

# Chapter 1

## Introduction

It is quite common for essays on the uptake of new information technology (IT) to start with a remark such as:

Security concerns are a major reason for holding back the take-up of new information technologies, thus preventing citizens and companies from reaping the full benefits these technologies would offer.

Statements like this are made by academics writing books on security, by consultants trying to convince customers of the value of their services, by vendors of security products or by government officials in charge of security programs. Security stories play well in the media, working with powerful motifs like the fall of the mighty (Microsoft) or the fear of invisible foes (viruses and worms creeping around in the Internet).

People with an axe to grind certainly have reasons to exaggerate the dangers we are facing and it is often difficult to obtain hard evidence for assessing the real size of the problem. On the other hand, threats are real, as anyone who has been a victim of a worm or virus will confirm. Indeed, the widespread use of open communications networks like the Internet, or of cellular telephone systems, has exposed a large user population to security threats. A fundamental understanding of the potential vulnerabilities of such networks, of the core protection mechanisms and of their limitations has thus become essential for IT professionals.

This book is about computer security. The original focus of computer security was on multiuser systems. Users had to be kept apart and unauthorized

users had to be stopped from modifying systems software. Today's focus is on computing devices that may figure as the end systems in a network. Many security issues stem from the fact that these devices are connected to a network and may in some way be attacked from 'untrusted' nodes. Traditional network security services protect traffic between nodes, with the task completed once a message has been safely handed over to the other side. Our problems start when the message has been received and is then processed within the end system.

Before moving to the technical content of this book, this first chapter will go over some important issues that have to be addressed when trying to implement security measures in practice. The deployment of security measures (and of IT in general) is a management decision, and technical security measures have to work hand in hand with organizational measures to be effective. Management decisions should be underpinned by some analysis of current risks and threats. Hence, we will give a brief survey of security management and of risk and threat analysis.

## OBJECTIVES

- Set the scene for our discussion of computer security.
- Give a brief introduction to security management.
- Cover the basics of risk and threat analysis.

## 1.1 ATTACKS AND ATTACKERS

Not so long after the first generation of cellular telephone systems had established a customer base, members of the British royal family found some very private phone calls reprinted in newspapers. These systems transmitted the traffic between a mobile device and a base station in the clear, so anyone with the right equipment could listen in on telephone calls. Second generation systems like GSM then included cryptographic mechanisms for protecting the radio link. Similar confidentiality concerns are raised by credit card purchases over the Internet. The basic Internet protocols provide no confidentiality protection so parties located between customer and merchant could capture card numbers and use them later for fraudulent purchases. Secure Socket Layer (SSL) was developed by Netscape to deal with this very problem.

However, the real danger may lurk somewhere else. Scanning Internet traffic for packets containing credit card numbers is an attack with a low yield. Badly protected servers at a merchant site holding a database of customer credit card numbers are a much more

rewarding target. There is documented evidence that such attacks have occurred, either to obtain credit card numbers or to blackmail the merchant.

*Identity theft*, i.e. using somebody else's 'identity' (name, social security number, bank account number, etc.) to gain access to a resource or service, exploits an inherent weakness in services that use non-secret identifying information to authenticate requests.

Vulnerabilities in software accepting external user input, like Internet browsers or mail software, may allow external parties to take control of a device. Attackers may corrupt data on the device itself or use it as a stepping stone for attacks against third parties. Worms and viruses make use of overgenerous features or vulnerabilities to spread widely and overload networks and end systems with the traffic they generate. The Internet worm of November 1988 is an early well-documented example of this species (Eichin and Rochlis, 1989). Denial-of-service attacks against specific targets have started to occur in the last few years. Resilience against denial-of-service attacks has become a new criterion in the design of security protocols.

Returning to the first example, as traffic between mobile and base station was unprotected, attackers could also capture the 'secret' identifiers used to authenticate customers for charging purposes. Major road intersections were a popular place for fraudsters to lie in wait. With these identifiers, attackers could then create cloned phones and make calls that were charged to the victim's account. There was a time when reportedly up to 50% of phone calls were fraudulent in some particularly badly affected networks.

