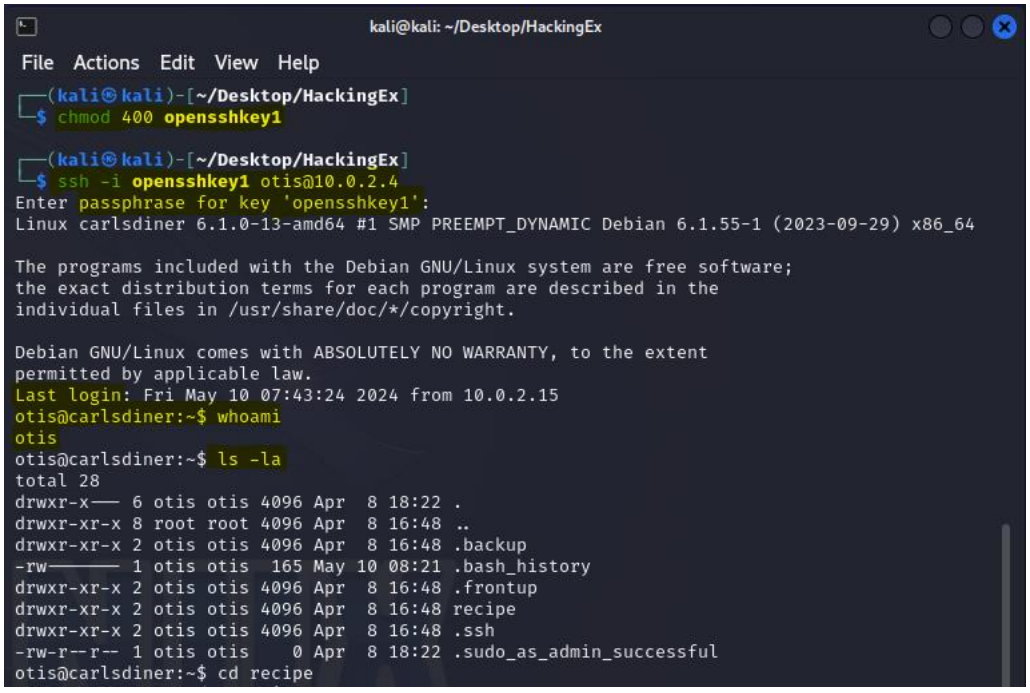
Since **southclericaz** was also the password for combinations.zip file in Diego, I felt like this authorization key and password should be of use somewhere else to gain access to the accounts.

I tried **ssh** on different users by running the **ssh -i opensshkey1 carl@10.0.2.4** command [6] and similarly for Adrijan and Bruno but was not successful. Next, I tried ssh remote login for user Otis using **ssh -i opensshkey1 otis@10.0.2.4** command [6], got some bad permissions error and rectified it by giving less permission to private key file using the **chmod 400 opensshkey1** command [7] as shown:



