This is CS50x

OpenCourseWare

Donate (https://cs50.harvard.edu/donate)

David J. Malan (https://cs.harvard.edu/malan/)
malan@harvard.edu
(https://www.facebook.com/dmalan) (https://github.com/dmalan) (https://www.instagram.com/davidjmalan/) (https://www.linkedin.com/in/malan/) (https://orcid.org/0000-0001-5338-2522) (https://www.quora.com/profile/David-J-Malan) (https://www.reddit.com/user/davidjmalan) (https://www.tiktok.com/@davidjmalan) (https://twitter.com/davidjmalan)

Cybersecurity

Passwords

- First, we poll the audience with a question: “Is your phone secure?”
  - About a third of the audience responded for each of “Yes”, “No”, and “Unsure”.
- The passcode to our phones might be four digits, longer, or nothing at all.

- Some of us might think that our password is more secure because they are random.
- The most common passwords (https://nordpass.com/most-common-passwords-list/) as of 2020, unfortunately, are easy to guess:
    1. `123456`
    2. `123456789`
    3. `picture1`
    4. `password`
    5. `12345678`
    6. `111111`
    7. `123123`
    8. `12345`
    9. `1234567890`
    10. `senha`
        - This translates to “password” in Portuguese.
- If we have one of these passwords ourselves, it would be easy for someone to break into our account or device, since they’re most likely to try these passwords first.
- So, we can start to quantify how secure our phones are, by quantifying how secure our passwords are.
- With a 4-digit passcode, we have 10,000 possible passcodes, from `0000` to `9999`. But a computer can generate all of them quickly, in just a few seconds.
- And someone can perform a **brute-force attack**, where they try all possible passwords until the correct one is found.
    - An adversary can even build a robot to tap all possible passcodes on a phone screen.
- We can demonstrate this by writing a program that prints out all possible products, or permuations, of 4-digit passcodes, in a programming language called Python:

```
from string import digits
from itertools import product

for passcode in product(digits, repeat=4):
    print(*passcode)
```

```
$ python crack.py
0 0 0 0
```

```
  0 0 0 1
  ....
  9 9 9 8
  9 9 9 9
  ```

  - This program takes less than a second to run.
  - We can imagine that an adversary might plug a cable into a phone, and use a computer with code to try all possible passcodes very quickly.
- With more digits, it will take longer for our passcode to be guessed, raising the cost for our adversary.
- We can also use a 4-letter passcode. Since we have 26 different letters possible for each place, and each of them can be uppercase or lowercase, we would have `52 x 52 x 52 x 52` possibilities, or more than 7 million possible passcodes.
- We can change our program to use letters instead of digits:

  ```
  from string import ascii_letters
  from itertools import product

  for passcode in product(ascii_letters, repeat=4):
      print(*passcode)
  ```

  ```
  $ python crack.py
  a a a a
  a a a b
  ...
  Z Z Z Y
  Z Z Z Z
  ```

  - Now this program takes more than a minute to run.
- We can expand our passcode to use any combination of characters, like letters, numbers, and symbols like `!` or `#`.
- On a typical keyboard, we'll have 32 symbols, in addition to 26 uppercase letters, 26 lowercase letters, and 10 numbers, for a total of 94 different symbols. So if we have an 8-character passcode, we'll be able to choose between `94 x 94 x 94 x 94 x 94 x 94 x 94 x 94` possibilities, which is over 6 quadrillion.
  - If each passcode takes a second to try, it would take more than 193,000 years to try all possibilities.
- We'll change our program to use every character:

  ```
  from string import ascii_letters, digits, punctuation
  ```

```
from itertools import product

for passcode in product(ascii_letters + digits + punctuation, repeat=8):
    print(*passcode)
```

  - Now, this program will take much longer to run.
- Longer passwords are harder to crack, or discover, but they are more difficult to remember.
- On many devices, like an iPhone or Android, trying to log in incorrectly too many times in a row will lock us out from further attempts, telling us to try again in a minute or more. It turns out that this is a security feature, slowing down our adversaries who might be trying to guess our passcode. Now, even with 10,000 possibilities for a 4-digit passcode, this might take 10,000 minutes or more.
- **Two-factor authentication** refers to the use of an additional format of information to log in, such as a one-time use code from a message or app. Since this tends to be something we *have*, in addition to what we *know* (our username and password), it's even more difficult for adversaries to log into our account.
- **Password managers** are applications that store login information for us, so instead of remembering many different, complex passwords, we only need to remember a single master password. Some popular ones include:
  - Credential Manager in Windows
  - Keychain in macOS
  - 1Password
  - LastPass
- Our password manager can generate, save, and fill long, unique passwords for all of our other accounts.
- The downside of a password manager might be a greater risk to us, if our master password is discovered or forgotten. Then, we'd lose all of our accounts at once.

Email, private mode

- In Gmail, there is new feature called "Confidential mode", where we are able to send emails and "Recipients won't have the option to forward, copy, print, or download this email."
- But this is a bit misleading, since we can still take a screenshot of the email, or even a photo of our computer's screen.
- Our browsers, too, might provide an "incognito mode" or "private mode", where our browsing history and other pieces of data aren't saved locally.
- But the websites we visit and our internet service provider might still see what we're visiting.

Encryption

- **Encryption** is the scrambling of information so that it can’t be read without a key to decrypt it.
- The phrase `T H I S W A S C S 5 0`, for example, might be simply encrypted to `U I J T X B T D T 5 0`, by rotating each letter once.
- `https://` is a secure, encrypted way for browsers to communicate with web pages, without anyone in between able to read the contents.
- **End-to-end encryption** means that our messages are encrypted between us and who we are talking to, so even if we are using a third-party chat or video service, the companies in between are not able to decrypt, or read the contents of our communications.
- Zoom, for example, had previously advertised end-to-end encryption, but only implemented it as encryption between us and Zoom. Only recently have they rolled out true end-to-end encryption between the participants in a meeting, but some features won’t work as a result, like cloud recording and phone call-in.
- To summarize, here are a few suggestions we have, to be more secure:
  1. Use a password manager
  2. Use two-factor authentication
  3. Use (end-to-end) encryption