GSM uses a challenge/response protocol for subscriber authentication and does not transmit secrets in this process, so this particular attack can no longer be mounted. However, this certainly does not imply that all charging problems have been solved. Today, we see attempts to lure unwitting customers into calling back to premium rate numbers owned by the attacker, using the existing charging system to get the victim's money. This is an attack (mis)using a technical system, rather than an attack exploiting a flaw in a technical system. Countermeasures are to be found at the human level, e.g. exercising caution before answering a call back request, and in the legal system, e.g. clarifying how user consent has to be sought for subscribers to be liable for charges to their account.

In the scenarios described above the attacks came from the outside. Keeping the enemy outside the castle walls is a traditional paradigm in computer security. However, typical statistics for the sources of attacks show that attacks from insiders account for a majority of incidents and the largest proportion of damages (United Nations, 1999). There is a suggestion that attacks via the Internet might change this picture, but insider fraud remains a considerable concern in organizations and in electronic commerce transactions.

It has been said that the goal of security engineering is to raise the effort involved in an attack to a level where the costs exceed the attacker's gains. Such advice may be short

sighted. Not every attacker is motivated by a wish for money. Employees who have been made redundant may want to exact revenge on their former employer. Hackers may want to demonstrate their technical expertise and draw particular satisfaction from defeating security mechanisms that have been put in their way. ‘Cyber vandals’ may launch attacks without much interest in their consequences. Political activists may deface the web sites of organizations they dislike.

There is similar variance in the expertise required to break into a system. In some cases insider knowledge will be required to put together a successful attack plan. In this respect, *social engineering* may be more important than technical wizardry (Mitnick and Simon, 2002). Hassling computer operators on the phone to give the caller the password to a user account is a favorite ploy. Some attacks require deep technical understanding. Other attacks have been automated and can be downloaded from web sites so that they may be executed by *script kiddies* who have little insight into the vulnerabilities or features these attacks are exploiting.

## 1.2 SECURITY

Software may crash, communication networks may go down, hardware components may fail, human operators may make mistakes. As long as these failures cannot be directly attributed to some deliberate human action they would not be classified as security issues. Accidental failures would count as *reliability* issues. Operating mistakes would be attributed to *usability* issues. Security is concerned with *intentional* failures. There may not always be a clear intent to achieve a particular goal, but there is at some stage a decision by a person to do something they are not supposed to do. As sketched above, the reasons for such actions can be manifold. The root cause of security problems is human nature.

Security practitioners know that ‘security is a people problem’ that cannot be solved by technology alone. The legal system has to define the boundaries of acceptable behavior through data protection and computer misuse laws. Responsibility for security within organizations, be they companies or universities, resides ultimately with management. Users have to cooperate and comply with the security rules laid down in their organization. Of course, correct deployment and operation of technical measures is also part of the overall solution.

## 1.3 SECURITY MANAGEMENT

Protecting the assets of an organization is the responsibility of management. Assets include sensitive information like product plans, customer records or financial data, and the IT infrastructure of the organization. At the same time, security measures often restrict members of the organization in their working patterns and there may be a

certain temptation to flaunt security rules. This is particularly likely to happen if security instructions do not come from a superior authority but from some other branch of the organization.

It is thus strongly recommended to organize security responsibilities in an organization in a way that makes it clear that security measures have the full support of senior management. A brief *policy* document signed by the chief executive that lays down the ground rules can serve as a starting point. This document would be part of everyone's employment handbook. Then, *security awareness* programs should be organized. Not every member has to become a security expert, but all members should know:

- why security is important for themselves and for the organization;
- what is expected of each member;
- which good practices they should follow.

The mirror image of users ignoring apparently unreasonable security rules is security experts treating apparently unreasonable users as the enemy. Trying to force users to follow rules they regard as arbitrary is not an efficient approach. Studies have shown that involving users as stakeholders in the security of their organizations can make users voluntarily comply with rules rather than looking for workarounds (Adams and Sasse, 1999).

