CHAPTER TWO

Usable Security
Why Do We Need It? How Do We Get It?

M. ANGELA SASSE AND IVAN FLECHAIS

SECURITY EXPERTS FREQUENTLY REFER TO PEOPLE AS "THE WEAKEST LINK IN THE CHAIN" OF SYSTEM SECURITY. Famed hacker Kevin Mitnick revealed that he hardly ever cracked a password, because it "was easier to dupe people into revealing it" by employing a range of social engineering techniques. Often, such failures are attributed to users' carelessness and ignorance. However, more enlightened researchers have pointed out that current security tools are simply too complex for many users, and they have made efforts to improve user interfaces to security tools. In this chapter, we aim to broaden the current perspective, focusing on the usability of security tools (or *products*) and the *process* of designing secure systems for the real-world context (the *panorama*) in which they have to operate. Here we demonstrate how current human factors knowledge and user-centered design principles can help security designers produce security solutions that are effective in practice.

Introduction

The need for people to protect themselves and their assets is as old as humankind. Peoples' physical safety and their possessions have always been at risk from deliberate attack or accidental damage. The increasing use of information technology means that individuals and organizations today have an ever-growing range of physical (equipment)

and electronic (data) assets that are at risk. To meet the increasing demand for security, the IT industry has developed a plethora of security mechanisms that can be used to make attacks significantly more difficult or to mitigate their consequences.

A series of surveys has shown that—despite ever-increasing spending on security products—the number of businesses suffering security breaches is increasing rapidly. According to the United Kingdom's Department of Trade and Industry's Information Security Breaches Surveys,[1] 32% of UK businesses surveyed in 1998 suffered a security incident, rising to 44% in 2000 and 74% in 2002, and reaching a massive 94% in 2004. The 2004 CSI/FBI Computer Crime and Security Survey[2] reports that U.S. companies spend between $100 and $500 per employee per annum on security. But purchasing and deploying security products does not automatically lead to improved security. Many users do not bother with security mechanisms, such as virus checkers or email encryption, or do not use them correctly. Security products are often ineffective because users do not behave in the way necessary for security mechanisms to be effective. For example, users disclose their passwords, fail to encrypt confidential messages, and switch virus checkers off. Why? Because most users today:

- Have problems using security tools correctly (for an example, see the classic paper on PGP by Whitten and Tygar, reprinted in Chapter 34 of this volume), or

- Do not understand the importance of data, software, and systems for their organization, or

- Do not believe that the assets are at risk (i.e., that they would be attacked), or

- Do not understand that their behavior puts assets at risk

Whitten and Tygar have identified a "weakest link property," stating that attackers need to exploit only a single error. Frequently human frailty provides this error: humans are invariably described as the "weakest link" in the security chain. But until recently, the human factor in security has been neglected both by developers of security technology, and by those responsible for organizational security. Kevin Mitnick[3] points out that to date, attackers have paid more attention to the human element in security than security designers have, and they have managed to exploit this advantage prodigiously.

The aim of this chapter is to show how human factors knowledge and user-centered design principles can be employed to design secure systems that are workable in practice and prevent users from being the "weakest link." We view secure systems as socio-technical systems; thus, we are not just concerned with improving the usability of security mechanisms for individual users: our aim is to improve the effectiveness of security, and reduce the human and financial cost of operating it.

---

1   Department of Trade and Industry, *Information Security Breaches Survey* (2004); *http://www.security-survey.gov.uk/*.
2   Ninth Annual CSI/FBI Survey on Computer Crime and Security (2004); *http://www.gocsi.com/*.
3   Kevin D. Mitnick and William L. Simon, *The Art of Deception: Controlling the Human Element of Security* (New York: John Wiley & Sons Inc., 2003).

Security of any sociotechnical system is the result of three distinct elements: product, process, and panorama.

Product

What do current security policies and mechanisms require from the different stake-holders? Is the physical and mental workload that a mechanism requires from individual users acceptable? Is the behavior required from users acceptable? What is the cost of operating a specific mechanism for the organization, in both human and financial terms?

Currently, we have a relatively small set of general security mechanisms, and general policies that mandate user behavior when operating those mechanisms. Usable security requires a wider range of security policies and mechanisms, which can be configured to match the security requirements and capabilities of different users and different organizations.

Process

How are security decisions made? Currently, security is seen to be the responsibility of security experts. During the system development process, security is frequently treated as a nonfunctional requirement, and is not addressed until functionality has been developed. We argue that the organization's security requirements need to be determined at the beginning of the design process, and that the development of security mechanisms should be an integral part of design and development of the system, rather than being "added on." When deciding on a security mechanism, the implications for individual users (workload, behavior, workflow) need to be considered.

