COMP 5407 Project 2: The way security really works: How human factors can be used to improve security

Terri Oda
terri@zone12.com
Carleton University

December 5, 2003

1 Introduction

Even the strongest security system can be broken using a sticky note. The first step is simple: a legitimate user writes down his or her password on the sticky note and leaves it in a convenient location, such as the top drawer in their desk, or even stuck to his or her computer screen. The next step is also simple: any attacker can then walk by and read the note. Once a legitimate password has been obtained, it can be fairly easy to gain access to a system.

All the rigorously-checked cryptography in the world can't help if a legitimate user discloses a password, yet many users write down their passwords and paste these notes to a computer screen for easy access. Sure, it makes things easy for the user, but it's also easy for anyone else in the building. Sometimes an attacker may not even need to gain access to a building – walking by outside and looking through the window is sufficient.

Nielsen makes a bold but probably not unreasonable claim:

> "Take a walk around any office in the world and you can collect as many passwords as you like by
>
> - looking at the yellow stickies pasted onto terminals,
> - finding the cheat sheet in the user's top drawer, or
> - searching the user's hard disk for the file that inevitably contains all required passwords in one conveniently machine-readable spot." [Nielsen, 2000]

Now, it's easy to blame the user. But placing blame doesn't make things more secure. And is the blame even in the right place? Patrick mentions that the "human error" approach was abandoned by airlines with good results: now that planes and related equipment are designed with human needs and limitations in mind, flight safety has greatly improved [Patrick, 2002].

Norman's successful book, *The Design of Everyday Things* [Norman, 1988] discusses the role of poor design in causing what many people consider to be human error. Many of his theories have been applied to the design of many “things,” but for some reason, so-called secure systems often ignore the design rules he proposed and administrators of such systems continue to blame the user for design failures.

By moving on from assuming users to be the source of problems, security experts can begin to look at the real causes of error. Technology can play a large role in preventing errors, but it can also play a large role in forcing them.

Unless system designers understand how users actually use systems, they risk encouraging users to make insecure choices, such as disclosing passwords, by enforcing the very rules intended to increase security. Section 2 discusses how while some password rules may theoretically make for better security, the reality is a very different story.

The perception is often that usability and security are opposing goals rather than needs that must both be taken into account. A quote attributed to Oliver Elphick goes, “Make it idiot-proof, and someone will breed a better idiot.” This is probably the way in which many security experts see their security-breaching users. But the users see it quite differently: The system was getting in the way of their work, so they just worked around it to ensure that more important things got done. Section 3 looks at user considerations that can lead to users making insecure choices.

It’s all fine to point a finger and say there’s a problem, but without a solution we are not necessarily further ahead. Thankfully, work has begun on finding ways to design systems to be both usable and secure. Some of these are summarized in Section 4.

2 How can “good security” rules cause bad security?

Adams and Sasse found one user whose comments clearly illustrated the relationship between rules intended to improve security and the compromise of security. Of his password, he said “... because I was forced into changing it every month I had to write it down.” [Adams and Sasse, 1999] Many users make insecure choices when presented with rules designed to make things more secure.

Security rules are not restricted to passwords, either. Consider cars: it used to be that cars could be hot-wired once someone had managed to get into the car, perhaps by breaking a window or forcing the lock on the door. But in order to make them harder to steal (that is, more secure) insurance companies and car manufacturers have come up with clever security devices. But with more security devices in place on higher end cars, car thieves have had to adapt. In some cases, that means fairly clever scams involving pretending to buy a car to get access to information about it. But other thieves bypass trying to disable the devices and resort to waiting until the owner returns to the car, keys in hand. [Tognazzini, 2003] This is a much more dangerous situation for the car owner! It may not be hyperbole to say that these “security” rules can even cost lives.

There are several commonly-used security rules that have unintended side effects as users find ways to cope with them. Nielsen claims that “security-enhancing” rules lead to one outcome: users write down their passwords [Nielsen, 2000]. However, even those who don’t write down passwords may be making poor security choices. This section looks at a few very common password rules intended to improve security, and discusses why they also cause security problems, and what those problems are.

2.1 Changing Passwords

Security Measure: Passwords must be changed regularly.