Organizations developing IT services or products have the additional task of providing security training for their developers. There is rarely a clear dividing line between the security-relevant components and the rest of a system. It thus helps if developers in general are aware of the environment a service will be deployed in and of the expected dangers, so that they can highlight the need for protection even if they do not implement the protection mechanisms themselves. Developers should also be alert to the fact that certain categories of sensitive data, e.g. personal data, have to be processed according to specific rules and regulations. Finally, developers should keep up to date with known coding vulnerabilities.

### 1.3.1 Security Policies

Security policies state what should be protected but may also indicate how this should be done. To maintain clarity in our terminology, we follow the definitions laid out in Sterne (1991) and distinguish between organizational and automated security policies. A policy has given objectives.

**Security Policy Objective** A statement of intent to protect an identified resource from unauthorized use.

A policy also has to explain how the objectives are to be met. This can be done first at the level of the organization.

**Organizational Security Policy** The set of laws, rules and practices that regulate how an organization manages, protects and distributes resources to achieve specified security policy objectives.

Within an IT system, organizational policies can be supported by technical means.

**Automated Security Policy** The set of restrictions and properties that specify how a computing system prevents information and computing resources from being used to violate an organizational security policy.

Automated policies address issues like the definition of access control lists or firewall settings, decisions on the services that may be run on devices and the security protocols used to protect network traffic.

### 1.3.2 Measuring Security

Measuring security is the holy grail of security engineering. To convince managers (or customers) of the benefits of a new security mechanism, wouldn't it be nice if we could measure the security of the system before and after introducing the mechanism? Indeed, it is difficult to reach well-founded management decisions if such quantitative information cannot be procured. Ideally, a *measurement* would give a quantitative result that can be compared to other measurements, not just a qualitative statement about the security of the product or system being analyzed.

- A *product* is a package of IT software, firmware and/or hardware, providing functionality designed for use or incorporation within a multiplicity of systems.
- A *system* is a specific IT installation, with a particular purpose and operational environment (CCIB, 2004a).

Measurements of a product are indicative of its potential security, but even a secure product can be deployed in an insecure manner. A notorious example is a service account whose default password is not changed. It is thus a task for security management to ensure that the security features provided are properly used. For a product, one might use the number of security flaws (bugs) detected as a measure of its security. Tracking the discovery of flaws over time may serve as the basis for predicting the time to the discovery of the next flaw. Relevant methodologies have been developed in the area of *software reliability*. These methodologies assume that the detection of flaws and the invocation of buggy code are governed by a probability distribution given a priori, or a given family of probability distributions where parameters still have to be estimated. Another proposal is to measure the *attack surface* of a product, i.e. the number of interfaces to outside callers or the number of dangerous instructions in the code (Howard, Pincus and Wing, 2003).

These proposals are measurements in the sense that they deliver quantitative results. It is open to debate whether they really measure security. How relevant is the number

of security flaws? It is sufficient for an attacker to find and exploit a single flaw to compromise security. It is equally open to debate whether such metrics could be the basis for a meaningful security comparison of products, given that it is rare to find two products that serve exactly the same purpose. It has therefore been suggested that these metrics are best treated as *quality* metrics and used for monitoring the evolution of individual products.

Security metrics for a system could look at the actual configurations of the products deployed. In a system with access control features, we could look at the number of accounts with system privileges or the number of accounts with weak passwords. In a networked system, we could look at the number of open ports or at the services accessible from outside and whether the currently running versions have known vulnerabilities. Such attributes certainly give valuable *status information* but do not really give the quantitative results desired from a measurement.

Specifically for computer networks, we could measure the connectivity of nodes in a network to assess how quickly and how far attacks could spread. We could also measure the time services are unavailable after an attack, or predict recovery times and cost of recovery for a given configuration and class of attacks.

In an alternative approach, we could try to measure security by measuring the cost of mounting attacks. We could consider:

- the time an attacker has to invest in the attack, e.g. analyzing software products;
- the expenses the attacker has to incur, e.g. computing cycles or special equipment;
- the knowledge necessary to conduct the attack.