Usability and security are often seen as competing design goals. But in practice, security mechanisms have to be usable to be effective—mechanisms that are not employed in practice, or that are used incorrectly, provide little or no protection. To identify appropriate policies and usable mechanisms, all stakeholders have to be represented in the process of designing and reviewing secure systems.

Panorama

What is the context in which security is operated? Even a very usable security mechanism is likely to create extra work from the users' point of view. It is human nature to look for shortcuts and workarounds, especially when users do not understand why their behavior compromises security. User education and training have a role to play, but changing individuals' behavior requires motivation and persuasion, especially when the users' own assets are not at risk. A positive security culture, based on a shared understanding of the importance of security for the organization's business, is the key to achieving desired behavior. Effective and usable security requires effort beyond the design of user interfaces to security tools, specifically:

- *Security training in operating security mechanisms correctly*. Effective security training goes beyond instruction: it includes monitoring and feedback. Monitoring of security performance is an ongoing activity; action needs to be taken when policies are breached. For instance, mechanisms that are too difficult to operate need to be redesigned, or sanctions need to be carried out against users who refuse to comply.

- *Security education*. Users' motivation to comply is based on understanding why their behavior can put organizational assets at risk. Education needs to instill personal and collective responsibility in users, but also in security designers, administrators, and decision makers.

- *Political, ethical, legal, and economic constraints surrounding the system*. Currently, decision making on security is largely driven by technical considerations (e.g., security mechanisms are selected according to technical performance). However, other requirements may conflict with or override technical performance.

In this chapter, we explore each of these points in more depth. As stated at the outset, our aim is to broaden the view of what is involved in building usable secure systems. At this point in time, we cannot offer a blueprint for building such systems, but we identify relevant knowledge and techniques that designers who aim to design systems that are secure in practice will find helpful.

Product: Human Factors, Policies, and Security Mechanisms

It is unfortunate that usability and security are often seen as competing design goals in security, because only mechanisms that are used, and used correctly, can offer the protection intended by the security designer. As Bruce Tognazzini points out in Chapter 3, a secure system needs to be actually, not theoretically, secure. When users fail to comply with the behavior required by a secure system, security will not work as intended. Users fail to show the required behavior for one of the following two reasons:

- They are unable to behave as required

- They do not want to behave in the way required

Impossible Demands

The current situation with computer passwords provides a good example of the first case: most users today find it impossible to comply with standard policies governing the use of computer passwords (see Chapter 7 in this volume). Remembering a single, frequently used password is a perfectly manageable task for most users. But most users today have many knowledge-based authentication items to deal with. We have multiple and frequently changed passwords in the work context, in addition to passwords and personal identification numbers (PINs) outside work, some of which are used infrequently or require regular change. The limitations of human memory make it impossible for most users to cope with the memory performance this requires.[4] As a result, users behave in ways forbidden by most security policies:

---

4   M. Angela Sasse, Sacha Brostoff, and Dirk Weirich, "Transforming the 'weakest link': a human-computer interaction approach to usable and effective security," *BT Technology Journal 2001* 19, 122–131.

- *Users write passwords down*. Externalizing items we have to remember is the most common way of dealing with memory problems. In office environments, users stick notes with passwords onto their screens, or maintain a list of current passwords on the nearest whiteboard.

  Similarly, many bank customers write their PINs on their cards. A less common remedy is to write or scratch the PIN on the ATM or its surroundings.

ANECDOTAL EVIDENCE

- A reality TV show set in a police station in the UK featured a whiteboard behind a PC used to log the movement of prisoners with a prominent reminder:

  "The password is *custody*."

- The customer relations manager of a UK building society received irate phone calls after a major re-branding exercise, in which ATMs and the surrounding environments had been restyled. The customers did not object to the new corporate color scheme, but rather, to the fact that the panels and surroundings onto which they had written or scratched their PINs had been replaced, and as a result they were unable to withdraw cash.

- *Users share passwords with other users*. Another common way of preventing loss of data due to the vagaries of human memory is by sharing the information widely, so if you cannot remember the password, you are likely to find a colleague who can.

- *Users choose passwords that are memorable but not secure* (when the mechanism allows this).[5] Many users choose passwords or PINs that are memorable but easy to crack (names of spouses or favorite sports stars, birth dates, 1234, 8888).

The standard password mechanism is cheap to implement and—once recalled—executed quickly. But in the preceding examples, users are knowingly breaking the rules, and the examples give a feeling for the despair that the ever-growing number of passwords and PINs induces in many users. A key human factors principle is not to impose unreasonable demands on users; in fact, designers should minimize the physical and, especially, the mental workload that a system creates for the user.

---

5   Sacha Brostoff and Angela M. Sasse, "Ten strikes and you're out: increasing the number of login attempts can improve password usability," CHI Workshop on Human-Computer Interaction and Security Systems (Apr. 1–6, 2003, Ft. Lauderdale, FL).