Security Goals: Password-guessing attacks such as the common dictionary attack are hindered by the "moving target" of the changing password. Dictionary attacks occur when an attacker attempts to guess the password using a "dictionary" of common passwords to make educated guesses. This can be compared to a brute force attack in which the attacker tries all possible passwords. Since dictionary attacks and brute force attacks requires either significant amounts of computing power or time, the idea is to change the password before an attacker can guess it.

There is also hope that this will reduce the risk of undetected security intrusions. In theory, any attacker who gains access will lose it when the password is changed.

Unfortunately, this seems unlikely to help in many cases. Although it is possible that changing the password after an intrusion has occurred will oust the attacker, it is usually trivial for an attacker to install a malicious code onto the system once he or she has gained access. This could be a "back door" so that an attacker can pass through without needing to enter using a regular user's password again, or something that logs all the keystrokes a user makes, including those involved in changing the password. As such, changing the password is not usually sufficient if an intrusion has occurred, and should not be counted upon to limit the damage from undetected intrusions.

User Problems: It is difficult to come up with so many memorable passwords. Typical schemes involve changing passwords every month, and it becomes easy to forget or mis-remember and try previous passwords instead of the current one.

User Solutions: The user writes the password down. Sometimes this copy is placed in a fairly safe place, but often it is simply placed somewhere convenient for the user. For example, it tends to be fairly hard for an attacker to get into a locked file cabinet, but if the password is left in an unlocked top drawer or under a mouse pad, it is fairly simple to find and acquire. On top of this, users may write down their usernames and the systems so that they don't forget which password goes with which system. This makes it even easier for an attacker to use the information.

Writing a password down does not always imply that it will be written on paper. Users may also store passwords on their computers, often in an easily-readable file on the Windows desktop or in their Unix home directory. This may be even more dangerous because gaining access to a desktop machine can be fairly easy: if viruses and spyware can get in, then there's no reason a more dangerous attacker couldn't. For some systems, all it would take is a malicious email to gain access to all of the user's files.

Users may choose passwords that are simpler, but also easier to guess. Some, like the name of a spouse, child, pet, etc. will be fairly trivial for an attacker who knows the user to guess. Some users will simply use their own name or initials, favourite sport or team, things related to hobbies, artists, movie stars, birthdays or other significant dates. Simple knowledge of the gender of the user can limit the selections further: TheNetworkAdministrator.com [Chick, 2003] claims that while women will use their wedding date or children's birthdays, men will rarely use those.

Other passwords are easily guessable even if the attacker doesn't know the user. In the movie Hackers [Moreu, 1995], they claimed that the four most commonly-used passwords were "love", “sex”, “secret”, and “god.” Real passwords may not be quite so simple (although those words do appear in many password dictionaries), but other common choices include “password” and the username.

Users make up schemes to make new passwords based on an older one. For example, a user might have the following sequence:

Month	Password
January	chipmunk1
February	chipmunk2
March	chipmunk3
...	...

Adams and Sasse note that while this would seem to increase memorability and thus security, in reality people often confuse the list (which would seldom be as nicely aligned with the months), which results in them having to write down their passwords [Adams and Sasse, 1999].

Even without that problem, none of these passwords is particularly hard to guess using a dictionary attack. If someone managed to find one password, it would be somewhat simple to guess what the next one would be, thus further limiting the effectiveness of changing passwords when it comes to stopping intruders who have already gained access.

2.2 Complex passwords

Security Measure: Passwords must be sufficiently complex. There are many common rules that may be required, such as the following:

- Password must contain at least one letter, number, and symbol.
- Password must be at least N characters long.
- Password cannot contain any dictionary words or proper names.

Security Goals: Dictionary attacks should be much more difficult because the dictionary of passwords used will be larger.

User Problems: Complex passwords are harder to remember, and easier to mix up. Users may mis-type the password more frequently even when they recall it correctly because the sequence of keystrokes is unusual, leading the user to believe that he or she has forgotten the password.

User Solutions: As before, users may write down their passwords.