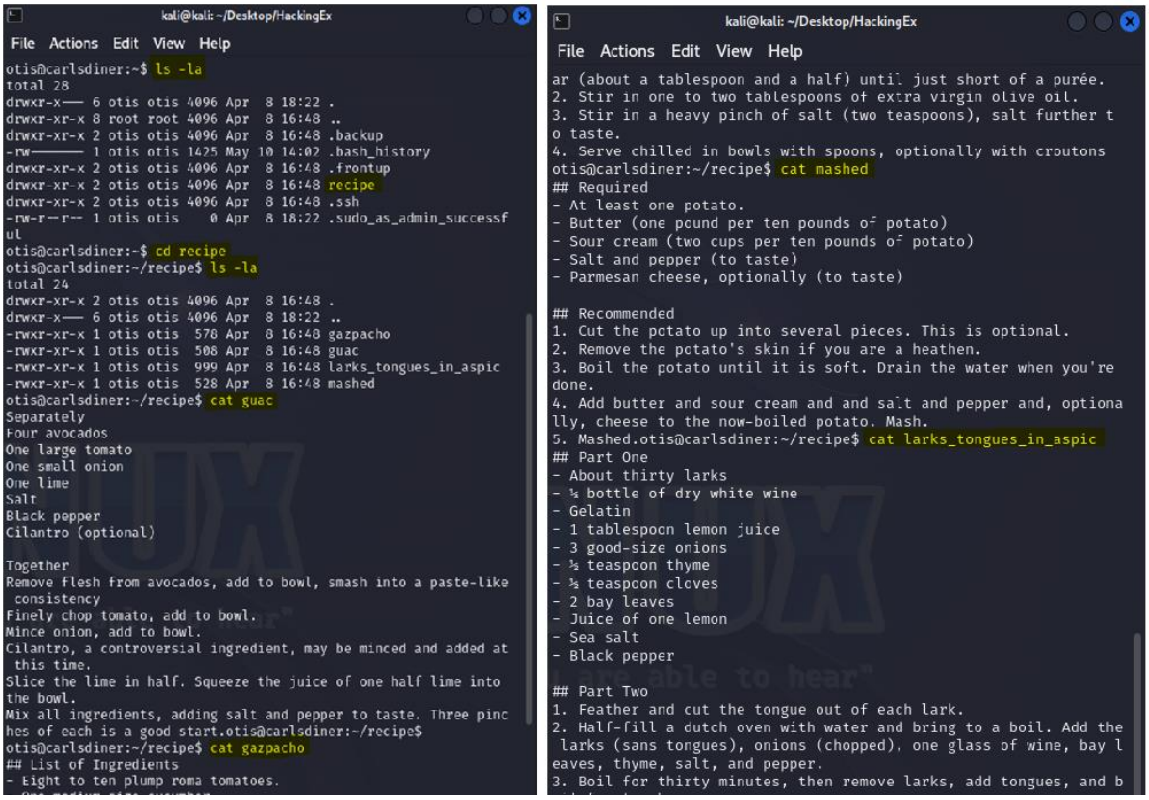


I ran the command **ssh -i opensshkey1 otis@10.0.2.4** again after rectifying the issue, entered the passphrase as **southclericaz** and was able to successfully access user **Otis** account as shown below:

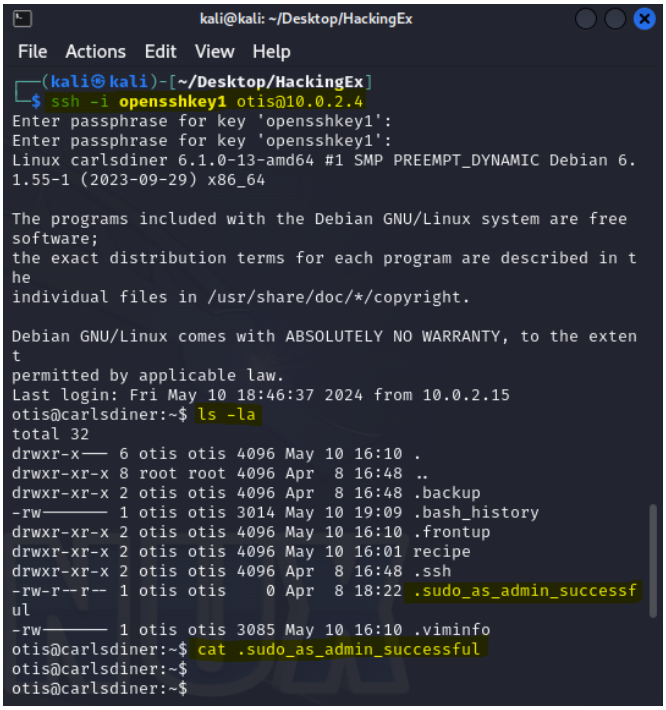


Without knowing the password for **Otis** who is in **sudoers** list, I cannot access any files or switch users or change passwords of other users. So, I started looking for clues to crack Otis password. I logged in to Otis account using ssh and southclericaz as passphrase, checked the contents of the account.