Frequently used passwords—that is, passwords used on a daily basis—are not a problem for the average user in an office context. Infrequently used passwords and PINs, however, can create significant problems—for instance, many people withdrawing money once a week have problems recalling a PIN. There are a number of ways in which the memory demands of passwords and PINs can be reduced:

- *Provide mechanisms that require users to recognize items rather than recall them*. Recognition is an easier memory task than recollection, and designers of graphical user interfaces (GUIs) have applied this principle for decades now. Recognition of images[6, 7] has already been used for security mechanisms; but even text-based challenge-response mechanisms (see Chapter 8) and associative passwords[8] can offer improvements over the unaided recall that current passwords require.

- *Keep the number of password changes to a minimum*. Login failures increase sharply after password changes[9, 10] because the new item competes with the old one.

- *Provide mechanisms that are forgiving*. Current password and PIN mechanisms require the item to be recalled and entered 100% correctly. Brostoff and Sasse found[11] that users do not completely forget passwords. Most of the time they confuse them with other passwords, do not recall them 100% correctly, or mistype them on entry. This means that given a larger number of attempts, most users will eventually log in successfully. They report that when the standard limitation of three attempts was removed, the number of successful logins increased from 53% to 93% within nine attempts. Not having to reset a password saves users considerable time and effort—the time, effort, and possible embarrassment involved in contacting a system administrator or help desk, and having to think of, and memorize, a new password. From the organization's point of view, a 40% reduction of resets saves considerably in system administrator or help desk time.

As mentioned previously, usability and security are often seen as competing goals. Security experts are often inclined to reject proposals for improving usability (such as the ones listed earlier) because the help given to users might help an attacker. There is a tendency to discount more usable mechanisms because they may introduce an additional vulnerability or increase risk. For example, changing passwords less frequently means that a compromised password may be used longer. However, we would argue that a

---

6 Rachna Dhamija and Adrian Perrig, "Deja Vu: A User Study. Using Images for Authentication." *Proceedings of the 9th USENIX Security Symposium* (Aug. 2000, Denver, CO).

7 *Passfaces* (2004); *http://www.realuser.com/cgi-bin/ru.exe/_/homepages/index.htm.*

8 Moshe Zviran and William J. Haga, "Cognitive Passwords: The Key to Easy Access Control." *Computer and Security*, 9:8, 1990, 723–736.

9 Brostoff and Sasse, 2003.

10 Sasse, Brostoff, and Weirich.

11 Sacha Brostoff and Angela M. Sasse, "Are Passfaces More Usable Than Passwords? A Field Trial Investigation," People and Computers XIV—Usability or Else! *Proceedings of HCI 2000* (Sept. 5–8, 2000, Sunderland, UK), 405–424.
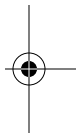
usable mechanism should not be dismissed immediately because it may introduce a new vulnerability or increase an existing one. Such a sweeping dismissal ignores the importance of human factors and economic realities, and—as Tognazzini points out in Chapter 3—the goal of security must be to build systems that are actually secure, as opposed to theoretically secure. For example, users' inability to cope with the standard requirements attached to passwords leads to frequent reset requests. This increases the load on system administrators, and in response many organizations set up help desks. In many organizations, the mounting cost of help desks has been deemed unacceptable.[12]

To cope with the increasing frequency of forgotten passwords, many organizations have introduced password reminder systems, or encouraged users to write down passwords "in a secure manner"—for example, in a sealed envelope kept in a locked desk drawer. But such hastily arranged "fixes" to unusable security mechanisms are often anything but secure:

- *Password reminders.* These may be convenient for users and a fast and cheap fix from the organization's point of view, but they create considerable vulnerabilities that can be exploited by an attacker, and the fact that the password has been compromised may not be detected for some time. For this reason, the FIPS password guidelines[13] mandate that forgotten passwords should not be reissued, but must be reset.

- *Encouraging users to write down passwords.* This violates the cardinal principle of knowledge-based authentication: that the secret should never be externalized. The "secure manner" of writing down passwords facilitates insider attacks. And relaxing the rules may seem to help users, but also has drawbacks. Simple but strong rules ("you should never write down this password, or tell anyone what it is") are easier for users to cope with than more permissive but complex ones ("it's OK to write down your password and keep it in a sealed envelope in your desk, but it's not OK to write it on a Post-it that you keep under your mouse pad").

The risks associated with changing passwords less frequently thus need to be weighed against the risks associated with real-world fixes to user problems, such as password reminders and writing down passwords. The FIPS guidelines actually acknowledge that the load on users created by frequent password changes creates its own risks, which in many contexts outweigh those created by changing a password less frequently. Allowing users more login attempts helps only a fellow user attacking the system from the inside, but makes no difference if the main threat is a cracking attack. Frequent changing or resetting of passwords, on the other hand, tends to lead users to create weaker passwords—more than half of users' passwords use a word with a number at the end,[14] a fact that helps crackers to cut down significantly the time required for a successful cracking attack.[15]

---

12 Sasse, Brostoff, and Weirich.

13 "Announcing the Standard for Password Usage," Federal Information Processing Standards Publication 112 (May 30, 1985).