Even if the rules are followed, only the bare minimum required by the rules may be followed. For example, if the user wants a password related to the word “squirrel” but they’re required to use at least one letter, number and symbol, they might chose the password “squirrel1!” Conveniently, the number one and the exclamation point are both on the same key, making it easier for a user to type and recall but also easier to guess.

Similarly, if the user wanted a password based on the word “gerbil” but was required to have a password of at least 8 characters, the user will likely just add something simple onto the end such as “gerbilll”.

Numbers, symbols or extra characters are most frequently are added to the beginning and end of passwords, and although those extra characters do increase the size of a dictionary, they do not necessarily make it large enough to render it infeasible for an attacker to break in using the expanded dictionary.

2.3 System-selected passwords

Security Measure: A system-selected password is assigned to the user. Usually, the user gets no choice input into how this password is created. These passwords are typically generated randomly by machine, and often have to follow special password rules as in Section 2.2.

Security Goals: Because it is known that users tend to select passwords within a small “dictionary” of possible passwords, we know we cannot trust users to come up with good, secure passwords. Assigning passwords from the entire set of possible passwords will make it harder for an attacker to guess passwords.

User Problems: These generated passwords are even more meaningless and thus are even harder for a user to remember.

Although many users don’t, users can choose secure but still meaningful passwords. For example, someone might choose the password “`~(k1)^2Frt`” which looks fairly meaningless, but makes sense to the user:

`~` is a tilde – a squiggle that reminds the user of her first pet, a ferret
`(k1)^2` represents the ferret’s name, which is Kiki.
k1 looks like “ki”, and `(k1)^2` is mathematical short hand for “k1k1”
`Frt` Is the user’s short form for the word “ferret”

But it is much harder to come up with a justification for a randomly-generated password than from a difficult password that you have created.

User Solutions: Once again, users will resort to writing down passwords or they’ll have to spend time getting them reset regularly.

2.4 Different logins for different systems

Security Measure: Each system has a separate login.

Security Goals: By requiring a separate username/password pair for each system, any successful attack will be limited to one system rather than compromising the whole.

In some instances, this is not so much a goal as laziness or lack of knowledge on the part of the administrators. It is often much easier, from an administration standpoint, to have separate logins. And, of course, some systems are run by different organizations that can’t or shouldn’t share information (for example, work email vs a free webmail service such as Hotmail).

User Problems: With more information to remember, there's just more to forget.

User Solutions: As before, users will choose simpler passwords and/or choose to write their passwords down.

In addition, users often use the same (or very similar) username/password pairs for multiple systems. This is convenient, but can be dangerous in some circumstances when the systems involved have very different levels of security or for whatever reason the administrators of one should not be able to gain access to the other. For example, it would be dangerous to re-use a work password to sign up for a contest on an untrusted site. The work password may be able to access sensitive information such as trade secrets, customer databases, or payroll information. Especially with web logins, users are often asked to provide an email address and a password, and the email address and other information requested could be used to figure out the system and username on which a password may also be used. Yet, if you want to be sure of remembering the "new" contest password, reusing a password would be one way to ensure that you do.

Sometimes, because of varying rules (see Section 2.2), users will be forced to choose slightly different usernames and passwords even if they intend to use identical ones. This then increases the likelihood that a user will mis-remember or confuse passwords when attempting to log in.

3 Why do users make insecure choices?

Now that we've established that "secure" rules can lead to insecure systems, we need to look at the factors that motivate people to make these choices. There are many factors that will encourage people to do things that result in decreased security.

3.1 Memory

Even from the first studies of memory we've known that nonsensical terms are difficult to recall. In the first scientific account of memory experiments [Ebbinghaus, 1885], Herman Ebbinghaus catalogued how quickly he forgot lists of nonsense syllables despite rehearsal. In Solso's description of this experiment, he says, "The nonmemorable terms ZAT, BOK, and QUJ were born to be forgotten and so they were" [Solso, 1998].

But although we have known for years that human memory works best when the information is linked and meaningful in some way, we try to remember meaningless passwords because they are "more secure."

On top of this, there are simply so many passwords to remember. Clear makes a long list of the security mechanisms and their related keys, cards, userIDs, PINs, user names, passwords, etc. that are required by a single university educator [Clear, 2002]. While each individual system may seem reasonable, the whole amounts to a huge mental load, especially when some of the passwords and PINs are not used frequently. For many people, the load becomes too great and they must rely on external reminders (sticky notes, palm pilots, etc.) to cope.