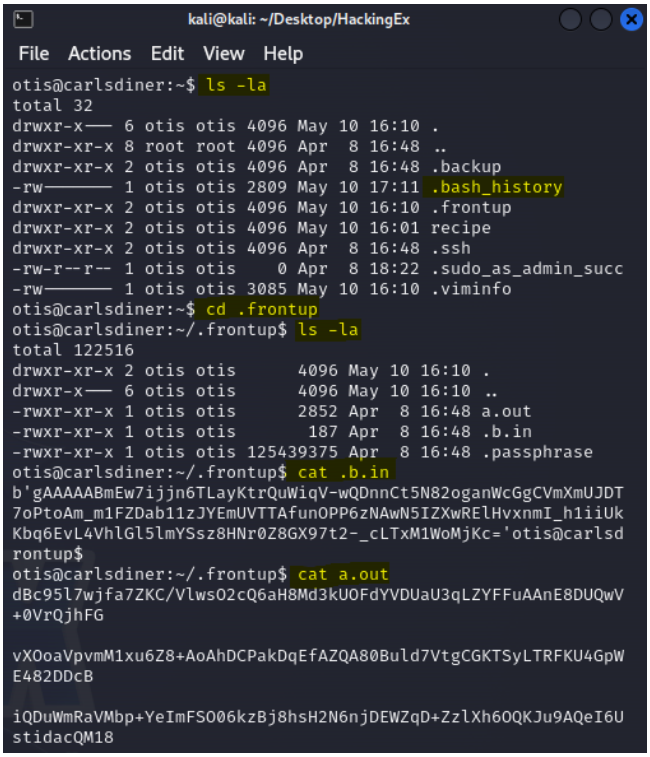
A. Recipes Folder: I found four recipes in the recipe folder, looked for clues in all the 4 recipes and tried some manual brute forcing which were not useful.



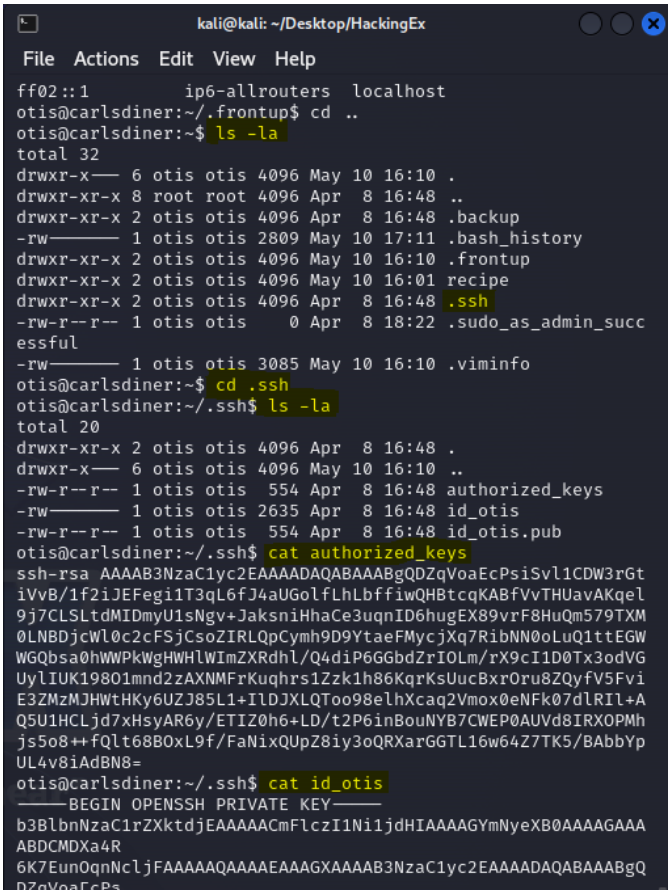
B. Hidden file .sudo_as_admin_successful: This hidden file was empty.