14 Sasse, Brostoff, and Weirich.

15 Jeff Yann, "A Note on Proactive Password Checking," *Proceedings of the New Security Paradigms Workshop 2001* (ACM Press).

Awkward Behaviors

Sometimes users fail to comply with a mechanism not because the behavior required is too difficult, but because it is awkward. Many organizations mandate that users must not leave systems unattended, and should lock their screens when leaving their desks, even for brief periods. Many users working in shared offices do not comply with such policies when their colleagues are present. If a user locks the screen of his computer every time he leaves the office, even for brief periods, what will his colleagues think? They are likely to suspect that the user either has something to hide or does not trust them. Most users prefer to have trusting relationships with their colleagues. Designers can assume that users will not comply with policies and mechanisms requiring behavior that is at odds with values they hold.

Another reason why users may refuse to comply is if the behavior required conflicts with the image they want to present to the outside world. Weirich and Sasse[16] found that people who follow security policies to the letter—that is, they construct and memorize strong passwords, change their passwords regularly, and always lock their screens—are described as "paranoid" and "anal" by their peers; these are not perceptions to which most people aspire. If secure systems require users to behave in a manner that conflicts with their norms, values, or self-image, most users will not comply. Additional organizational measures are required in such situations. For example, a company can communicate that locking of one's screen is part of a set of professional behaviors (e.g., necessary to have reliable audit trails of access to confidential data), and not because of mistrust or paranoia. Labeling such behaviors clearly as "it's business, not personal" avoids misunderstandings and awkwardness among colleagues. In organizations where genuine security needs underlie such behavior, and where a positive security culture is in place, compliance can become a shared value and a source of pride.

For designers of products aimed at individual users, rather than corporations, identifying security needs and values ought to be the first step toward a usable security product. The motivation to buy, install, and use a security product is increased vastly when it is based on users' security needs and values—in Chapter 24 of this volume, Friedman, Lin, and Miller provide an introduction to value-based design and further examples.

Beyond the User Interface

The need for usability in secure systems was first established in 1975, when Saltzer and Schroeder[17] identified the need for *psychological acceptability* in secure systems. Traditionally, the way to increase acceptability has been to make security mechanisms easier to use (by providing better user interfaces). The most widely known and cited

---

16 Dirk Weirich and M. Angela Sasse, "Pretty Good Persuasion: A First Step Towards Effective Password Security for the Real World," *Proceedings of the New Security Paradigms Workshop 2001* (Sept. 10–13, Cloudcroft, NM); (ACM Press), 137–143.

17 Jerome H. Saltzer and Michael D. Schroeder, "The Protection of Information in Computer Systems," *Proceedings of the IEEE*, 63:9 (1975), 1278–1308.

paper on usability and security, "Why Johnny Can't Encrypt" (reprinted in Chapter 34 of this volume), reports that a sample of users with a good level of technical knowledge failed to encrypt and decrypt their mail using PGP 5.0, even after receiving instruction and practice. The authors, Alma Whitten and Doug Tygar, attributed the problems they observed to a mismatch between users' perception of the task of encrypting email and the way that the PGP interface presents those tasks to users, and they proposed a redesign to make the functionality more accessible.

User-centered design of security mechanisms, however, is more than user interface design. The case of PGP presents a good example. The problem lies less with the interface to PGP and more with the underlying concept of encryption (which predates PGP). The concept of encryption is complex, and the terminology employed is fundamentally at odds with everyday language: a cryptographic key does not function like a key in the physical world, and people's understanding of "public" and "private" is different from how these terms are applied to public and private keys. This will always create problems for users who do not understand how public-key encryption works. While some security experts advocate educating all users on the workings of public-key encryption so that they can use PGP and other encryption mechanisms, we argue that it is unrealistic and unnecessary to expect users to have the same depth of understanding of how a security mechanism works. Some computing people in the 1980s argued that it would never be possible to use a computer without an in-depth knowledge of electronics and programming; arguing that all users will have to become security experts to use systems securely is similarly misguided. The conceptual design approach, pioneered by Don Norman,[18] has been used to make complex functionality available to users who don't understand the detailed workings of a system, but have a task-action model ("if I want this message to be encrypted, I have to press this button").

