Usable Privacy & Security I
Scribe Notes
February 7, 2006

Presentation by Colleen Koranda
Scribe Notes by Levi Broderick

Today's presentation covers Security and Usability, chapters 1, 2, 3, and 32.

The Need-to-Know Principle

How does the user contribute to the security of a system?

To security departments, the users are generally the weakest links in the chain forming any security system. They will subvert the system (whether intentionally or not) by writing passwords down on sticky notes attached to their monitors, by choosing an easily-guessed password such as the name of a spouse, and by falling victim to social engineering attacks. In short, the only thing predictable about users is that they generally cannot be trusted.

Security departments tend to inform users about security on a need-to-know basis. That is, the departments assume that the security of a system is undermined if users know too much about it. They hope that by limiting the weakest links' knowledge of the system, the system itself is made more secure.[1]

However, recent research suggests that this is a self-fulfilling prophecy. Users judge their own importance in the security chain to be minimal, so they see themselves as a low risk to any security mechanisms in place. As a result, they tend to be more lax about their own contribution to the security of the system. A self-fulfilling prophecy occurs: users are seen as the weakest link, thus they become the weakest link.

Educating Users

How can we make users aware of their behavior and its possible negative contributions to security?

---

1 *Security and Usability*, p. 643.

One approach to increasing user understanding is to simply educate the user. Consider what happens when a user is asked for his login ID and password. Many users are unaware of the difference between the two:

- **User ID:** *(identity)* Provides identification of a user to a system or to other users of that system. In most cases, a user's ID does not need to be kept secret.
- **Password:** *(authentication)* Verifies that the person providing the identity is actually the person to whom that identity belongs. A password must generally be kept hidden from everyone except the user whose identity the password is tied to.

Today's presentation focuses on passwords and the methods by which we can educate users about their proper selection and usage.

Activity I

Colleen provides a handout ("Initial Password Activity") that scores the care with which users select and manage passwords. Take the survey provided from the point of view of a person who uses passwords regularly. There are a few dozen statements in the survey, each one followed by a variable number of points.

1. Start with a zero score.
2. For each statement,
    a. if the statement applies to you, add to (or subtract from) your score the number of points indicated; or
    b. if the statement does not apply to you, leave your score unchanged.
3. Upon completion of the survey, you will end up with a score ranging from +16 to -78.

How did our class fare?

Nobody responded that he or she had a positive score after all statements were evaluated. Most of the class had scores between -10 and -20, and a handful had scores worse than -20. It was brought up that the questions asked cast a bit of a wide net and that there was no reasonable expectation that a person could satisfy all of these points. However, these are the methods that password cracking programs use to break into user accounts, and computers do excel at repetitive and time-consuming tasks.

Are all of these points necessary for security, though? Perhaps there are times that we want people to have access to our accounts, so as a result we need to choose easily-remembered passwords. Computer repair technicians need to know the administrative password to a system in order to perform their duty, though this certainly violates some of the points listed in the handout. The result of this is that password security is in practice often subjective based on what is being protected, the need for others to access the system, and ease of use for the authorized user. We have certainly all been guilty of bad practices at some point.

Choosing Good Passwords

We want users to choose passwords that they are not likely to forget, but we want these passwords to be difficult for an attacker to guess. Providing instant feedback to users during the password selection process is crucial for assurance that this is done properly. A user needs to know why his choice of password is insecure as part of the education process.

One web-based utility[2] for measuring a password's strength does just this; it also provides some tips for increasing the strength of a password. For example, entering the word *apple* into the password entry form produces the following response:

- Your password was found in our dictionary, please read on "Selecting a good password" and try again!
- Increasing the length of your password to 8 or more chatacters [sic].
- Using uppercase chracters [sic] in your password.
- Using special characters in your password.
- Using numbers in your password.

Of course, it would be nice if users could reliably memorize complex sequences of random characters, such as *qOv"$:*zoq(C{5]z* (equivalent to a 106-bit key). This is not a reasonable expectation. What about hex sequences such as *CB4B3363B6921EFB* (64-bit)? Again, it is likely that users will have a difficult time remembering such sequences.