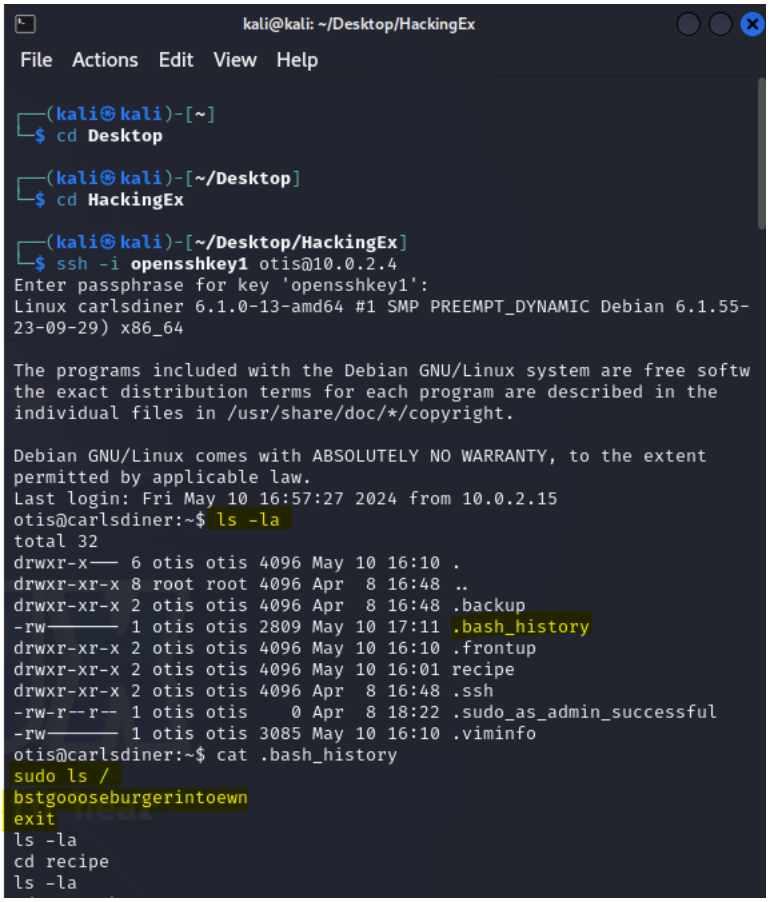
C. Hidden Folder .frontup: I checked the .frontup folder and found 3 files there with some encrypted texts and keys which were not useful.



D. Hidden Folder .ssh: I checked the .ssh folder and found 3 files there with some encrypted texts and authorized keys, I ran john on the keys but not useful since I had already cracked it to be southclericaz.



E. Hidden file .bash_history: Next, I opened the hidden bash_history file and checked the entries. The first three entries in the bash_history was suspicious to me since I had not used those commands in my **otis@carlsdiner** terminal.