However, the way in which people interact with security policies and mechanisms is not limited to the point of interaction. It is a truism of usability research that a bad user interface can ruin an otherwise functional system, but a well-designed user interface will not save a system that does not provide the required functionality. Designers can expend much effort on making a security mechanism as simple as possible, and find that users still fail to use it. Using a well-designed security mechanism is still more effort than not using it at all, and users will always be tempted to cut corners, especially when they are under pressure to complete their production task (as we will discuss later in this chapter). To make an effort for security, users must believe that their assets are under threat, and that the security mechanism provides effective protection against that threat.

18 Donald A. Norman, "Some Observations on Mental Models," in D.A. Gentner and A.A. Stevens (eds.), *Mental Models* (Hillsdale, NJ: Erlbaum).

Process: Applying Human Factors Knowledge and User-Centered Approaches to Security Design

The process of building a secure system is vital to its effectiveness. The *process* is the means by which security needs are assessed, policies are elaborated, and countermeasures are designed. As with any software development project, the right mix of participants, expertise, and methodology is vital in ensuring a system that is actually secure. To achieve this, designers of secure systems need to consider that security is not the primary goal of users and organizations. The role of security is a supporting one—to protect assets and activities that users care about or that are part of the production activity of business organizations.

Security Is a Supporting Task

Two further concepts that are key to designing successful security applications are *goals* and *tasks*. Human behavior is essentially goal driven, so the effective and efficient execution of tasks that help users attain goals is a key principle for designing successful systems. Human factors analysts distinguish between *production tasks* (those that are required to achieve the goal or produce the desired output) and *supporting tasks* (those that enable production tasks to be carried out in the long run, or be carried out more efficiently, but are not essential to achieving the goal). Security—like safety—is a supporting task. Production tasks are the reason why a system exists, and if production tasks cannot be completed effectively and efficiently, the system will cease to exist. Users put production tasks first; organizations, sensibly enough, do the same from a higher-level perspective. This understanding leads us to a number of insights for security design:

- *Security tasks must be designed to support production tasks*. Security tasks must not make demands on users that conflict with the demands of their production tasks. The performance requirements for a security task must be derived from the performance requirements for the production task. The current reality is that security mechanisms are often chosen without consideration of the production tasks, and individual users are often left to make a choice between complying with security regulations on the one hand or getting their job done on the other—and the choice they make is predictable. When security needs require a reduction in the efficiency of a production task, the need for the extra effort has to be communicated clearly to users. Tradeoffs between production tasks and security should not be made by security experts, who naturally prioritize security over efficiency. Rather, these decisions should be made in consultation with those in charge of business processes and workflow.[19]

- *Users need to understand and accept the need for security tasks*. In an ideal world, we would have systems where security is integrated seamlessly and demands no extra effort. We could, for instance, imagine a gait recognition system that identifies users as they walk

---

19 Sacha Brostoff and Angela M. Sasse, "Safe and Sound: A Safety-Critical Approach to Security Design," New Security Paradigms Workshop 2001.

up to a door and open it to those who are authorized, remaining shut to those who are not. In reality, however, even a well-chosen and well-configured security mechanism demands extra effort—in the gait example, users may need to remember to carry a token that identifies them and to make special arrangements to take visitors into the building. To avoid users' natural inclination to shortcut security, they need to understand and accept the need for the security task, and be motivated to comply with it.

A Process for Designing Usable Secure Systems

Zurko and Simon[20] were among the first to point out that current security mechanisms make unreasonable demands on all stakeholders: system administrators and system developers, as well as users, struggle with the increasing amount and complexity of work involved in keeping systems secure.

System administrators struggle with the increasing workload involved in securing systems at all possible levels (hardware, operating system, network, applications), keeping up with patches, registering users, and managing accounts.

Many developers feel overwhelmed by the complexity involved in securing the systems they develop. Often, security weaknesses are introduced because developers do not realize the security implications of their design decisions. Because security is seen as a nonfunctional requirement in software engineering terms, the need to secure functions is often not considered until the design is completed. Users often compound this problem by asking to see functions working as early as possible. Even when a security analysis has been done at the outset of the project, the implications for design may not be considered because they are kept in a document separate from the system specification. Today, developers are often left with the responsibility for making security decisions in new applications.

To address these issues, Flechais, Sasse, and Hailes[21] have proposed an integrated development method for secure systems that does the following:

- Brings together all stakeholders (system developers, owners, users, administrators, and security experts) to carry out a risk analysis and to consider the practical implications of proposed security mechanisms in the context of use

- Integrates security into the software engineering documentation that developers refer to throughout the development process

Appropriate and Effective Guidance for Information Security (AEGIS) is a sociotechnical software engineering methodology for creating secure systems based on asset modeling, security requirements identification, risk analysis, and context of use. The purpose is to