These two issues related to memory (difficulty of remembering non-meaningful items and number of items that must be remembered) are probably the most significant, but they are not the only issues related to human memory that need to be considered. Sasse et al. list the most important memory issues as follows [Sasse et al., 2001]:

- working memory capacity is limited
- memory decays over time, so people may forget or only partially remember items
- recognition is easier than recall
- frequently used items are more readily recalled than infrequently used ones
- items will linger in memory even when no longer needed
- meaningful items are more memorable than non-meaningful ones
- associating items can aid recall, but similar items will compete with each other on recall

3.2 Social issues

There is a bit of a societal stigma associated with security: people who are cautious with their passwords are described as paranoid or untrusting, [Sasse et al., 2001]. Others may feel that they are pedantic and overly formal for no good reason. There may even be the perception that they are untrustworthy as well as untrusting.

The perception of computing as anti-social is often cited as a factor that discourages women from entering computing-related fields [Henson, 2002], and the addition of security constraints may reinforce that perception of computers in users. This can be particularly dangerous when users feel that their people skills are very important to their job. A person working in Human Resources may be particularly motivated to show that they trust others and can be trusted, yet this person may also have access to very sensitive payroll information that should be protected.

For many security experts, being paranoid is considered a good quality (For example, a quick web search using Google turns up articles with titles such as *Security: selected readings for paranoid sysadmins*, or *Online Security: Only The Paranoid Survive.*), but for users it is not. For a user, refusing to give your password to colleagues (eg: so they can get at important files while you're on vacation) may come across as rude, and locking your computer implies that you don't trust your colleagues even when you've just gone to get a coffee. Spending the necessary time to stay informed about security issues in all the software a user may employ on a regular basis may be perceived as antisocial or a waste of time.

3.3 Poor system setup or poor work practices

If you are ill, can anyone access your files? Frequently, people must share passwords because the system is not designed in such a way that information can readily be shared. The sample password policy provided by SANS specifically lists "don'ts" that include revealing a password to a boss, or to co-workers on vacation [SANS, 2003]. Why? Because these are common practice in workplaces, even though they work at odds with good password security.

Once people get into the practice of revealing passwords, it doesn't seem strange when someone phones up, claiming to be from IT, and asks for their password to check for something. Most times, it really *is* someone from IT, but it could equally be an attacker. Kevin Mitnick, perhaps the world's most famous hacker, testified that he'd obtained more passwords by tricking users than through more technical means [Sasse et al., 2001].

This practice of password-sharing also hinders other security goals of passwords. The U.S. Federal Information Processing Standard for password usage recommends that individual passwords be used to establish illicit use and establish accountability [FIPS, 1985], but if a co-worker commonly uses a user's password, it becomes much harder to establish who is really at fault. A frustrated employee could easily abuse the system to cause problems for co-workers.

Poor system setup can go beyond passwords. Consider the case of email viruses. Many viruses have been spread based on the fact that one popular piece of software, Microsoft Outlook, automatically executes certain types of attachments. (For example, the Sobig virus, discovered this summer, spread in this manner by using a .pif attachment [Nahorney and Gudmundsson, 2003].) While Outlook has options to make it more secure, these are not the default choices, leaving uninformed users vulnerable.

Sometimes users will be required to make otherwise insecure choices in order to do what they want to do. For example, one of the assignment submission systems at Carleton included an applet which was not signed. In order to submit assignments, students were required to allow unsigned applets to execute code on their machines. Allowing all sites to use untrusted applets is not a particularly secure choice, but few browsers allow a per-site customization. Even Internet Explorer, currently the most popular browser, only allows different settings for a few security zones. In order to submit work, users were forced by the system to lower their security settings. Those lowered settings could potentially affect more than the one site, perhaps even all other sites unless the student remembers to change the settings before and after submitting each assignment. Many users of other systems have had the experience of turning security down or off (eg: disabling a firewall) in order to do what they wanted to do [Dourish et al., 2003].