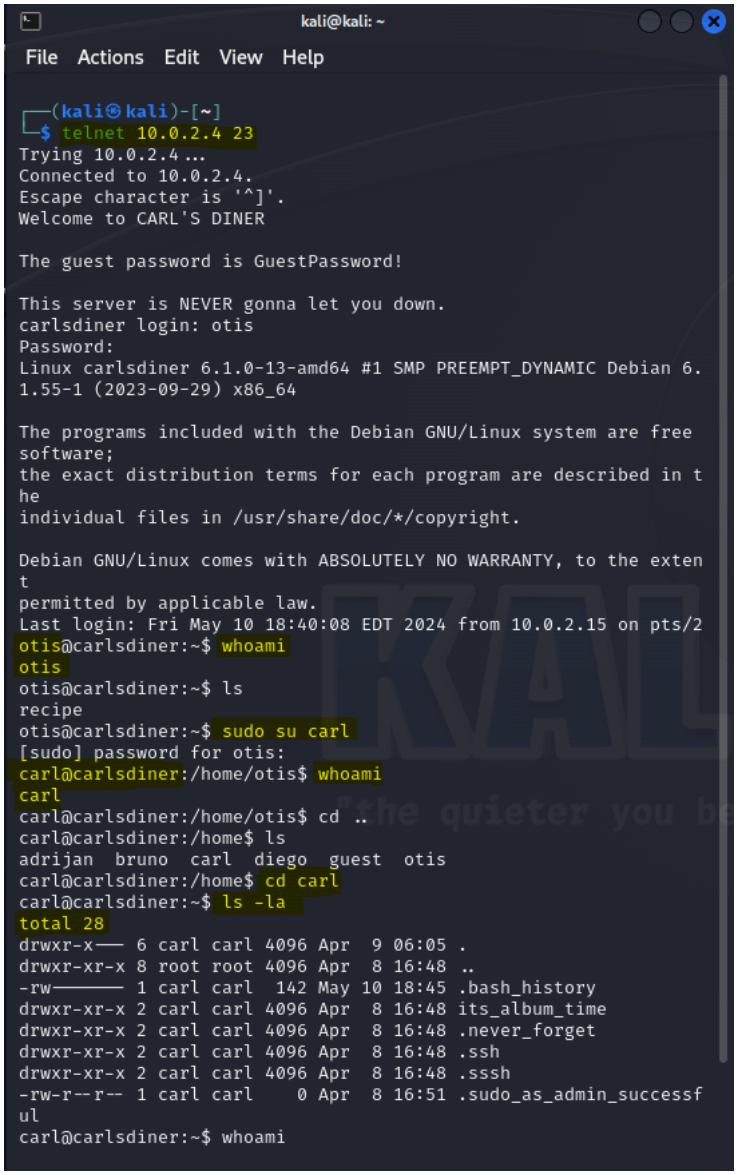
I tried bstgoooseburgerintoewn as password for logging in as otis and failed. I corrected the spelling and tried **bestgooseburgerintown** as password for otis and was able to login to the hackex_x86 VM successfully as shown below:

Otis: bestgooseburgerintown

```
carlsdiner login: otis
Password:
Linux carlsdiner 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1 (2023-09-29) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri May 10 16:10:39 EDT 2024 from 10.0.2.15 on pts/0
otis@carlsdiner:~$
otis@carlsdiner:~$
```

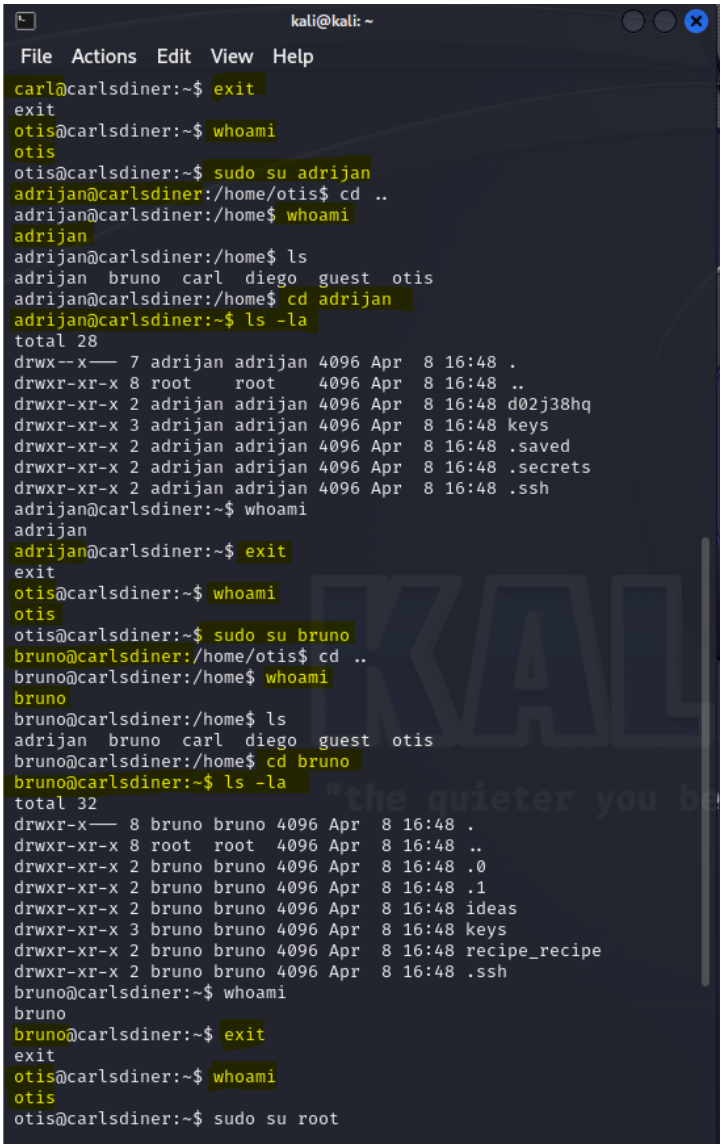
Since otis has sudo access permissions and I cracked Otis's password, I can **switch** to other users including root using **sudo su <username>** command [8] which will ask for Otis's password, **access** all the user information, **change passwords** of users, **change file permissions** and many more **by logging in as Otis** as shown below:

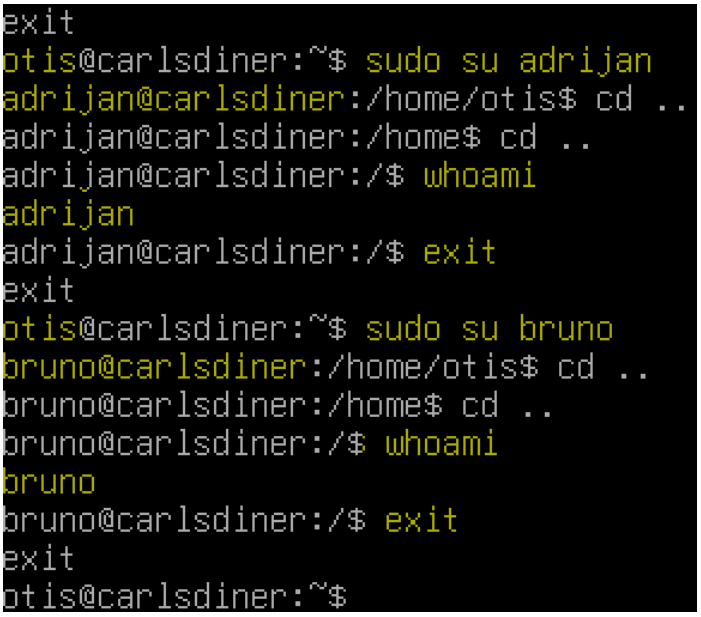
Gaining access to **Carl** using otis login and accessing Carl's information:



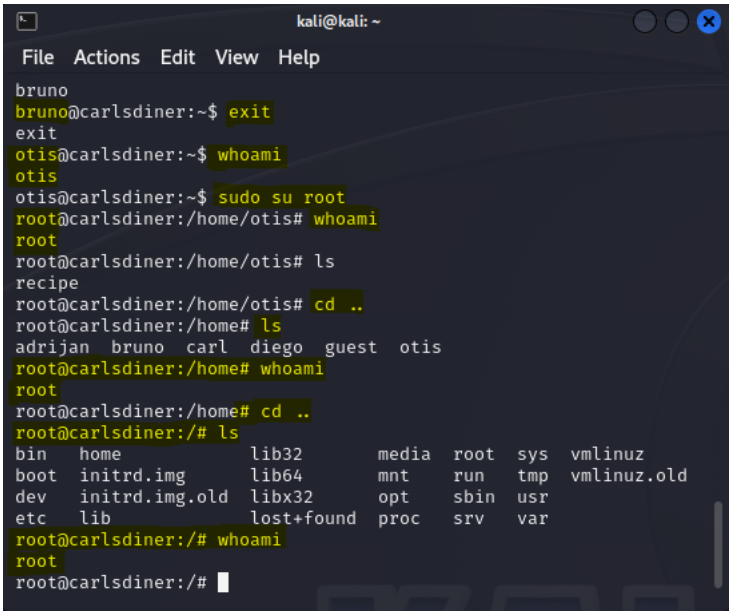

```
otis@carlsdiner:~$ whoami
otis
otis@carlsdiner:~$ sudo su carl
carl@carlsdiner:/home/otis$ cd ..
carl@carlsdiner:/home$ cd ..
carl@carlsdiner:/$ ls
bin   etc         initrd.img.old  lib64       media  proc  sbin  tmp  vmlinuz
boot  home        lib             libx32      mnt    root  srv   usr  vmlinuz.old
dev   initrd.img  lib32           lost+found  opt    run   sys   var
carl@carlsdiner:/$ whoami
carl
carl@carlsdiner:/$ exit
```