However, the cost of discovering an attack for the first time is often much larger than the cost of mounting the attack itself. Today, *attack scripts* are readily available so that attacks can be launched with very little effort or knowledge of the system vulnerabilities being exploited.

As yet another alternative, we could focus on the assets in the given system and measure the risks these assets are exposed to. Section 1.4.4 gives an overview of risk and threat analysis. In summary, desirable as security measurements are, we have at best metrics for some individual aspects of security and the search for better metrics is still an open field of research.

### 1.3.3 Standards

Prescriptive security management standards that stipulate which security measures have to be taken in an organization exist for specific industry branches. Typical examples

are regulations for the financial sector<sup>1</sup>, or rules for dealing with classified material in government departments<sup>2</sup>.

Other management standards are best described as codes of best practice for security management. The most prominent of these standards is ISO 17799 (International Organization for Standardization, 2001). It is not a technical standard for security products or a set of evaluation criteria for products or systems. The major topics in ISO 17799 are as follows.

- Establishing organizational security policy: this document provides management direction and support on security matters.
- Organizational security infrastructure: responsibilities for security within an enterprise have to be properly organized. Management has to be able to get an accurate view of the state of security within an enterprise. Reporting structures should facilitate efficient communication and implementation of security decisions. Security has to be maintained when information services are being outsourced to third parties.
- Asset classification and control: to know what is worth protecting, and how much to spend on protection, an enterprise has to have a clear picture of its assets and of their value.
- Physical and environmental security: physical security measures (fences, locked doors, etc.) protect access to business premises or to sensitive areas (rooms) within a building. E.g. only authorized personnel get access to server rooms. These measures can prevent unauthorized access to sensitive information and theft of equipment. The likelihood of natural disasters can depend on environmental factors, e.g. is the area subject to flooding?
- Personnel security: your own personnel or contract personnel can be a source of insecurity. There should be procedures for new employees joining and for employees leaving (e.g. collect keys and entry badges, delete user accounts of leaving members.) Enforced holiday periods can stop staff hiding the traces of fraud they are committing. Background checks on new hires can be a good idea. In some sectors those checks may be required by law, but there may also be privacy laws that restrict which information an employer may seek about its employees.
- Communications and operations management: the day-to-day management of IT systems and of business processes has to ensure that security is maintained.
- Access control: access control can apply to data, services and computers. Particular attention should be applied to remote access, e.g. through the Internet or dial-in connections. Automated security policies define how access control is being enforced.

<sup>1</sup>E.g. the Payment Card Industry (PCI) Data Security Standard supported by Visa.

<sup>2</sup>E.g. the US policy stating that the encryption algorithm AES can be used for top-secret data with 192-bit or 256-bit keys (CNSS, 2003).

- Systems development and maintenance: security issues should be considered when an IT system is being developed. Operational security depends on proper maintenance (e.g. patching vulnerable code, updating virus scanners). IT support has to be conducted securely (how do you deal with users who have forgotten their password?) and IT projects have to be managed with security in mind (who is writing sensitive applications, who gets access to sensitive data?).
- Business continuity planning: put measures in place so that your business can cope with major failures or disasters. Measures start with keeping backups of important data kept in a different building and may go on to the provision of reserve computing facilities in a remote location. You also have to account for losing key staff members.
- Compliance: organizations have to comply with legal, regulatory and contractual obligations, as well as with standards and their own organizational security policy. The auditing process should be efficient while trying to minimize its interference with business processes. In practice, these aspects often pose a greater challenge than fielding technical security measures.

Achieving compliance with ISO 17799 can be quite an onerous task. The current state of your organization vis-à-vis the standard has to be established and any shortcomings identified have to be addressed. There exist software tools that partially automate this process, again applying best practice, only this time ensuring compliance with the standard.

## 1.4 RISK AND THREAT ANALYSIS

Many areas of engineering and business have developed their own disciplines and terminology for risk analysis. This section will give a brief overview of risk analysis for IT security. Within IT security, risk analysis is being applied:

- comprehensively for all information assets of an enterprise;
- specifically for the IT infrastructure of an enterprise;
- during the development of new products or systems, e.g. in the area of software security.