3.4 Users don't understand how attacks occur

Many users think that an outside attacker will not be able to guess, for example, their spouse's name. If they realized that attackers use password dictionaries that often include common first names, they might realize that such a password is actually fairly easy to guess.

In addition, many users assume that attacks are always from the outside. Unfortunately, this isn't true. Discontented and former employees account for up to 65% of security breaches according to the FBI [Handley, 2002]. If the attacker is inside the company, they probably know or can find out things such as a spouse's name, children's names, and any number of other commonly-used password items. (For some examples of common passwords, see the end of Section 2.1.)

3.5 It gets in the way of more important things

This is one of the most important barriers to security. If secure practices are seen to get in the way of the more important things a users does (or those things which a user perceives to be most important), then these practices will be avoided by users.

A large number of support calls (estimates go as high as 70% [Trickey, 1998]) are password-related, and a large percentage of these calls are for forgotten passwords. It is fairly common practice to have users "locked out" so they can only make a limited number of attempts to log in before they must make a support call. Getting locked out means a larger amount of wasted time for the individual, and time spent making support calls does not particularly reflect well upon an individual in a workplace. When it's so simple and seemingly harmless to write down a password, and the alternative is a lot of wasted time and potentially some loss of face, the choice seems clear. After all, the wasted time is guaranteed – an intrusion isn't all that likely, right?

3.6 Security seems excessive or unnecessary

Many users underestimate the value of the data with which they work, and will continue to do so unless given specific feedback [Adams and Sasse, 1999]. Although some documents, for example, may seem inconsequential or even obvious, they may contain sensitive data that is not known outside a company.

A user who rightly perceives that they have no access to sensitive files could be wrong about being target for an intruder. Even a lower-privileged account can be used as a stepping stone to gain access to more sensitive data, but even when users realize that their own account is vulnerable, they do not often know how this could affect the entire system. Users underestimate their role in security. Adams and Sasse found that many users felt that since they were unimportant, they would never be targeted, although in reality an attacker may not necessarily target specific individuals but rather try to get into any account [Adams and Sasse, 1999].

3.7 It's someone else's problem

Douglas Adams describes the ultimate device for hiding things in his *Hitchhiker's Guide to the Galaxy* trilogy: It's called a "Someone else's problem field" (SEP) and it makes people think that the subject cloaked in the field, no matter how odd, is simply someone else's problem and need not be worried about.

While his SEP is a fictional device that generates a field over things that might otherwise be noticeable, security designs apparently suffer from a similar effect. Studies have shown that people, be they users or designers, have a tendency to delegate security to others [Flechais et al., 2003], [Dourish et al., 2003]. The other entity may be technology ("Oh, the server will handle the security"), another individual ("Well, my cousin who knows more about computers set it up") or an organization ("The IT department's system administrators handle that"). This is particularly noticeable when combined with the attitude that security gets in the way of work: it seems logical that it should be handled by someone else so the user's important work is not delayed. Unfortunately, it is often not someone else's responsibility and users, particuarly managers and users with higher status in a company, may mistakenly believe it is rather than taking personal responsibility.

3.8 Attitudes of security experts towards users

Parker (in [Adams and Sasse, 1999]) suggests that the *need-to-know* approach has been adopted by many security departments. Users are told very little because they are seen as a security liability. But as we've seen from previous sections (such as Section 3.4) it is actually lack of knowledge that can be dangerous. If users knew more, for example, about how dictionary attacks occur, they would understand better how to create good passwords. But if security experts continue to think of users as dumb because of the mistakes they make, and reinforce that by not teaching them anything, then we will be stuck with the status quo for a very long time.

This attitude that users are “lusers” is a significant barrier to good security. Just as in the airline industry, we need to look beyond “the user did that because the *user* is dumb and didn’t read the manual” to “the user did that because the *system* didn’t explain what the consequences were.” And we are unlikely to do so if users are seen as just a liability rather than an active participant in developing secure systems.

4 How can things be improved?

Thankfully, as well as finding problems with current security, people have been finding solutions. There are many interesting solutions which involve alternatives to regular passwords and PINs, and more which simply suggest things to evaluate when creating a secure system. Many of these guidelines have their roots in human factors and human computer interaction studies. Designers of security systems should be required to understand the same principles that underly other user-centred design practices.