---

20 Mary E. Zurko and Richard T. Simon, "User-Centered Security," New Security Paradigms Workshop 1997.

21 Ivan Flechais, Angela M. Sasse, and Stephen Hailes, "Bringing Security Home: A Process for Developing Secure and Usable Systems," *Proceedings of the New Security Paradigms Workshop 2003.*

provide system developers with simple and intuitive tools for producing a secure system that takes end user needs into account and promotes security buy-in. The core processes of AEGIS are shown in Figure 2-1.
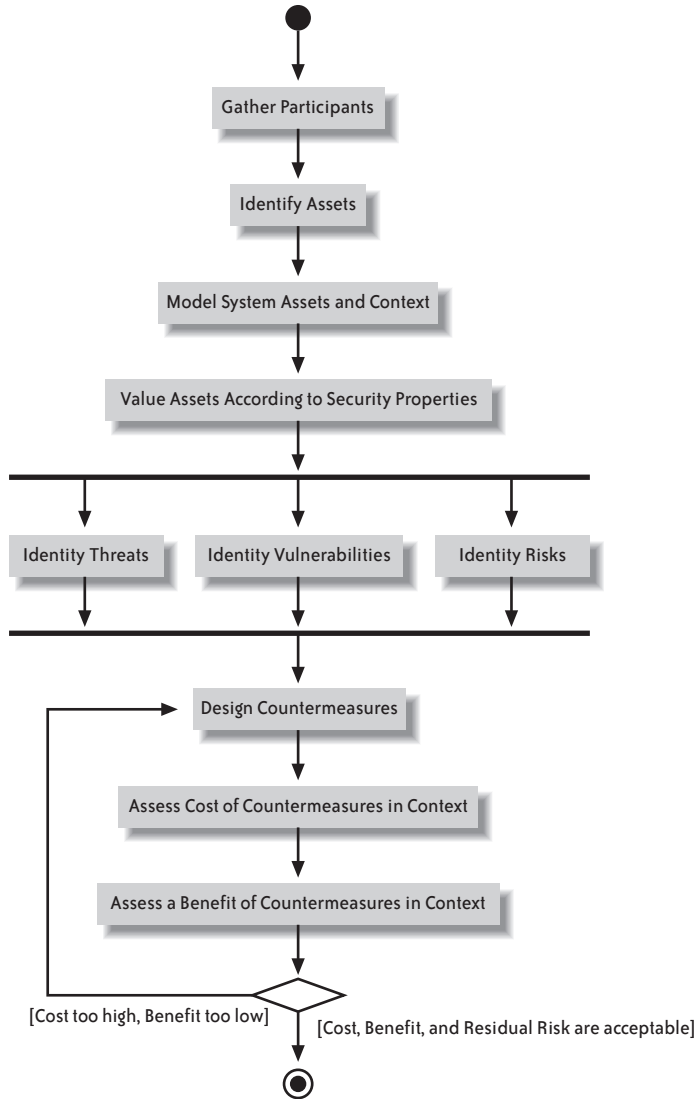

FIGURE 2-1. *AEGIS activity diagram*

The core AEGIS processes consist of:

1. *Gathering participants in the design process*. This requires identifying and ensuring the participation of key stakeholders, including users, managers, and system owners.

2. *Identifying the system's assets.* Assets represent the most fundamental valuables in a system. These include hardware or software components, physical artifacts, employees, etc.

3. *Modeling assets in the context of operation*. As seen from the HCI design technique known as *contextual design*,[22] understanding the context in which the system operates is a useful tool for designing a practical and usable system. Modeling the physical and cultural environment of the assets provides greater information about the system that can then inform the security design.

4. *Identifying security requirements on the assets*. Getting stakeholders to assign a value to the assets according to certain security properties (such as confidentiality, integrity, and availability) gives a clear insight into which aspects of the system are most important. This also provides greater clarity into which aspects of security deserve the most attention—for example, providing a high degree of availability requires a different architecture from satisfying a high confidentiality requirement. Figure 2-2 shows an example model in which assets, context, and security requirements have been recorded.

5. *Conducting a risk analysis in which vulnerabilities, threats, and risks are identified*. Together with the identification of important security requirements, this allows the identification of areas in which the system is at risk and the potential impact to the system is deemed to be high.

6. *Designing the security of the system*. This design is based on the security requirements identified by the stakeholders and the risk analysis highlighting areas where the system is unacceptably vulnerable. At this point, countermeasures are proposed and evaluated against both their cost and their effectiveness. The contextual information identified previously is important in assessing the cost of the countermeasure to the system as a whole—this includes financial, organizational, and user costs. Identifying the benefit of the countermeasure depends on an assessment of the effectiveness of that measure at preventing, detecting, or reacting to identified risks. Based on a better understanding of the impact of the countermeasure, it can then either be accepted or rejected as a part of the architecture.

By involving stakeholders in the security analysis, AEGIS provides several benefits:

• It provides increased awareness of security in the participants, allowing them to identify a number of problems and issues with security themselves, and providing a wealth of information about the needs of stakeholders. This information is elicited and recorded in the asset model, which is used throughout the security design.

