10/17/12: Usable Security

Antonio Lupher

October 24, 2012

1 Paper: "So Long, and No Thanks for the Externalities"

1.1 User Attention

- Think of user attention as a limited resource: users have a limited budget for security and compliance.
  - User attention is a super-precious resource.
  - Attention is often more limited than other resources we are used to thinking of as being limited (e.g. computing power)

1.2 Excessive Warnings

- Excessive security warnings, dialog boxes and prompts sometimes are used as a blame transfer by programmers, transferring blame to users.
- Security antipattern: You can often infer from the dialog boxes where the points of disagreements were in the programming team.They couldn't decide what the right thing to do was, so the compromise was to give the user the option to choose.

1.3 Security as Vaccination

- If enough people do it (institute security best practices), the entire community benefits.
- Phishing example: if enough people report sites, they are shut down, reducing profitability.
- Free-riding: If most people use best practices, then the few who aren't using them still benefit.
- What about mobile app security?
  - Research shows that only a fraction of people ever look at warnings and even fewer understand what's being said.
  - Obscure/ambiguous messages make things worse, e.g., what's a "sensitive log"?
  - Apps don't get to explain why they are requesting certain permissions (iOS 6 does give limited explanations, though).
  - If a small percent of power users see warnings, then maybe they can exert some back pressure to make it harder for malware or grayware to be used. Everyone benefits from their negative reviews.

1.4 Security Economics

- Is the authors' economic model a good method of predicting user behavior? Is it right to assume that users will make a cost/benefit analysis and decide whether to act based on that?
- Social effects: I might not worry about phishing unless I've been phished in the past or know someone who has been phished. Does user behavior change based on awareness of the problem and its consequences?
- Users might behave rationally (in line with cost benefit analysis) but for irrational reasons.
- Some research on mobile phones shows that when interviewing people about their concerns, one of the common concerns was loss/theft of device and access to data. This was highly correlated to experience (people who had lost a phone or knew someone else who had lost one were very worried).
- Be skeptical that users act in line with cost/benefit analysis. A lot of psychology research shows that people don't always act rationally. However, if we're giving advice that is economically irrational, that's bad.

1.5 Other reasons that explain low rate of security technology uptake

- Risk bias: Psychology literature shows that there is a known cognitive bias that affects how we think about risk.
- Game example: Player can choose from 2 games:
  - Game 1: win $500 every time
  - Game 2: flip a coin; heads = win $1000, tails = win $0
- The expected value of both games is the same ($500).
- Researchers asked a large segment of the population: Which game would you play?
  - More people tend to prefer Game 1: maybe there is a nonlinear utility function? Preference is for lock-in of sure gain.
- Follow-up question: when rephrased in terms of loss, e.g. sure loss vs. gamble for chance of no loss, people prefer to gamble for the chance of not losing anything. People lock in sure gains, but prefer to gamble about losses.
  - Preference for low variance for gains, high variance for losses = chance not to lose anything. If you're going to lose something, might as well lose big.
- Security parallel: if I don't click on the link, that's a sure loss. If I click on a link, that's the gamble. There's a chance that everything will be okay, but there's a chance that I can lose everything.
  - Possible objection: computer illiteracy. People don't know what they"re being warned about.

- Compliance: a security technique gets used even more than rational cost/benefit analysis would suggest due to compliance requirements: auditing, security controls that demonstrate that logs can't be compromised, etc. - penalties include not just fines, but executive-level prosecution, which drives more security requirements
  - Illogical policies like changing password every 90 days. This has no basis in security: if someone is guessing, changing the password won't change the expected time to success. This essentially forces users to pick weaker passwords, write down, etc., which leads to weaker security.
  - However, changing password periodically can prevent users from using the same password across websites.
- What does this mean for security researchers? Why bother with security?
  - Economic incentives: bank costs/losses are significant. Security research is valuable because the banks need it. We need to strike a balance, though, because too much security can reduce use and profits.
  - Note that fixing bugs and security vulnerabilities doesn't cost user time.
  - We should take into account these suggestions and costs in our research. Need to be measuring this in addition to performance cost, etc. Where possible, make security automatic so that the user doesn't have to worry about it.
  - In cases of cyberwarfare, cyberespionage and state-sponsored security initiatives, the economic analysis doesn't apply.
  - There exist non-monetary losses as well: loss of confidential data, privacy, etc.

2 Case Studies

2.1 SSH: Assess the usable security of SSH

- Users who use SSH are typically more computer-savvy (at least they know what a key and host are), so maybe they will be more careful?
- Review of SSH scheme: If user SSH's to a host that he/she hasn't connected to before, SSH displays a warning about not having connected before. The user can inspect the key, add the key to a file, etc. If a key for this host is saved, but the key provided by the host is different, the user gets a very loud warning.
- How secure is this scheme against man-in-the-middle attacks?
  - The user doesn't usually check the host's public key through out-of-bound checking.
  - No security against MITM on the first use, but good security on all other connections.
  - Design decision: SSH could give user a loud warning any time that the public key can't be verified.
    - Users would become accustomed to loud warning and would fall for a real MITM attack.
- SSH approach to key management is Trust on First Use (Key Continuity Management).