Informally, risk is the possibility that some incident or attack can cause damage to your enterprise. An attack against an IT system consists of a sequence of actions, exploiting weak points in the system, until the attacker's goals have been achieved. To assess the risk posed by the attack we have to evaluate the amount of damage being done and the likelihood of the attack occurring. This likelihood will depend on the attacker's motivation and on how easy it is to mount the attack. In turn, this will further depend on the security configuration of the system under attack.

To disentangle the various strands of investigations that have to be pursued in the process of risk analysis, we will refer to *assets*, *vulnerabilities* and *threats*, and calculate risk as a

function thereof. Informally:

$$Risk = Assets \times Threats \times Vulnerabilities.$$

In the process of risk analysis, values are assigned to assets, vulnerabilities and threats. In *quantitative* risk analysis, values are taken from a mathematical domain like a probability space. For example, by assigning monetary values to assets and probabilities to threats the expected loss can be calculated. In *qualitative* risk analysis, values are taken from domains that do not have an underlying mathematical structure. Risk is calculated based on rules that capture the consolidated advice of security experts.

### 1.4.1 Assets

First, assets have to be identified and valued. In an IT system, assets include:

- hardware: laptops, servers, routers, PDAs, mobile phones, smart cards etc.;
- software: applications, operating systems, database management systems, source code, object code etc.;
- data and information: essential data for running and planning your business, design documents, digital content, data about your customers etc.;
- reputation.

Identification of assets should be a relatively straightforward systematic exercise. Valuation of assets is more of a challenge. Some assets, such as hardware, can be valued according to their monetary replacement costs. For other assets, such as data and information, this is more difficult. If your business plans are leaked to the competition or private information about your customers is leaked to the public you have to account for indirect losses due to lost business opportunities. The competition may underbid you and your customers may desert you. Even when equipment is lost or stolen you have to consider the value of the data stored on it, and the value of the services that were running on it. In such situations, assets can be valued according to their importance. As a good metric for importance, ask yourself how long your business could survive when a given asset has been damaged: a day, a week, a month?

### 1.4.2 Vulnerabilities

Vulnerabilities are weaknesses of a system that could be accidentally or intentionally exploited to damage assets. In an IT system, typical vulnerabilities are:

- accounts with system privileges where the default password, such as 'MANAGER', has not been changed;
- programs with unnecessary privileges;
- programs with known flaws;
- weak access control settings on resources, e.g. having kernel memory world writable;
- weak firewall configurations that allow access to vulnerable services.

*Vulnerability scanners* provide a systematic and automated way of identifying vulnerabilities. Their knowledge base of known vulnerabilities has to be kept up to date. Organizations like SANS or Computer Emergency Response Teams (CERTs) provide this information, as do security advisories of software companies.

Vulnerabilities can be rated according to their impact (level of criticality). A vulnerability that allows an attacker to take over a systems account is more critical than a vulnerability that gives access to an unprivileged user account. A vulnerability that allows an attacker to completely impersonate a user is more critical than a vulnerability where the user can only be impersonated in the context of a single specific service. Some scanners will also give a rating for the vulnerabilities they detect.

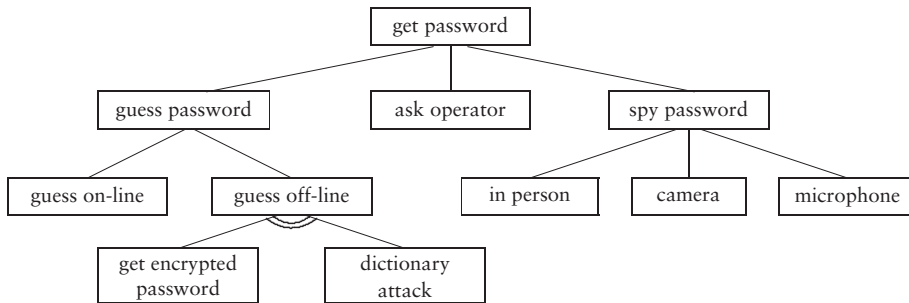
Terminology in IT security is notoriously imprecise, and you might find vulnerability scanners that are marketed as risk analysis tools. It is of course perfectly reasonable to use a different conceptual framework than the one sketched here, so the burden is on you to find out what any ‘risk analysis tool’ is actually offering, and then place it in the framework of your choice.