• The security aspects of the system become much more accessible and personal. This can be invaluable in combating security apathy, and can be a powerful means of overcoming a lack of security motivation.

• By providing a simple model through which the security properties of the system can be discussed by stakeholders, communication during the design of security is improved—and that supports better security decision making.

---

22 Hugh Beyer and Karen Holtzblatt, *Contextual Design* (San Francisco, Morgan Kaufmann, 1977).
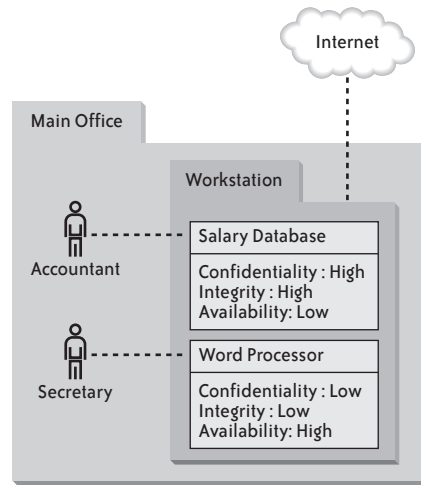
FIGURE 2-2. *Sample AEGIS asset model*

Panorama: Understanding the Importance of the Environment

The environment surrounding the process of developing security is also extremely important to the effective operation of the product (security mechanism). During the design of the security, a number of factors not necessarily related to any security needs can influence the final product. The personal responsibility of participants for the resulting security, the enthusiasm of high-level management for security (even their presence in the design process), pressure to achieve functional features, time-to-market, personal agendas, ethical constraints, industrial involvement, and legal requirements—all influence security design in one way or another.[23]

The cultural panorama surrounding security does not stop affecting it after the design is complete, but continues even after it has been put to use. An analysis by Flechais, Riegelsberger, and Sasse[24] has identified the influences and mechanics of trust relationships in the operation of secure systems. In most current cases, existing trust relationships in an organization facilitate the breaking of security policies and practices. In fact, given the right (and currently widespread) environment, simply adhering to existing security policies can undermine social relationships within a group of peers. The authors argue that trust relationships are beneficial for organizations by promoting *social capital*[25] (i.e., trust based on shared informal norms that promote cooperation)[26] and that the organizational culture and the actual security should be designed to support both trust relationships and adherence to policy.

23 Flechais, Sasse, and Hailes.
24 Ivan Flechais, Jens Riegelsberger, and Angela M. Sasse, "Divide and Conquer: The Role of Trust and Assurance in the Design of Socio-Technical Systems," Technical Report, 2004.
25 Mitnick and Simon.
26 Brostoff and Sasse, 2001.

The Role of Education, Training, Motivation, and Persuasion

While a well-designed security mechanism won't put off users, it also won't entice them to make the extra effort that security requires. In many home and organizational contexts, users lack the motivation to make that extra effort. User education and training can be used to explain the need for the effort that security requires, but changing users' knowledge and understanding does not automatically mean they will change their *behavior*. Dhamija and Perrig,[27] for instance, found that a sample of users with weak passwords had "received relevant training" and did know how to construct strong passwords; however, they chose not to comply with the request to construct strong passwords. The first point to make here is that there is a difference between education and training: while *education* is largely about teaching concepts and skills, *training* aims to change behavior through drill, monitoring, feedback, reinforcement, and—in the case of willful noncompliance—punishment. Because social engineering attacks often bypass technology altogether to obtain access or information, Mitnik and Simon[28] emphasize that effective security education and training should:

- Not only focus on correct usage of security mechanisms, but also address other behaviors—for example, checking that callers are who they claim to be

- Encompass all staff, not only those with immediate access to systems deemed at risk

Many organizations simply provide security instructions to users and expect these instructions to be followed. The material disseminated may even threaten punishment. However, the threat of punishment alone won't change users' behavior—rather, if users see that rules are not enforced, they will lose respect for the security in general, and the result is a declining security culture. Even though some security experts advocate rigorous punishment as a way of weeding out undesirable user behavior, this is not an easy option. Policing undesirable behavior—detection and punishment—requires considerable resources, can be detrimental to the organizational climate, and may have undesirable side effects (such as increasing staff turnover). Given that sanctions have an effect only if they are applied, and given that there may be undesirable side effects, an organization would be well advised to specify sanctions only for a small set of key behaviors that it deems to be putting key assets at risk.
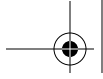
Weirich and Sasse[29] identified a set of beliefs and attitudes held by many users who do not comply with security policies:

- Users do not believe they are personally at risk.

- Users do not believe they will be held accountable for not following security regulations.