2.2 Email Encryption

- Consider applying key continuity management to email security:
  - Every email sent will contain public key as header
  - If recipient hasn't seen email from sender with public key before, the key will be added to a database
  - On receiving subsequent emails, the recipient will check that the emails were signed with public key (an attacker would not be able to send a valid signature).
- Doesn't work for phishing emails if it's the first time that the user has received email from that address.
- Compare to security on browsers: On a browser, you want to check that if you visited a website yesterday and again today it's still the same site. Need different warnings: one for "haven't connected before, here's the certificate", second: "already connected in past, but it's a different key
- Key continuity management seems to be pretty pragmatic strategy for SSH, but not for web? What's the difference?
  - Different use pattern connection to new websites is common, SSH to new servers less so
  - Could potentially do both: signed by CA, show messages

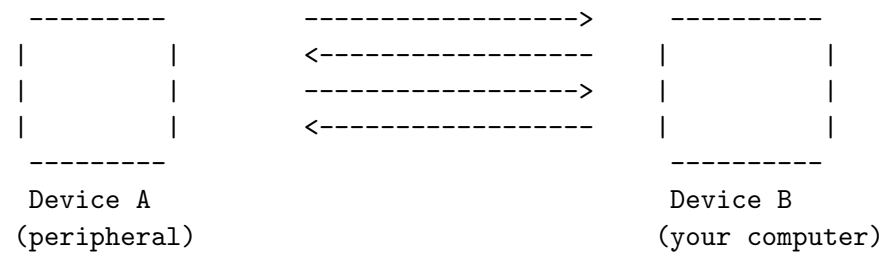
2.3 PGP: Web of Trust

- Key management in PGP for encrypted email uses web of trust concept:
  - "People whom I personally know and whose public key I have verified will then sign a certificate saying that another user's public key is real and verified."
- Led to "key-signing parties" at conferences: just show up with drivers license, then verify each others' keys
- Creates giant graph where path length can determine the level of trust for a key.
- Problems:
  - One rogue node can impact everyone's security by signing bogus certificates.
  - Not very resilient: very fragile and complicated.
  - Trust is not transitive. There's a difference between saying that you have verified someone's key and trusting that another person has verified someone's key.
    - Airport security example: On a San Francisco - Chicago - Washington D.C. flight you don't have to go through TSA again during layovers. Likewise, if you started at a small airport instead of SFO, you still wouldn't have to re-screen. Security for everyone depends on weakest link

2.4 Reformatting a Hard Drive

- In one study, researchers bought 158 used hard drives on eBay. It turned out that 12 were fully wiped, while 146 had some data or metadata that had not been wiped.
  - One drive had been in an ATM and had bank account numbers, spreadsheets, etc.!
- Windows has a scary prompt for reformatting hard drives, but it only overwrites the metadata (superblock, etc.). The contents of the files are still on the disk. Even if the files were manually deleted, the files aren't automatically overwritten.
  - This is done for performance reasons: it takes several hours to write data to overwrite data on disk. DoD wipes take even longer (have to overwrite several times).
  - This is a usability failure, since users expected data to be gone, but it persisted.
- How do we make this better?
  - Add a secure reformat option and notify the user ahead of time if data won't be wiped.
  - Scrub file on deletion: but this would be slower than status quo.
  - Launch a background process that periodically scrubs files that have been deleted (zeroes them out).
  - Encrypt individual files, it's enough to just wipe the key to delete the file.
  - Full disk encryption: delete the key and it's all gone.
  - Extra option: secure delete of file. While this might be slower, user can select whether or not to do this (already available on Mac).

2.5 Device Pairing



- What if there is a MITM attack when a pairing between a device and your computer first happens? (It can hinder key exchange)
- One possible crypto approach:
  - Send public keys, do key exchange, etc.
  - Each takes a hash of all the exchanges, if there is no MITM, then the hashes will be the same
  - If there is a MITM, there will be different messages, i.e. a different hash.
  - Provide a LED on device to indicate hash
  - On computer, show a "Do you see FX2982 on the device", "yes/no" dialog
  - The risk is that users will just click yes

- Better: ask users to fill in the code that they see (i.e. they are protected from laziness)
  - But users not happy to type in the code
- Best: offer a multiple choice dialog: 2 different codes, none of the above.
  - Only 1 in X options that a random click will protect them
- Use other channels to transmit information: e.g., get the key to a wifi network at Starbucks by audio processing on the signal sent over the sound system.

2.6 User Education

- Maybe we just need to educate users better? Train them about risks?
- Are users lazy? Or maybe it's our fault for not creating systems that are really easy to use for users.
- Sometimes education doesn't have the effect you'd think it would:
  - In a study on phishing training sites, researchers trained users on how to detect a phishing attack. They recruited subjects, gave them simulated websites, asked them to say which ones were real, then gave training, and finally tested again. This way, they could measure the correlation between the true number of phishing sites and number of sites that people actually thought were phishing sites.
    - Before training, the number was uncorrelated.
    - After training, the number of suspected phishing sites was still uncorrelated to actual phishing sites, but users reported that twice as many sites were phishing sites! Users were more scared.
  - This is an example of counter-productive education.