CS 261 Scribe Notes: Usable Security

James Bartusek

April 7, 2021

1. Authentication

Class discussion: Shortcomings of passwords.

- People have many accounts, and it is hard to remember all passwords.
  - You could use the same password for all your accounts, but now your security is only as strong as your most vulnerable account.
  - Typical users have a few passwords, including one complicated "high security" password for banks, etc., and maybe some easy to remember low security ones that they use for everything else. The average is about 6 passwords.
- Passwords are often easy to guess.
  - A study of 70 million yahoo mail accounts ranked passwords based on most commonly used, which gives a measure of how many guesses an attacker would have to make to break into your account. It was found that 1 percent of accounts use one of the 10 most frequently occurring passwords, and 50 percent of accounts use one of the 1,000,000 most frequently occurring passwords (20 bits of entropy).
  - Some mitigations: i) requirements about special symbols to include in passwords, ii) preventing the use of dictionary words, iii) keep track of statistics on other users' passwords and don't let users use common passwords, iv) limit number of guesses, or start requiring CAPTCHA after a few attempts, v) don't store passwords in the clear, hash instead (but attackers can still compare hashes).
  - Password managers can help, but 1) users believe that password managers are less secure, and 2) users are worried about availability / functionality, i.e. that they would forget their master password and lose access to all accounts.
  - A Cambridge study on types of password instructions found the following. For users that were instructed to make a password with at least 7 characters and 1 number, 32 percent were hacked. For users that were instructed to create password based on mnemonics, 6 percent were hacked.

What are some alternatives to passwords? Something to keep in mind is that overuse of passwords condition people to enter their private passwords into any reasonable looking form, which is easy for attackers to exploit.

- "Remember me" feature. This somewhat mitigates the issue above, since users will only enter their password once per website. However this can result in forgetting passwords that you never need to enter.
- Email a link for verification.
  - Indeed, some people make use of "forgot my password" whenever they log in so that they don't have to remember their password. One potential issue is that email is not necessarily secure - standard protocols between mail servers are not encrypted/authenticated, though if two mail servers support encryption they can agree to encrypt. However, taking advantage of this and eavesdropping on internet connections is not usually easy to do.

- Link verification is especially helpful against phishing because attackers have to get someone to do something other than their normal behavior.
- Risk factors for clicking on link: i) vulnerability on browser exploited by website (less common now), ii) trusting the website and entering private information.

2. Warnings and Indicators

- Habituation is a significant issue. People may begin to notice false positives and then never heed future warnings, similar to justifying not wearing a seatbelt because nothing bad happened a few times. Also, warnings can interfere with people getting their work done, which encourages them to ignore.
- Say you have a device that you want to pair (e.g. over bluetooth) with your laptop. A strawman solution: First, devices exchange public keys. Then, a message pops up on your computer with MAC address of the other device asking for approve or deny. If approve, public keys do key exchange to set up the key. How good is this design? Can you come up with a better design? (breakout room discussion)
- Possible improvement: instead of MAC address that people may not check, generate a 4 digit code on one device, and enter it on the other device.

3. Discussion by Orr: SafeSlinger [FLK+13]

- Say that a group of users would like to share their public keys with each other and set up a secure group communication channel. What could go wrong? If they communicate via the cloud, an adversary could impersonate users and gain access to the communication channel.
- What if all users are in the same room? Bluetooth is slow and other potential solutions such as NFC are not widely supported, so the best option may still be to use an untrusted cloud.
- Local Group Key Exchange goals: security, usability, scalability, portability. The authors design an app that accomplishes these goals.
- Protocol:
  - Specify the number of users in your group.
  - The cloud generates a random id for each user and sends to each device. The users compare and find the minimum id, which they each return to the cloud.
  - A group Diffie-Hellman key exchange is performed, after which everybody in the group should share a secret key. This key is hashed to a three word phrase, which is returned to each user (along with two other random options).
  - The users determine the phrase that is shared among all their devices and they each select it.
- Analysis:
  - Why specify group size? So that the server or anyone else can't sit and listen in.
  - Why identify the group? To make sure other people using the app in another location aren't substituted into the group.
  - Why verify with the three word phrase? To prevent tampering / man-in-the-middle attacks.

References

[FLK+13] Michael Farb, Yue-Hsun Lin, Tiffany Kim, Jonathan McCune, and Adrian Perrig. Safeslinger: Easy-to-use and secure public-key exchange. 09 2013.