27 Dhamija and Perrig.
28 Mitnick and Simon.
29 Weirich and Sasse, 2001.

- The behavior required by security mechanisms conflicts with social norms.

- The behavior required by security mechanisms conflicts with users' self-image. (The perception is that only "nerds" and "paranoid" people follow security regulations.)

There can be no doubt that security in general, and IT security in particular, currently suffers from an image problem. Education campaigns (similar to those employed in health education) can be effective only if they make users believe that something they care about is at risk. In the most recent CSI/FBI survey,[30] the overwhelming majority of respondents stated that their company needed to invest in raising security awareness. When users cannot be motivated, persuasion needs to be employed. In this chapter, we present some examples of persuasion designed to improve security behavior in the corporate context; Fogg[31] offers techniques for designing applications and interfaces that intrigue, persuade, and reward users to achieve desired user behavior in general.

Building a Security Culture

Earlier in this chapter, we emphasized the importance of having organizations integrate security into their business processes, and argued that the best motivation for users to exhibit desired security behavior is if they care about what is being protected, and understand how their behavior can put these assets at risk. These two arguments provide the foundation for the next key point: organizations must become actively involved in security design. They need to build a security culture as much as they need to build a system of technical countermeasures. Although some organizations understand that risk analysis is the bedrock of security design, many still do not understand the role of security in their business/production processes. Too many organizations are still copying standard security policies, deploying standard security mechanisms, and leaving decisions about security largely to security experts. Security decisions are then often made in an ad hoc fashion, as a "firefighting" response to the latest threat.
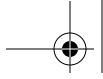
Organizations need to become actively involved in making decisions about what should be protected, and how. This requires performing a risk and threat analysis, and making decisions based on what makes economic sense for the business, instead of trying to meet abstract standards set by security experts. Although many companies already use risk analysis methods, they often fail to consider the interests and needs of all stakeholders—such as users—and the economics of security are currently not well understood.

Once specific security goals appropriate to the organization have been established, role models are essential to change behavior and rebuild the security culture. This will require buy-in from the top. Currently, senior managers sometimes exhibit bad security behavior because they believe that they are too important to bother with "petty" security policies. The security experts to whom they have delegated responsibility for the organization's

---

30 Ninth Annual CSI/FBI Survey on Computer Crime and Security (2004); *http://www.gocsi.com/*.

31 B. J. Fogg, *Persuasive Technology. Using Computers to Change What We Think and Do* (San Francisco: Morgan Kaufmann, 2003).

responsibility are often unable to force senior managers—who have the power to fire them—to comply with security policies. We would argue that the ultimate responsibility for security, as for safety, always should lie with senior management. Security experts can advise, implement, and monitor, but they cannot take sole responsibility for making an organization's security work.

An additional approach worth considering is to make secure behavior a desirable trait. This can be done through social marketing or by making such behavior part of professional and ethical norms.[32] Organizations that deal with confidential customer data, for instance, must make clear to all of their staff that they have a duty to safeguard such data from unauthorized access or tampering.
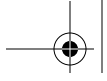
Conclusion

This chapter started with the observation that only usable security is effective security, and outlined how human factors knowledge and user-centered design techniques can be applied to increase usability. Effective security requires us to look beyond the user interface to security tools, where most of the current research and development effort is focused. Changing undesirable user behavior is a complex task, and one that cannot be achieved by education or punishment alone. An organization is a sociotechnical system, and security design needs to address both technical and human aspects. Furthermore, security needs to be integrated into the business processes of an organization to be workable in practice and economically viable.

As a first step in this direction, Brostoff and Sasse[33] have adapted Reason's model of human error (a sociotechnical model for improving safety behavior in organizational contexts) to security. Reason's model is a good starting point because safety and security share the "supporting task" problem. Two key differences are that the benefits of safety are more obvious to most users, and that safety does not have adversaries who actively seek to attack. In many Western countries, health and safety regulations have led to significant changes in organizational culture with respect to employee safety. Responsibility for safety lies with management, for they allocate resources; this chapter has made the argument that security needs to be viewed in a similar way.

32 Sasse, Brostoff, and Weirich.

33 Brostoff and Sasse, 2001.

About the Authors

M. Angela Sasse is the Professor of Human-Centred Technology in the Department of Computer Science at University College London. After obtaining an M.Sc. in Occupational Psychology (from the University of Sheffield) and a Ph.D. in Computer Science (from the University of Birmingham), she joined UCL in 1990 to teach and research design and evaluation of emerging technologies. Since 1997, her research has focused on user-centered approaches to security, privacy, and trust.

Ivan Flechais is a departmental lecturer in the software engineering program at Oxford University, and his main lecturing and research interests are in the area of computer security. Prior to this, he graduated with a B.Sc. in computer science from University College London, and then stayed on at UCL with a Ph.D. researching security design and the importance of people in computer security.