“Tog” (Bruce Tognazzini, a human-computer interaction specialist) describes the security system at the hospital in which his wife works. Each user needs four sets of passwords in order to navigate the system. He suggests that the designers of this supposedly secure system really need to be shown what happens in practice:

> “[T]ake [the security engineer] into the offices in the hospital and let him see the four sets of user names and password clinging to the monitors on yellow stickies (e. g., Post-It Notes) or, for the more security-minded, slid into the top drawer where no one would think to look.” [Tognazzini, 2003]

He goes on to describe other problems with the four-password system, such as the fact that it had been common practice to fax patient records (thus avoiding the secured portion of the system) even though the fax machine was in the hall and easily accessible to any patient walking by.

Clearly, making sure that security engineers are aware of the actual ways in which their systems are used can greatly increase their awareness of what does and doesn’t work. A site visit such as the one Tog describes could do wonders for both the doctors using the system and the security of the final product. This sort of analysis has been recommended for years by human computer interaction specialists, yet it seems to be ignored frequently, particularly in the case of systems which are supposed to be secure.

Similarly, designers who are versed in human factors principles need to be versed in security. Several papers ([Dourish et al., 2003], [Flechais et al., 2003]) mention that people have the tendency to delegate security to other entities, which sometimes results in security being added to a system after the fact rather than being integrated properly.

This section outlines other guidelines which can be applied directly to security mechanisms. These guidelines are not all strictly technological – many encourage better attitudes towards security in the hopes that users will then be more motivated to maintain higher levels of security.

4.1 Limit the memory load

As seen in Section 3.1, nearly every system will be affected by memory issues because even if an individual system does not require a huge load, it’s unlikely that a user will use only one system.

4.1.1 Try to avoid multiple passwords

- Provide a single sign-on whenever possible.

  For example, logging in to a workstation can give the user access to all the appropriate things in a transparent way. Or multiple systems can share the same user name and password, so although the user signs on more than once, the userid and password is the same for each login.

- Use something you have

  While passwords are "something you know," authentication can also be done by "something you have" such as a smart card, or even a file on the computer (to make it seem like a single sign-on). Often these things are used in conjunction with passwords (eg: a bank card) but they can also be used alone (eg: many lab access cards do not require a PIN).

- Use alternative types of passwords.

  There are a number of systems which are based upon recognition rather than recall. (Examples of alternatives to traditional password systems are listed in [Sasse et al., 2001]) The Passfaces^{TM} system takes further advantage of what we know about human memory by using faces. Humans have especially high facial recognition compared to recognition of objects.

  It is also possible to find altnerate types of passwords which although they still require recall without cues, are easier to remember, such as graphical passwords. [Jermyn et al., 1999]

  See section 4.1.3 for some other discussion of other authentication methods.

Four or five is the maximum number of unrelated, regularly-used passwords that a user can be expected to handle comfortably [Adams and Sasse, 1999]

4.1.2 Teach users tricks for password creation

Users can be taught to create memorable but still secure passwords.

An excellent example can be found in [Sasse et al., 2001], where passwords are related to opinions regarding Star Trek characters. The statement, "Me, I am NOT impressed by Seven of Nine" becomes the password "m,IaNib7" – reasonably strong, yet still memorable. Any memorable phrase can be used for this purpose. Song lyrics work particularly well for many people, and most people can recall many different songs.

4.1.3 Use cues

Try using cued recall. That is, prompt users for the information they must remember [Patrick, 2002]. This technique is in fairly common use now as a back-up system to passwords. If the user forgets his or her password, then he or she is prompted with a question either of the system's choosing, sometimes selected from a list provided by the system (Selections might include "What is your birth date?" or "What is your mother's maiden name?"), or a question created by the user (such as "What was your grade 7 locker combination?").

It should be noted that while this is easier for users to remember, it is difficult to ensure that the answers to these questions aren't obvious to an attacker who has some knowledge of the user.

An informal study found that email accounts secured by this method were easily compromised by friends and even acquaintances who had some knowledge of the target [Taylor et al., 2002].

