Usable Security
November 2, 2011

1 SSL Certificates

Building from the prior lecture, the general theme is to suggest that there are examples of usable security done right. Last paper provides evidence that browser changes are valuable. In the past, it was easy to dismiss dialog boxes despite their dire messages. After the studies, it is evident that in-your-face messages that interrupt workflow are effective. A good example of such a message is the invalid SSL certificate warning where before, invalid certs produced an easy-to-dismiss dialog whereas now, it produces an interstitial page that requires much more effort to get past.

Food for thought: In Firefox 3.5, it became much harder to add exceptions. So much harder that a bug report was actually filed by a user who was compromised that had to keep adding exceptions for every site. Does this indicate better usability? Should the user have realized s/he had been compromised?

In providing usable security, why allow the user to click through to a suspected phishing site?

1. Risk of false alarms
2. Users dislike control being taken away from them
3. Its common for websites to change certificates – if the user knows better, s/he should be allowed to visit the site.

More generally, Adam Barth highlights that design should consider users over authors over implementers over specifiers over theoretical purity.[1]

1.1 Habituation

Furthermore, SSL certificates are prone to generating warnings. In a 2007 survey of all websites that accepted SSL certificates, 62% had certs that would trigger browser warnings (but the majority of those were unused sites). Of the top one million, 44% generated warnings, and of the top one thousand, 19% generated warnings. Despite this percentage going down for websites with higher Alexa ranking, it strongly implies mismanagement of SSL certificates.

This risks user habituation since general web browsing could present far too many errors that makes it difficult for the user to distinguish between true warnings and poorly managed SSL certs. Consider the scenario presented in the paper where Firefox 2s warning system blocked 100% of phishing attacks whereas IEs warning system only blocked half of them with the key differentiating factor being that Firefoxs warnings looked different from other warnings whereas IEs warnings looked similar to other warnings the users might have been already accustomed to. In the latter case, users tend to consider it a false alarm. This suggests that warning mechanisms work when they are triggered only when appropriate. Ultimately, the lesson is that if multiple warnings with different severity look the same, then habituation presents a large threat to user safety.

1.2 Discussion

Perhaps warning systems can utilize two different black lists: one list of sites proven to be malicious which get blocked without user input and another where the site is malicious with high probability, but it provides an option to the user to click through. This finer granularity and less choices could be more effective in protecting the user.

Challenges Whats an appropriate way to keep those two lists up-to-date and what are the repercussions of an outdated list?

Is there a canonical way to distinguish between severity of warning messages? The Chrome team classifies errors by scope of the compromise between

1. malicious code executes are user
2. complete browser compromise
3. compromise effects one website

2 Usable Security Interfaces

2.1 File Deletion

Motivating study: Researchers purchased around 150 hard disk drives from eBay and imaged them for data. The assumption was that while most people reformat their hard drive, they inaccurately assume that this process destroys their data. Truth is, reformatting only deletes the metadata while the actual files still exist on disk. Unsurprisingly, the researchers were able to retrieve data. 12 out of 158 hard drives were properly sanitized while the rest had not. Most interestingly, one hard drive was used in an ATM and the researchers found account numbers which, needless to say, is bad. Independently, NYTimes conducted a similar test with used copiers and found that the copiers contained a hard drive which recorded every image ever scanned – including sensitive documents.

2.2 Why did this happen?

Simply, the interface is misleading. Move to trash is not informative enough while messages such as All files will permanently be erased are interpreted as a promise rather than a warning. This suggests a redesign of the interface. But how?

Ideas

1. "format" actually write random bits (which is very slow and raises usability issues)
2. full drive encryption is default (also has usability concerns)
3. consider delete vs. secure delete
4. provide information about formatting with the drive
5. change the warning message presented during formatting (for example, instead of suggesting that formatting will delete all data, it could say data "may be deleted but can be recoverable by motivated people")

Of these ideas, 1 gained from traction with hardware supporting ATA secure erasure[2]. However, 3 may be the most effective because the option for a true deletion of data changes the mental model of how file deletion works and make the user think about security rather than just the task at hand.

2.3 Bluetooth Pairing

Consider a classic scenario where a Bluetooth device is to be paired with a host device. How would this pairing be done securely given the threat of a man-in-the-middle attack that could force the host and/or the peripheral to pair with a device of the attackers choosing and intercept/transmit data.

- **Approach 1** Initiate communication, client sends public key, host sends public key, generate session key (key exchange). Shortcomings Fails if attacker is present during initial communication (attacker can intercept the keys and control the session key!)
- **Approach 2** Use an out-of-band mechanism such as PIN input where the PINs are hash values of the data communicated between devices. This approach works if, out of the two devices, one at least has input while the other at least has output and there is a human operator verifying the result. For example, given 4 hashes, ask the user to pick the right value. This interrupts the user from just clicking OK at every dialog and encourages the user to pay attention. Shortcomings It may be too farfetched to assume that the user will pay attention to the entire hash value which allows the attacker to manipulate the options to present a number that looks similar to the real hash value. More generally, options need to be carefully chosen. As such, an option such as "none of the above" is dangerous territory if it were ever to be a valid choice since that leaves too much room for attackers to conduct successful attacks.
- **Approach 3** In-band link-layer approaches.[3]

2.4 Approaches to establishing security

- Bump: record smartphones accelerometers readings to pair them [4] (readings arent unique enough – attackers can replicate motions)
- Facebook: show pictures of friends (profile pictures arent always recognizable)
- Gmail: back-up email address, SMS, helpdesk that asks questions that can be computed from Googles end based on your email corpus (back-up emails are unreliable, SMS number may not be present, trying to guess what the user knows is prone to failure)
- Bank of America: SiteKey: user registers an image to be associated with their account and that picture is presented to ensure that the webpage isnt a phishing site (MITM is still possible as long as the phisher is prepared to ask the user security questions, 90% of the users log in anyway despite the lack of SiteKey image)[5]
- Video games: asking the user to describe their in-game avatar.

Shortcomings A lot of verification questions are public knowledge! Case in point: Sarah Palins email was hacked using known information.[6] Prior class project composed a set of verification questions and corresponding answers to find that the entropy of answers is far too low. Approaches like SiteKey are feel good mechanisms since they dont actually provide anything beyond the illusion of security. Studies have found that sites that utilize systems like SiteKey observed increase in fraud rates, but the companies are happy with maintaining the addition since it makes customers feel safe.

As a final note, it would seem that the most effective method of ensuring secure transactions is to utilize a hardware device with a UI (like an uncompromised smartphone) that can give details of each transaction to protect against MITM attacks.

References

[1] `http://www.schemehostport.com/2011/10/priority-of-constituencies.html`

[2] `http://tinyapps.org/docs/wipe_drives_hdparm.html`

[3] http://www.usenix.org/events/sec11/tech/full_papers/Gollakota.pdf

[4] http://bu.mp/

[5] http://www.bankofamerica.com/privacy/index.cfm?template=sitekey

[6] http://www.wired.com/threatlevel/2008/09/palin-e-mail-ha/