Recently, word lists have been used to translate random character sequences into random word sequences, operating on the theory that lists of words are easier for users to remember than

---

[2] http://www.securitystats.com/tools/password.php

sequences of random characters. The biometric word list[3] used by PGP, for example, translates the random hex number *20296BFE380DD2* (56-bit) into the word sequence *bison certify glitter yesteryear classic asteroid standard*.[4]

Diceware[5] uses a similar principle to translate consecutive rolls of a die into a random passphrase. For example, if twenty consecutive rolls (52-bit) produce the sequence *56342-11561-46253-66444*, Diceware produces the phrase *tacit alger portia 41st.*

Then again, users don't necessarily have to explicitly memorize passwords. Other methods call for passwords to be *reconstructed* from recalled words or phrases. As an example, Colleen takes the first letter from each word of the phrase "I've fallen, and I can't get up" to create the password *If,aIcgu*. Other examples can be seen in slide 8 of today's presentation.

The goal of all of these methods is to reduce the possible attacks against a password to a brute-force search. If the password is sufficiently long, a brute-force search is infeasible. Assuming that an attacker can perform 100,000 encryption operations per second, a password consisting of only lowercase letters with a 6-character length (28-bit) takes under an hour to crack using brute-force. On the other hand, a password consisting of any displayable ASCII character with a 12-character length (79-bit) would take 150 million millennia to crack. This timespan is around ten times the current estimated age of the universe.[6]

Motivating Users

Unfortunately, education by itself cannot change user behavior. Users who do not follow security policies often share beliefs or attitudes.[7] Among these:

- Users do not believe that they are personally at risk.
- Users do not believe they will be held accountable for not following security regulations.
- The behavior required by security mechanisms conflicts with social norms.

---

3 ftp://ftp.pgpi.org/pub/pgp/7.0/docs/english/PGPWinUsersGuide.pdf

4 http://en.wikipedia.org/w/index.php?title=Biometric_word_list&oldid=33350930

5 http://www.diceware.com/

6 http://map.gsfc.nasa.gov/m_uni/uni_101age.html

7 *Security and Usability*, p. 27.

- The behavior required by security mechanisms conflicts with self-image.

Another approach to getting users to understand and to follow security policy involves getting users more motivated about the policy. Users' motivation can be improved if they both care about the data being protected and understand how their current behaviors put them at risk. To accomplish this, security departments must involve the users more and must have a more active appearance and role in the organization.

Usable Systems

One way of motivating users is to design a usable system. Since security systems involve users so heavily, they should take the users into account as they are being designed. For example, a system that assigns users passwords of random characters monthly might seem good on its face from a theoretical viewpoint. However, considering the discussion earlier, users cannot possibly be expected to reliably memorize these. This has a spiraling effect:

> *I cannot remember my password. I have to write it down. Everyone knows it's on a Post-it in my drawer, so I might as well stick it on the screen and tell everyone who wants to know.*

Sasse and Flechais describe that the security of any sociotechnical system is the result of three elements: product, process, and panorama.[8] How, then, do we relate these to password security?

Product

The actual security mechanism of a system.

Is a password meaningful to you?

- **Pros:** Meaningful passwords are often easier to remember. Since the password is so easy to recall, the user is less likely to write it down or to forget it.
- **Cons:** If the password is meaningful to you, chances are good that an attacker understands that it's meaningful to you and will guess it on one of his first login attempts.

Does the password change regularly?

- **Pros:** The attacker has a moving target. He only has a limited time in which to discover

---

[8] *Security and Usability*, ch. 2.

your password and perform his attack before he has to start over.

- **Cons:** These types of passwords are harder for users to recall.
- NIST provides guidelines[9] for choosing the maximum lifetime of users' passwords.

Is the password system-generated?

- **Pros:** An unbiased and random system can generate passwords that are inherently more secure.
- **Cons:** These types of passwords are harder for users to recall.

Is this password used frequently?