4.1.4 Reduce changes

Reduce the number of forced changes. When forced changes must occur, give users plenty of warning so they have time to come up with another good password [Adams and Sasse, 1999]. (The standard Windows log-on does give warning and it doesn't seem to help much, but it's better than no warning.)

4.1.5 Allow more attempts

Increase the number of failed attempts required before a user is locked out. Many users seem to recall part of a password, and may be able to enter it correctly given more tries [Sasse et al., 2001].

4.2 Provide feedback/information

Since one of the problems seems to be that users are simply unaware of security concerns, it makes sense to make them aware. It is particularly important that users see that security is taken seriously by the organization. One method for increasing awareness is to provide feedback regularly. User education will only be successful if users are motivated to learn more.

- Publish security reports including existing and potential threats. Many organizations try to avoid admitting to intrusions, but by not making users aware they do themselves a disservice because users will continue to underestimate the risks. If users are aware of the losses and potential losses for the organization then their willingness to be careful and their perception of security mechanisms will change for the better [Adams and Sasse, 1999].

- Let users know what information is sensitive and what isn't so they can act accordingly. Classifications on documents such as ”confidential” make it easier for users to know what sort of secure behaviour is needed to ensure that documents are safe, and many users are willing to help when they are aware of the sensitivity of a document [Adams and Sasse, 1999].

- Provide detailed feedback during password creation. By explaining what was wrong and *why* a given password is wrong, users will become more aware of the reasoning behind the rules and have a better sense of the importance of system security [Adams and Sasse, 1999].

- Take password infractions seriously [Adams and Sasse, 1999]. suggests that punishment is not the best option, (See Section4.3 for more on this.) but if nothing is done about security compromises then users get the impression that security doesn't really matter.

4.3 Improve employee morale

This may seem irrelevant to security until you recall the large percentage of breaches accounted for by discontented and former employees (as mentioned earlier, this number may represent up to 65% of intrusions [Handley, 2002]). Simply making sure employees are happy can greatly reduce your chance of attack, and can definitely decrease your chance of attack by an attacker with additional inside knowledge.

4.4 Make it easy

Yee suggests a number of design principles that can help users make better security choices [Yee, 2002]. Some of these have been summarized and grouped together here.

4.4.1 Make the path of least resistance the secure one

The easiest and most natural way of doing something should be the correct way.

Norman describes how most adults can put together a Lego toy motorcycle without any instructions simply because the constraints of what fits limit the choices [Norman, 1988]. We probably cannot constrain users so that they won't be able to write down passwords, but it is often possible to make secure choices the default choices for a system; thus, users who do nothing (taking the easiest path) will still have secure systems.

4.4.2 Make the interface clear and informative

- Ensure that the interface does what it says it does. If the expected abilities of the interface and the actual abilities don't match up, how can we expect users to make appropriate choices?
- The effects of any security-relevant action should be explained to the user before the action is taken. The interface also needs to help the user find actions that suit his or her goals.
- It should be easy for the user to tell what access has been granted and to whom.

4.5 Make access the user's choice

Once the user can tell what security settings are available, they should have the choice of changing them. The user should have the choice of allowing programs, websites, etc. to have privileges, (For example, allowing javascript to run only for certain websites) and these should be granted explicitly.

As Norman suggests [Norman, 1988], it should be easy for the user to change his or her mind once an action has been taken. This needs to include revoking access that a user has granted intentionally in the past, as well as simple mistakes.

5 Conclusions

It is time for people to look at security from a user perspective and include users in the design process. Tog delivers this message scathingly, yet perhaps most succinctly:

> "The universities, at least as evidenced by their graduates, are only interested in theory. That needs to change, and change now. The yellow sticky phenomenon has become so pandemic that it has received attention in both newspapers and business journals. I

> realized that many of these professors don't get out a lot, but they are at least supposed to read. Turning out graduates at this late date who are making security worse, instead of better, is just simply irresponsible." [Tognazzini, 2003]

The things cited in this paper are a good start, but they cannot be applied blindly. Each system should be designed with its users in mind. Some things will work in some environments but not others (for example, some alternatives to passwords require graphical capabilities that may not always be available or desireable). We need to consider the user environment, be it social, cultural or phsyical, and users' attitudes and desires. This may seem to be a lot of work, and that is part of the reason that useability is not always evaluated. However, it is hardly excessive to do this extra work if security is really a concern. Organizations need to balance the risk of intrusion with the work required to secure a system when deciding what to create [Flechais et al., 2003]. There are many circumstances in which this work would be justfied by the increased security that would result.

