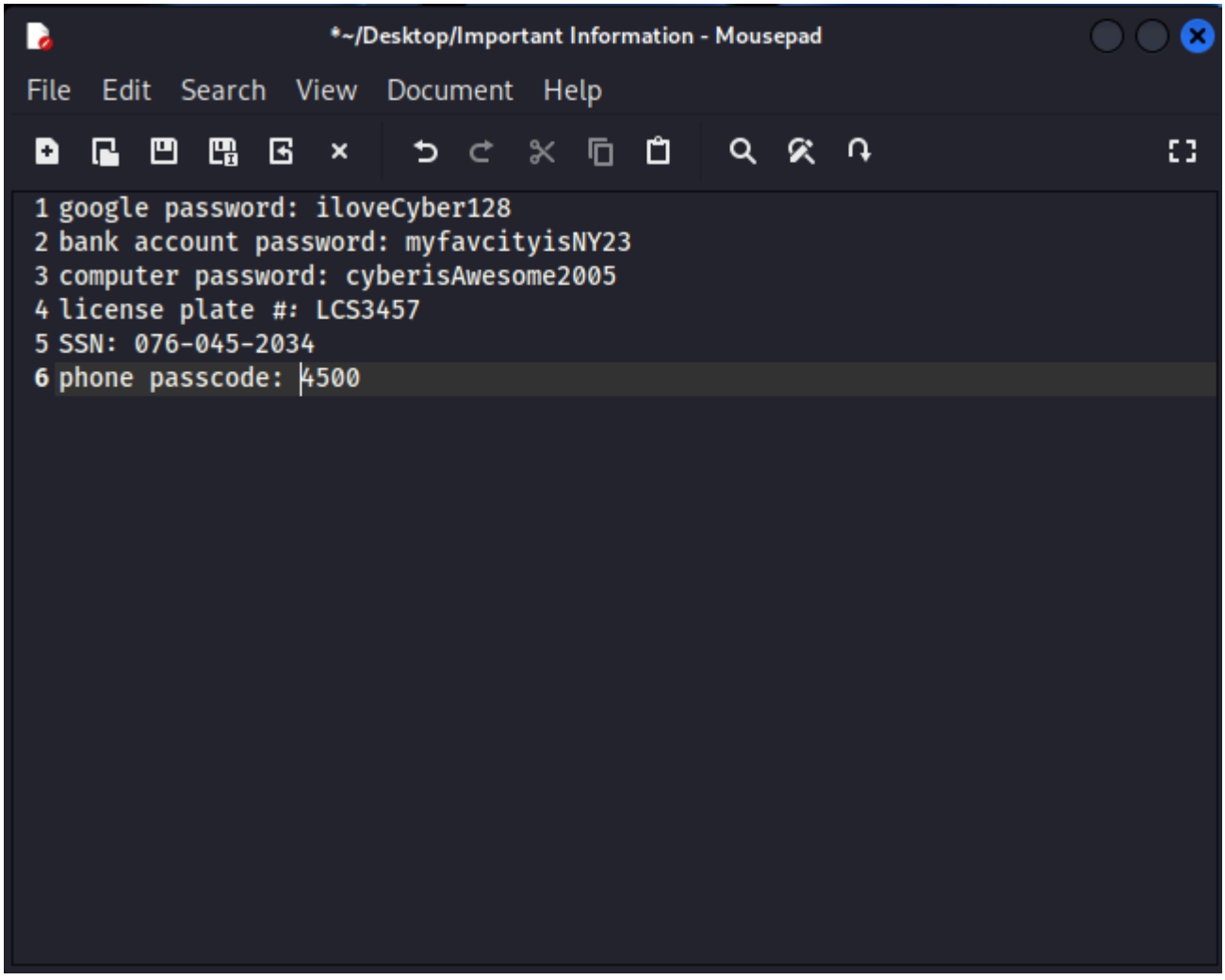
Cyber Attacks Practice
12/14/25

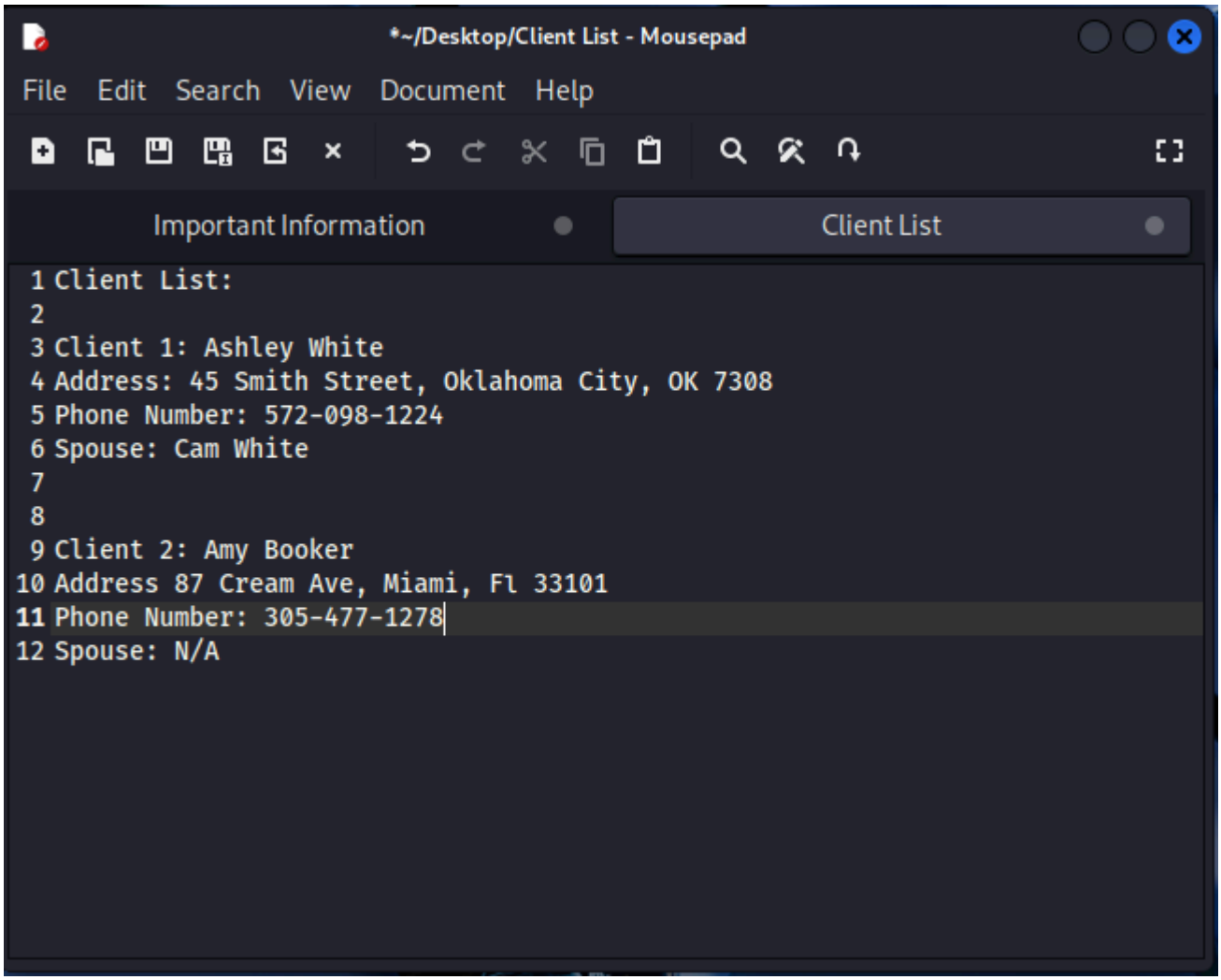
The purpose of this lab is to practice different attacks on different components of a machine. I will be using a virtual machine running Kali Linux to create and execute these exploits.

I will create password-encrypted files and use John the Ripper to crack these passwords. The goal is to identify which passwords were considered strong and which were

weak.

These are the files I want to password-encrypt because they have sensitive data that I want to protect.