### 1.4.3 Threats

Threats are actions by adversaries who try to exploit vulnerabilities to damage assets. There are various ways to identify threats. We can categorize threats by the damage done to assets. For example, Microsoft’s STRIDE threat model for software security lists the following categories (Howard and LeBlanc, 2002).

- Spoofing identities: the attacker pretends to be somebody else.
- Tampering with data: e.g. security settings are changed to give the attacker more privileges.
- Repudiation: a user denies having performed an action like mounting an attack, or making a purchase.
- Information disclosure: information may lose its value if it is disclosed to the wrong parties (e.g. trade secrets); your organization may face penalties if it does not properly protect information (e.g. personal information about individuals).
- Denial of service (DoS): DoS attacks can make web sites temporarily unavailable; there have been stories in the press that businesses use such attacks to harm competitors.
- Elevation of privilege: a user gains more privileges on a computer system than he/she is entitled to.

Then we can identify the source of attacks. Would the adversary be a member of your organization or an outsider, a contractor or a former member? Has the adversary direct access to your systems or is the attack launched remotely?



○ Figure 1.1: Attack Tree for Obtaining Another User's Password

We can also analyze how an attack is executed in detail. An attack may start with innocuous steps, gathering information needed to move on to gain privileges on one machine, from there jump to another machine, until the final target is reached. To get a fuller picture of potential threats, a forest of *attack trees* can be constructed. The root of an attack tree is a generic attack. The nodes in the tree are subgoals that must be achieved for the attack to succeed. Subgoals can be broken into further subgoals. There are AND-nodes and OR-nodes. To reach an AND-node, all subgoals have to be achieved. To reach an OR-node, it is enough if one subgoal is achieved. Figure 1.1 gives a basic attack tree for the attack 'get password'. A password can be obtained by guessing, or by tricking an operator to reveal it, or by spying on the user. Guessing could be on-line or off-line. For off-line guessing, the attacker needs the encrypted password and has to perform a dictionary attack. The attacker could spy on the victim in person (so-called shoulder surfing), direct a camera at the keyboard or direct a microphone at the keyboard and distinguish the keys pressed by sound.

It is possible to assign values to the edges in an attack tree. These values can indicate the estimated cost of an attack, the likelihood that it will occur, the likelihood that it will succeed or some other aspect of interest. From these values, the cheapest attack, or the most likely attack, or the attack most likely to succeed can be computed.

Attack trees are thus a formalized and structured method for analyzing threats. Threat assessments become reproducible as the overall assessment of a threat can be traced to the individual assessments of subgoals. If the final result appears implausible, the tree can be consulted to see which subgoals were most critical for the final result, and those individual valuations may be adjusted to more 'sensible' values. This remark explains why the construction of attack trees is more an art than a science. You need experience to know when to readjust your ratings for subgoals, and when to adjust your preconceived opinion of the severity of a threat. You also need experience to know when to stop breaking up subgoals into ever more subgoals, a phenomenon known in the trade as *analysis paralysis*.

Threats can be rated according to their likelihood. The likelihood depends on the difficulty of the attack, on the motivation of the attacker and on the number of potential attackers. *Attack scripts* automate attacks, making it easy to launch the attack. They are also likely to be available to a larger set of attackers. Hence, such attacks would be rated more likely than an individual hand-crafted attack.

#### 1.4.4 Risk

Having rated the value of assets, the criticality of vulnerabilities and the likelihood of threats, we now face the tricky task of calculating our risks.

