Mobile permission Jan 27, 2021

Scribe: Eric Chen

Previously, we have multi-user OS architecture:

- User-based
- Main threat: another user attacks me
- Problems:
  - malware (if download a malicious app),
  - vulnerable program (a music program that has vulnerabilities, a mp3 that has malware).
  - 10% of computers have malware on them

Mobile: a completely new architecture

app bases:

- Threat: another app attacks the user, or another app
- want to install third party app developer, but still want ot be safe
- app based model has 10x reduction in risk
- academia totally missed this, came from industry.

Question: why did academia miss apps as new architecture?

- We need to define "apps". Every website is an app in some ways.
- This is good from business perspective. Apple can cut sales from app store.
- hard paper to write, take a lot of engineering; evaluation is hard to write.
  - Academia wants to patch the system. How do we secure legacy apps. In stead of how to come up with an ideal system?
- Student question: Why do they want to build such system? Is is too risky? Motivation is on web. Place too much burden on developers - apps can be vulnerable.
- Windows has app store and third party, and start to support sandboxing.

Question: What is the challenges of retrofitting the apps on the platform?

- backward compatibility. Access files in their own directory(hurt why we use desktop, sandboxing make it difficult to program).

- Example: Photoshop, take a photo, and take into the photoshop.
    - need to have access files across different apps.

Question: suppose we are going to share some files across apps?

- Trusted UI. Use file picker to pick the files.
- open the app in some app and open it in another app. (Open photo in camera app, open in photoshop)

< Discussion on whether we should webcam >

< most people agree to open webcam >

< discussion format, raise hand vs just say it, some suggest moderating >

Question: When Android was a baby, do people propose any other systems of permissions?

Not really.

Abhishek: models that are (...)

capability based security. In berkeley, we have data capsules.

Android: users are in control. In papers, how users manage the permissions.

Usability

- **Bias**: Design for ourselves vs design for everyday-person
    - security person: understand what it means, this is good
    - average person: don't understand
- myers brigg personality test: e.g. sensing vs intuition, thinking vs feeling.

Question: What do we counteract this bias

- experiment on real people. Do user studies, select average people, representative. Do a lot of surveys.
- Craiglist(learning the attitudes) + app(survey a lot of people, low overhead, scalable) in the paper is very helpful. We can adapt techniques in HCI studies.
- start from low fi user studies. Hallway study(design product and find pitfalls)

User attention as a scarce resource

- cry wolf: seen notification very often, so don't pay attention.
- e.g. workers in enterprise: 23 times a day to log in the system, prove identity, single sign on. Ask a lot people to do with security.
- blame transfer: developer not sure what to do, so let the user to decide.

Question: Can we only ask when absolutely necessary or when risk is large, reasonable defaults

- It is hard to define what reasonable is - we are biased.
- Some may think it is secure, but others don't. For example, gyroscope, attacker can find the password by gyroscope. Security researchers are at a better position than end user to understand
- User want to get from point A to point B without caring consequences.
- Prof: different people have different concerns/ values, what is right for individual users. Some users very concerned. System developer cannot anticipate different users.
- use machine learning to predict what users want the app to do
- Implications: app collects location - what does the app learn from you? Average people don't aware that it is collecting home / work address - don't just what technique, but also implications
    - How you phrase those questions? Users are led by the questions.

Discussion

Launch the useable security question:

Question: How do you do usable email encryption?

Task: sign a key pair and encrypt the email. 1/3 is able to do that, several people send private key to another organization(they don't understand). Document was written by cryptographers.

Question: Totally automatic. Users don't have to be involved

- PGP: have to encrypt: right public key etc.
- HTTPS overlay, web encryption. (Central authority verify keys) - still more complicated
- Key generation is based on some email address. Identity based encryption: public key and private key are email address,

Question: What if I don't want central authority?

- key transparency: decentralized, multiple authority
- Make a compromise, such as SSH, trust-and-use verification
  - However, the email is different from SSH. Email is one-time use, fire-and-forget, SSH is connection based, remember for next time.
  - Mail client: create a new account, create a pair account automatically, include the public key in the email. When other clients see the email, remember the key.
- Student question: what is our threat model: there are design space, we can do HTTPS between major mail servers. e.g. Gmail and Yahoo

A transcript from the zoom chat: Anonymized the names:

- BTW was there a canonical answer to why did academia miss this?
- I actually had a very subjective thought which is that I never click on in-app ads, so their method of using Admob to find users could have been problematic
- I don't either, but maybe that's are bias :)
- Not directly related to apps and permissions but there are some companies proposing password-less authentication.
  BeyondIdentity: https://www.beyondidentity.com/
- An example for when computer scientists don't understand when risk is large: in one of the papers, volume control is given as an "easily revertible" permission that should be automatically given. But for a visually-impaired user, a change in volume could render the device unusable
- (Actually this is just what David just said, "not all users are the same")