Only by taking users into account and trying to solve useability problems in secure interfaces can we ensure that systems will be secure in practice rather than just in theory.

References

[Adams and Sasse, 1999] Adams, A. and Sasse, M. A. (1999). Users are not the enemy: Why users compromise computer security mechanisms and how to take remedial measures. *Communications of the ACM*, 42(12):40–46.

[Chick, 2003] Chick, D. (2003). P for passwords. *TheNetworkAdministrator.com*. `http://www.thenetworkadministrator.com/passwords.htm` Accessed Nov 23, 2003.

[Clear, 2002] Clear, T. (2002). Design and usability in security systems: daily life as a context of use? *ACM SIGCSE Bulletin*, 34(4):13–14. COLUMN: Thinking issues.

[Dourish et al., 2003] Dourish, P., de la Flor, J. D., and Joseph, M. (2003). Security as a practical problem: Some preliminary observations of everyday mental models.

[Ebbinghaus, 1885] Ebbinghaus, H. (1885). Über das gedächtnis: Intersuchungen zur experimentellen psychologie. Translated by H. A. Ruger and C. E. Bussenius, 1913 and reissued by Dover Publications, 1964.

[FIPS, 1985] FIPS (1985). Password usage (publication 112). *Federal Information Processing Standards Publication*.

[Flechais et al., 2003] Flechais, I., Sasse, M. A., and Hailes, S. M. V. (2003). Bringing security home: A process for developing secure and usable systems. In *ACM/SIGSAC New Security Paradigms Workshop, Switzerland*.

[Handley, 2002] Handley, C. (2002). Inside security attacks are more frequent than external. *ITWeb: The technology news site*.

[Henson, 2002] Henson, V. (2002). Howto encourage women in linux. *The Linux Documentation Project*.

[Jermyn et al., 1999] Jermyn, I., Mayer, A., Monrose, F., Reiter, M. K., and Rubin, A. D. (1999). The design and analysis of graphical passwords. In *Proceedings of the 8th USENIX Security Syposium, Washington, D.C.*

[Moreu, 1995] Moreu, R. (1995). Hackers. Directed by Iain Softley.

[Nahorney and Gudmundsson, 2003] Nahorney, B. and Gudmundsson, A. (2003). W32.sobig.f@mm. *Symantec Security Response*.

[Nielsen, 2000] Nielsen, J. (2000). Security & human factors. *Jakob Nielsen's Alertbox*.

[Norman, 1988] Norman, D. A. (1988). *The Design of Everyday Things*. Basic Books, New York.

[Patrick, 2002] Patrick, A. (2002). Human factors of security systems: A brief review.

[Patrick et al., 2003] Patrick, A. S., Long, A. C., and Flinn, S. (2003). Hci and security systems. In *CHI 03 extended abstracts on Human factors in computer systems*, pages 1056–1057. Workshop session.

[SANS, 2003] SANS (2003). Password policy. *The SANS Security Policy Project*. Sample policies provided by the SANS Institute. Accessed Nov 27, 2003.

[Sasse et al., 2001] Sasse, M., Brostoff, S., and Weirich, D. (2001). Transforming the 'weakest link': A human/computer interaction approach to usable and effective security. *BT Technology Journal*, (19):122–131.

[Solso, 1998] Solso, R. L. (1998). *Cognitive Psychology, Fifth Edition*. Allyn and Bacon.

[Taylor et al., 2002] Taylor, K., Oda, S., and Zhu, C. (2002). Private communication.

[Tognazzini, 2003] Tognazzini, B. (2003). D'ohlt #2: Security d'ohlts. *Ask Tog*. Nielsen Norman Group.

[Trickey, 1998] Trickey, F. L. (1998). Secure sso: Dream on? *Information Security Magazine*.

[Yee, 2002] Yee, K.-P. (2002). User interaction design for secure systems.