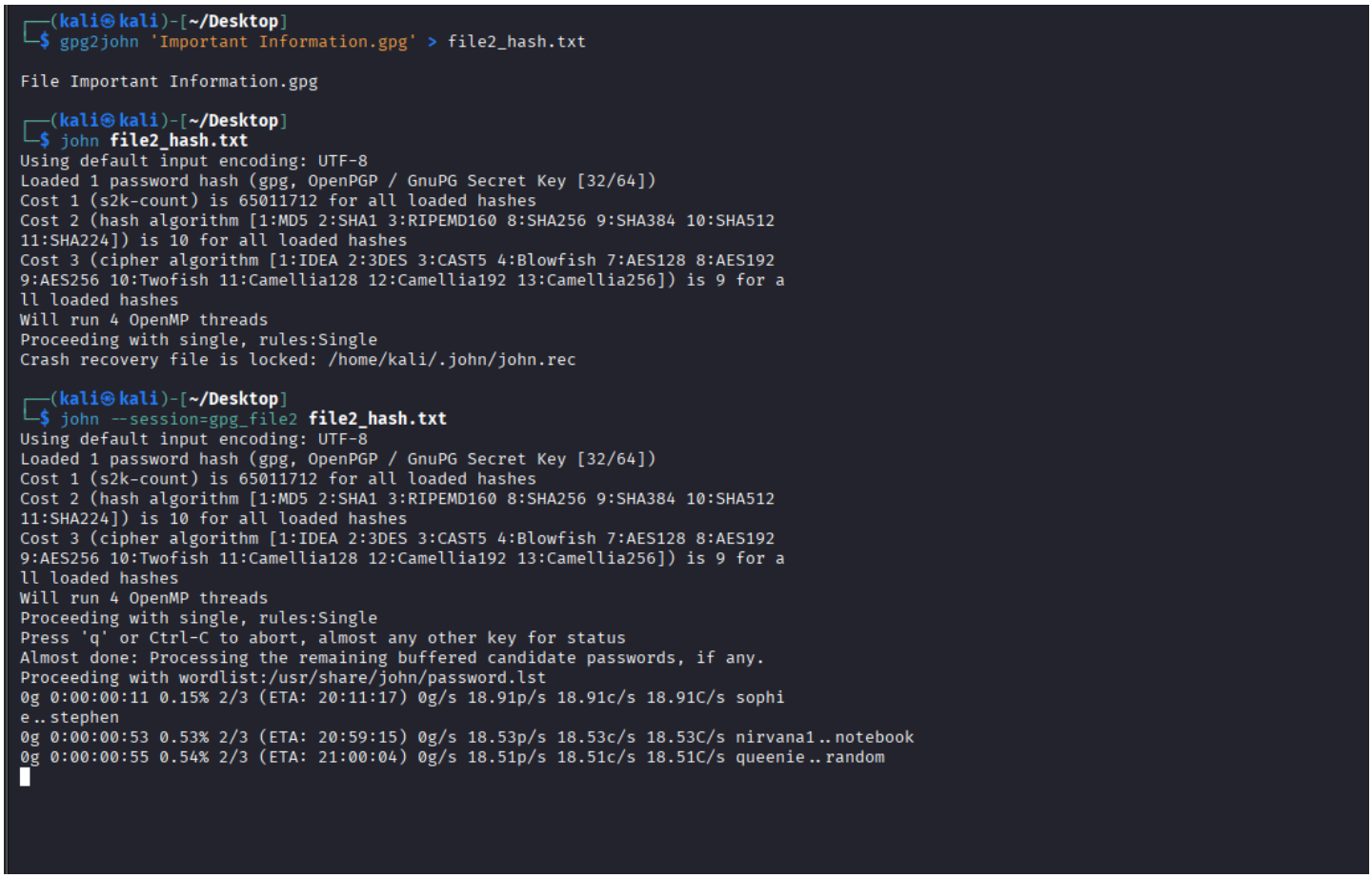
Now I will encrypt both of these files with passwords using the GPG command. One with a strong password and one with a weaker password.

GPG is one of the strongest password encrytpers and simply using the gpg command again can reverse the encryption, but for this experiment I want to use John The Ripper to actually find out what password was used to encrypt.

Even with a relatively strong password (the weaker password I used), according to the updates I get when I extracted the hash and attempt to break it, it takes approximately 1 hour. .

```
(kali㉿kali)-[~]
$ cd Desktop

(kali㉿kali)-[~/Desktop]
$ gpg2john 'Client List.gpg' > gpghash.txt

File Client List.gpg

(kali㉿kali)-[~/Desktop]
$ john gpghast.txt
stat: gpghast.txt: No such file or directory

(kali㉿kali)-[~/Desktop]
$ john gpghash.txt
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 10 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 9 for all loaded hashes
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
0g 0:00:00:13 0.29% 2/3 (ETA: 19:02:09) 0g/s 38.34p/s 38.34c/s 38.34C/s chloe..compaq
0g 0:00:01:24 1.62% 2/3 (ETA: 19:13:59) 0g/s 38.19p/s 38.19c/s 38.19C/s trash..turbo2
0g 0:00:01:59 3.98% 2/3 (ETA: 18:37:11) 0g/s 38.19p/s 38.19c/s 38.19C/s Gemini..Germany
0g 0:00:02:00 4.00% 2/3 (ETA: 18:37:23) 0g/s 38.19p/s 38.19c/s 38.19C/s Jaguar..Jenny1
0g 0:00:02:49 5.02% 2/3 (ETA: 18:43:28) 0g/s 38.14p/s 38.14c/s 38.14C/s Gilgamesh..Goethe
0g 0:00:02:50 5.04% 2/3 (ETA: 18:43:36) 0g/s 38.15p/s 38.15c/s 38.15C/s Jetta1..Joy
0g 0:00:04:32 7.25% 2/3 (ETA: 18:49:50) 0g/s 38.08p/s 38.08c/s 38.08C/s minnie1..mouse1
0g 0:00:05:07 7.98% 2/3 (ETA: 18:51:28) 0g/s 38.09p/s 38.09c/s 38.09C/s vanessa1..vermont1
0g 0:00:05:08 8.01% 2/3 (ETA: 18:51:27) 0g/s 38.09p/s 38.09c/s 38.09C/s action1..artist1
```

With the stronger password it takes twice as long and I had to create a new session to ensure that I could crack both at the same time. I will let them run and see what happens.



Unfortunately, I could not crack these passwords in time for this lab, but my research does show that strong passwords made using GPG are intended to take time to crack and sometimes can be infeasible. In this case, if this password cracking were time sensitive, this would be infeasible.