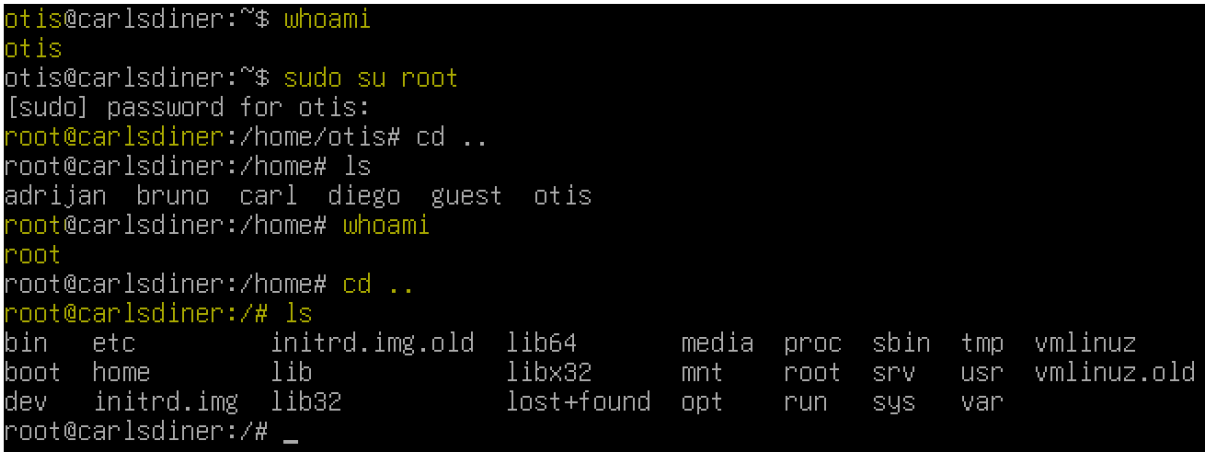
Gaining access to **Adrijan and Bruno** using otis login and accessing Adrijan's and Bruno's information:





Gaining access to **root** login using otis and root information:





Overall, the hacking exercise was too interesting, made me learn a lot of things including patience and perseverance. Thank you so much for this amazing learning opportunity!!!

I successfully gained access to all the user accounts – guest, diego, otis, carl, adrijan, bruno and root using otis with sudo access. I cracked passwords for 3 users as shown below:

SL No	User	Password
1	guest	GuestPassword!
2	diego	p@ssword1
3	otis	bestgooseburgerintown

References:

[1] Referred from in-class hacking exercises, homework assignments, command line --help option for usage and useful reference links shared in this exercise.

[2] Password Attacks, Available: Password Attacks John the Ripper

[3] Outskirts by Tomas Transtromer, Available: Outskirts by Tomas Tranströmer - Poems

[4] Base64 Decode and Encode, Available: Base64 Decode and Encode - Online

[5] Base64 to File, Available: Base64 to File | Base64 Decode | Base64 Converter | Base64

[6] A Beginners Guide to SSH for Remote Connection, Available: A beginner's guide to SSH for remote connection on Linux | Opensource.com

[7] How to Crack SSH Private Key Passwords, Available: How to Crack SSH Private Key Passwords with John the Ripper « Null Byte

[8] Su Command in Linux, Available: Su Command in Linux (Switch User) | Linuxize