In quantitative risk analysis, expected losses could be computed in the framework of probability theory, based on monetary values for the assets and probabilities for the likelihood of threats. Such a method has the pleasing feature of being based on a well-established mathematical theory, but also has the considerable drawback that the ratings we obtain are often based on educated guesses. In short, the quality of the results we obtain cannot be better than the quality of the inputs provided. We could consider other mathematical frameworks, such as fuzzy theory, to make some provisions for the imprecise nature of our ratings. There are areas of risk analysis where quantitative methods work, but more often the lack of precision in the inputs does not justify a mathematical treatment.

In qualitative risk analysis:

- assets could be rated on a scale of critical – very important – important – not important;
- criticality of vulnerabilities could be rated on a scale of has to be fixed immediately – has to be fixed soon – should be fixed – fix if convenient;
- threats could be rated on a scale of very likely – likely – unlikely – very unlikely.

A finer granularity of scaling could be provided, e.g. numerical values from 1 to 10. Whatever scheme is used, guidance has to be given on how to assign ratings. The mapping of the ratings for assets, vulnerabilities and threats to risks is often given by a table drawn up to reflect the judgment of security experts. The DREAD methodology that complements STRIDE may serve as an example of a scheme for qualitative risk analysis (Howard and LeBlanc, 2002).

- **Damage potential:** relates to the values of the assets being affected.
- **Reproducibility:** one aspect of how difficult it is to launch an attack; attacks that are easy to reproduce are a greater risk than attacks that only work in specific circumstances.
- **Exploitability:** relates to the effort, expertise and resources required to launch an attack.

- **Affected users:** for software vendors, another important contributing factor to damage potential.
- **Discoverability:** when will the attack be detected? In the most damaging case, you will never know that your system has been compromised. (In World War II, German intelligence refused to believe that many of their encryption schemes had been broken.)

#### 1.4.5 Countermeasures – Risk Mitigation

The result of a risk analysis is a prioritized list of threats, together with recommended countermeasures to mitigate risk. Risk analysis tools usually come with a knowledge base of countermeasures for the threats they can identify.

It might seem trivially true that one should first go through a risk analysis before deciding on which security measures to implement. However, there are two reasons why this ideal approach may not work. Conducting a risk analysis for a larger organization will take time, but the IT system in the organization and the world around will keep changing. So, by the time the results of the analysis are presented, they are already somewhat out of date. Moreover, the costs of a full risk analysis may be difficult to justify to management.

For these reasons, organizations may opt for *baseline protection* as an alternative. This approach analyzes the security requirements for typical cases and recommends security measures deemed adequate. One of the best known IT security baseline documents is maintained by the German Information Security Agency (BSI, 2003).

### 1.5 FURTHER READING

Anderson's book on security engineering gives an excellent insight into the full extent of the challenges faced in security (2001). A good discussion of security management, and of IT security in general, can be found in Smith (1993). A discussion on the various meanings of the term security policy is given by Sterne (1991). The observations on defining security policies in commercial organizations made by Martin Smith (1993) are still valid today. The management of information security risks is discussed by Alberts and Dorofee (2003).

### 1.6 EXERCISES

**Exercise 1.1** Discuss further options for measuring the security of products and systems.

**Exercise 1.2** On the computing system you are using, identify the software components that potentially could incorporate security mechanisms.

**Exercise 1.3** Should a risk analysis of a computer center include flooding damages to computing equipment even when the center is in a high and dry location?

**Exercise 1.4** Conduct a risk and threat analysis for a mobile phone service, taking into account that calls are transmitted over a radio link between mobile phone and base station, and that with international roaming a subscriber can use the service in so-called visited networks when traveling abroad. Consider the subscribers' and the network operators' viewpoints in your analysis.

**Exercise 1.5** Bank customers can withdraw cash from automated teller machines (ATMs) using a cash card and a personal identification number (PIN). Conduct a risk and threat analysis for this application, both from the customers' and the banks' viewpoints.

**Exercise 1.6** Consider the theft of a central server from a university department. Which assets could be damaged if this happens? Construct an attack tree for this threat.