- **Pros:** It is more likely that the user has committed it to memory.
- **Cons:** These types of passwords are harder for users to recall.

Process

How are security decisions made?

Security tasks must be designed to support production tasks. The AEGIS process can be helpful when designing the security of a system. Even if it is not used, it is imperative that users are involved in the design of a security system. This makes the system more accessible and more personal to them, giving them a vested interest in doing their part to maintain the security of the system.

This is no different with passwords. Consider a system that uses two-factor authentication – our system will use a monthly-changing password and a physical token. Imagine that the designers and users of this system are worried about forgetting their password, but they agree that it is unlikely that the physical tokens will be lost. (Perhaps the token plays auxiliary roles, and thus the user would know quickly were his token stolen.) In such a case, it might be better to use a shorter, more memorable password (like a PIN), placing more security in the fact that users are unlikely to be separated from their tokens.

Panorama

Take the context and the environment into account.

---

[9] http://www.itl.nist.gov/fipspubs/fip112.htm

With the two-factor authentication example provided above, it is still important that a user not write down his password. We try to mitigate this by choosing a type of password (in this case, a short PIN) that is able to be committed quickly to memory.

User education and training is crucial. We need to teach and to reinforce proper behavior through drills, monitoring, and feedback. Additionally, we need to involve more people than those who operate systems deemed at-risk. By including all staff, we create a more complete, more inclusive environment in which users make security decisions. We combat the notion that security is limited only to "nerdy" and "paranoid" people by highlighting role models. This way, users feel better about making wise security decisions.

Activity II

Colleen presents two scenarios in which security needs must be addressed. Each has its own requirements and limitations, so a method which excels under one scenario may not even apply to the other. Work with a small group of people to discuss each environment and possible approaches for addressing its specific needs.

Keep in mind a few notes:

- Users excel at getting around subverting security systems. We want users to comply with the security systems. One method for doing this is to make it difficult or undesirable for a user to take a course of action other than those recommended by the security system.
- The previous bullet implies that simply educating users about proper usage of the security system is not enough.
- Note the pros and cons of each decision you make (or reject).
- It might be helpful to refer to the design checklist in *Security and Usability*, p. 42.

How did our class respond?

For the second scenario, the class suggestions involved the use of two-factor authentication. It is very likely that workers in a hospital environment carry identification badges, so those can serve as encoded tokens with which to access a terminal. These badges can be augmented with passwords and PINs to provide a good balance of security and ease of use. For example, the user can swipe his identification and input a PIN in a very short amount of time, and this can serve as the usual login mechanism.

One problem with this is that badges can be stolen, and attackers can probably discover a user's PIN through something as simple as subtly looking over one's shoulder while the system is in use. The groups determined that a badge is most likely to be lost while the user is on break or if the user accidentally leaves his card in the terminal. To solve the first problem, the system can demand a password in place of a PIN when it detects that the user had not logged in for some time or is scheduled to be on break at the moment.

The second problem can be solved through a combination of methods. We realized that since the user has to prominently display his badge while at work, it should probably be physically attached to him. Imagine that the badge is at waist-level and attached by a clip and a retractable cord. Then the slot into which to insert the badge can also be at waist level, and it can hold the card (and physically keep the user at the workstation) until the user has opted to logout, at which point the card will be released and the user is free to go.

As for the password itself, some suggestions revolved around basing it on a medical term. For example, the hospital might feature a "disease of the month" and base the password on something related to that disease, such as the name of a medication which might be used to combat it. This particular example is probably not a good solution, but the general idea is to have something that doctors could easily remember while forcing most attackers to stand at the terminal with a bulky medical reference text. This would hopefully delay the attacker long enough for him to be discovered before he is able to login to the system.

Other suggestions include:

- If there has been a long stretch of inactivity at a terminal, force the next user to enter his password.
- Have the terminal displays recessed into whatever face they're in, making it difficult to read the screen unless you're standing directly in front of it. Similarly, recess the keyboard so that others standing around are less likely to discover passwords or PINs simply by looking.