CS 261 – Human Factors in Security and Usable Security

Scribed by: Sudeep Juvekar
Date: 26th October 2009

Administrivia:

1) Homework 2 is up on the site: **due on 11/09/2009** (in 2 weeks)
2) Project proposal **due on 10/30/2009** (this Friday)

Usable/User-centric Security

Building/designing systems with user behavior in mind. An area of huge impact in today's world! (Dave: "If I were a grad student, I would be working in this area!)

Common intuition for designers: User == clone of "me". Design a security policy/mechanism best suited for "me". This intuition is biased and wrong, because:

1. A system designer is a trained professional and is much more knowledgeable than an average user. Hence, the rules applying to a system designer might not hold for an average user.
2. Demographics: A system designer in a company/university exposes to an entirely different set of people (e.g. all individual in a same age group/with same interests) while designing the system. This is, again, not true for an average user of his/her system.

To correctly analyze how a user interacts with a system, one has to look at it from a point-of-view of user personality. **Myres-Briggs**[1] type indicator is a good way to look at these personalities. It classifies personalities in four categories.

1) **I/E (Introvert/Extravert):** Classification based on how a person extracts energy; external interaction (Extravert), vs internal reflection (Introvert)
2) **S/N (Sensing/iNtuition):** Perceiving the world or information gathering; using concrete facts (Sensing), vs more abstract reasoning (iNtuition)
3) **T/F (Thinking/Feeling):** Decision making; logically (Thinking), vs by empathy (Feeling)
4) **J/P (Judging/Perceiving):** Lifestyle, indicates whether the individual dominantly uses T/F (Judging) or S/N (Perceiving) functions.

A study has found that among computer scientists, the personality-types **TJ** are the most common of all. However, the rest of the population only has about **8%** of **TJ** personality types. This explains why the designers intuition might often be wrong and we need ways to counter the bias of the designer.

How to counter the designer bias?

1) **Domain experts/User representatives:** Ask user representatives about the design. Works best when designing systems for specific group of users (e.g. DARPA grant administrators)
2) **User testing/User studies:** A study of more broader class of users. Might help evaluating a prototype and suggesting features
3) **User customization:** Inferring Myres-Briggs personality types of users and customizing user interfaces according to them. Some systems do have different basic and expert user interfaces.
4) **Anthropology:** Study user population, understand user better
5) **Psychology:** Understand the behavior better, users have many cognitive biases; take them into account while designing the system.

Phishing:

To understand how phishing works, we must study a bit of human psychology. Studies have shown that humans are not completely rational beings. Our behavior is generally not governed by a Bayesian inference/decision making, where we compute the possible gains for every decision made at every decision point. Such a reasoning would be highly inefficient. Instead, our behavior is generally broken down into 3 levels:

1) **Skill-based:** Automatic actions, heavily practiced (e.g. walking)
2) **Knowledge-based reasoning:** Conscious/sophisticated reasoning, usually time-consuming
3) **Rule-based reasoning:** Intermediate between the two. Uses some heuristics (rules) based on past experiences while making a decision

Sometimes, more than one rule applies at a particular decision point. The brain uses a mechanism called **frequency gathering** to select between the alternatives. It weighs some rules higher than others based on past experience of success/failure using those rules. Higher the past success, more likely will the rule be used!

This user behavior might be exploited by phishing attacks as follows:

1) A phishing site might mimic some legitimate site for which the brain has established a success-rule. e.g. a phishing site might copy the visual pattern of a popular site, where user has successfully logged on in past.
2) **Dialog boxes:** Dialog boxes are usually ineffective against these attacks, since user has clicked on them in past and has discovered that their warnings are false-alarms in many cases. The user, thus develops a rule to ignore these boxes. Users also ignore these boxes since they do not understand them.

A good security design should incorporate these issues related to user behavior. Some suggestion are:

1) Making users learn rules favorable to the security policy
2) inferring the rules learned by users and using them to implement the policy suitable for user behavior

How phishing works: paper review

The user study presented in the paper[2] has some issues:

1) Users were clued that the study was conducted on phishing. This made them extra cautious while participating in the study. A common solution for this issue is deception: lying about the purpose of study (e.g. claiming the experiment to be designed for testing the use of browsers etc)
2) “Sense of risk”: There is an inherent dilemma to the test designer. The designer can not expose the user to any risk (e.g. asking their credit card details), but at the same time must make the user feel that she is at risk (simply asking the user to use a fake credit card number does not give him/her any sense of risk).
3) Size/distribution of user group: A very common shortcoming of user studies. Are the user groups representative of the overall users? Many security papers, for obvious reasons, use college students as test subjects. Most of the times, they are not representative of average users.
4) Lab vs. user's natural environment: Hard to know whether the fact that the experiments are conducted in a lab affect user's behavior.
5) Authority figures/demand effects: A user has a tendency to follow instructions from authority and is not acting “on his/her own” during experiments. There is also an inclination of a user for pleasing the authority/instructor. His/her response/behavior might be biased towards it during the experiment.

Past mistakes in designing secure systems:

1) **UAC:** Earlier versions of Windows allowed users to operate with Administrator privileges. Recently, in Vista, a user has limited privileges. When a task with administrator privilege has to be performed, it pops-up a window asking the user for administrator password. The check for the privilege happens in the kernel, at which point it loses track of all user action and source of the request. Due to this policy, user gets annoying pop-ups all the time! A solution would be using trusted paths, tracking down user actions to infer authorization.
2) Pop-ups in general during a user action are annoying and a user tends to ignore them. e.g. an “Are you sure” Pop-up while deleting a file generally annoys the users. A better alternative might be using the Mac's scheme of allowing users to recover from their possible mistake; i.e. To undo their delete.
3) Interrupting ordinary tasks/workflow of a user and asking a security

related question annoys them. Users tend to take short cuts and ignore these question. Many past systems designed to do this might thus fail to achieve their purpose.

4) **SSH/SSL pop-ups:** Pop-ups telling the users that the security certificate of the site is invalid/expired. People tend to ignore these pop-ups. Some studies have shown this to be a rational choice, since even legitimate sites do not update their security certificates. Some browsers are now changing their policies about these pop-ups. E.g., Firefox 1.0 used to display these pop-ups, Firefox 2.0 replaced the site by a page saying that the certificate is invalid, while, Firefox 3.0 uses self-signed certificates or sometimes displays 404-HTTP error. Self-signed certificates, however, are vulnerable to man-in-the-middle attacks. A possible solution to this problem might be using DNSSEC.
5) Bluetooth pairing: Before communicating with a computer, a bluetooth device must have the same crypto-key as the computer. Many device manufacturer, however, used 000...0 as a crypto-key for their devices, thereby leaving them vulnerable.

References:

[1] The Myres-Briggs type indicator: http://en.wikipedia.org/wiki/Myers-Briggs_Type_Indicator

[2] Why Phishing Works: http://people.seas.harvard.edu/%7Erachna/papers/why_phishing_works.pdf