Introduction to Computer Security

Conference Paper · January 1997

DOI: 10.1007/3-540-49248-8_19 · Source: DBLP

Bart De Decker
KU Leuven

Introduction to Computer Security

Bart De Decker

K.U.Leuven, Department of Computer Science,
Celestijnenlaan 200A, B-3001 Leuven, Belgium
Bart.DeDecker@cs.kuleuven.ac.be

Abstract. The nineties set off the "information age". Companies, organisations, the whole society have become utterly dependent on computers for their proper functioning. Since information gathering, processing and distributing have become so important, it should be treasured as a strategic asset, and therefore, properly protected. In this paper, we first focus on the security policy. Then we examine the major threats that may compromise the security of information systems. Finally, we present an overview of security measures is presented.

1 Introduction

The nineties set off the "information age". Companies, organisations, the whole society have become utterly dependent on computers for their proper functioning. However, although the efficiency has improved rapidly, in many organisations, the condition of computer security has never been so poor. Actually, it is worsening, and this for several reasons.

- During the last decade, the information technology has evolved very fast. Security was not an issue in the beginning, and has never been able to keep pace.
- At first, computers were used in invoicing and wages administration. Later on, they were involved in stock management, EDI, ... Nowadays, they are a necessary tool for strategic decisions.
- Networking is booming. The central mainframe has been replaced by a LAN of PCs or workstations. Because of the pressure of the market, these LANs are connected to the Internet or the public telephone network.
- The *democratization* of information technology moved the computers out of the computing centre into the work place.

One aspect of democratization, the **lowering of thresholds**, needs some elaboration. Although security incidents are often (wrongly) associated with break-ins by hackers or criminal organisations, most incidents are caused by **insiders** (employees of the organization, maintenance people, ...). (See also tab. 1.) This is not surprising, since insiders have less hurdles to take and possess inside information. Moreover, a computerized office is an attractive target for fraud:

B. Preneel, V. Rijmen (Eds.): COSIC'97 Course, LNCS 1528, pp. 377–393, 1998.

- there is very little exposure; if well prepared, a security breach takes less than a second;
- controls –if present– are performed by computers, not by people;
- most information is centralized and available from the PC in the office;
- the fraud can be repeated over and over again; one does not have to be greedy: ten thousand times one hundred equals one million, and will probably be less detectable;
- the fraud can be automated, without any human intervention

Beside incidents caused willfully by insiders, there are three other classes of incidents. See tab. 1.

Table 1. Frequency of Security Incidents

Frequency	**Reason**
50–60%	Errors *due to inexperience, non-chalance, panic reaction, ...*
15–20%	Insiders *i.e. (former) employees, maintenance people, ...*
10–15%	Disasters *such as flooding, lightning stroke, fire, ...*
3–5%	Outsiders *i.e. hobbyist, hackers' club, competitor, organised crime (foreign) intelligence agency, ...*

The high percentage of **errors** that constitute a security breach, stems from the fact that more unexperienced users are working with computers; some of them can barely type or lack any IT-knowledge. These unexperienced users can be very harmful if the system itself is not sufficiently protected. Also, users can panic when confronted with a break-in, thereby aggravating the security breach, instead of stopping it. Finally, by moving computers into the work place, accidents will happen more frequently: a server-machine in the office may seem an ideal place for a plant; however, plants need watering, and few servers will survive spilt water.

Disasters are fire, flooding, explosions, lightning strokes, storm, but also major hardware failures, etc. These are hardly or not at all accounted for. Many companies will not even survive a situation where most of the IT-infrastructure is destroyed, because there is no backup-site that can take over the data processing.

Outsiders range from the computer hobbyist, who gets a kick from breaking into other computers, to competitors who are interested in your secret research results or in your sale's strategy (industrial espionage), to national or foreign intelligence agencies. However, one can expect an increasing amount of break-ins caused by criminal organisations (the mob) who will try to subvert the computing infrastructure in order to bribe the company later or because they have been hired by the competition.

2 Security Policy

Vast amounts of resources are being spent on "securing" the computer infrastructure. Every time a major security breach appears on the front page, some countermeasures are hastily installed. One can hardly expect any security without a policy. Every organisation should spend enough time and resources on defining a security policy and on implementing the necessary measures.

The *security policy* should at least treat the following topics:

- the general objective; this serves as the justification of the policy;
- the importance of information technology for the organization;
- the limited validity of the policy; it should be short term; this ensures that the policy will be reviewed and adapted every year or two;
- the allocation of sufficient resources (budget and personnel);
- the specific objectives, requirements and responsibilities.

The implementation of a security policy will only succeed if the policy is endorsed by top management.

2.1 Information Quality

In an organization, there are several information flows; some are more valuable than others. They can be characterized by the following quality labels:

- Some information is **confidential**. For instance research results should be kept secret for the competition, but also the law enforces the protection of the privacy of the individual.
- **Integrity** deals with the reliability of the information. This means that the information is *correct* and *authentic*. In a network environment, this means that the information has not been tampered with, and is no replay of a previous communication. Some applications (e.g. electronic commerce) will even require that sender, (or receiver) cannot repudiate the date sent (or received).
- Information should be **available** when needed. Although often overlooked, availability also involves the timely processing and distribution of the information. Denial of service attacks, which are in general very difficult to counter, can jeopardize the continuity of the processing and hence, the survival of the organization.

In order to qualify the information, the users of the computer infrastructure should be interrogated. Questions to be answered include:

1. Questions concerning the **confidentiality**:
   - "What are the consequences (financial, legal, ...) if this information falls into the wrong hands (colleague, competitor, press, ...)?

  - ”Who will benefit from this information? (competition, press, foreign intelligence agency, . . .)”
  - . . .
2. Questions concerning the **integrity**:
  - ”What happens when the data is erroneous, incomplete or obsolete?”
  - ”What will happen if information is lost, replayed, or delayed?”
  - ”What errors can be introduced deliberately?”
  - ”What are the weakest issues?”
  - ”Who will benefit from these errors?”
  - . . .
3. Questions concerning the **availability**:
  - ”How IT-dependent is the unit?”
  - ”What are the time-critical aspects?”
  - ”What is the expected response time?”
  - ”What are the implications (financial, legal, . . .) if the system breaks down for . . . hours/days?”
  - ”When are most queries/processing done?”
  - . . .

2.2 Risk Analysis

When the security policy has been formulated, it should be implemented. The policy itself specifies *what* should be protected, but does not impose any measures. Before any security plan is drawn up, one needs to know what are the most likely security breaches to occur, and what implications are involved. Usually, this is determined through risk analysis. (See also fig. 1.)

The inputs for the risk analysis process are:

- the security policy,
- the possible adversaries (insiders, competitors, press, . . .)
- the known weaknesses of the IT-infrastructure,
- possible threats.

Table 2. Losses should be all-inclusive

	Direct losses	**Indirect losses**
Material losses	wages overtime . . .	overdue payment . . .
Immaterial losses	frustration . . .	negative publicity loss of goodwill . . .

For each threat, the *probability* of occurrence is determined (often, one has to rely on an educated guess). Then the implications of the threat are examined:

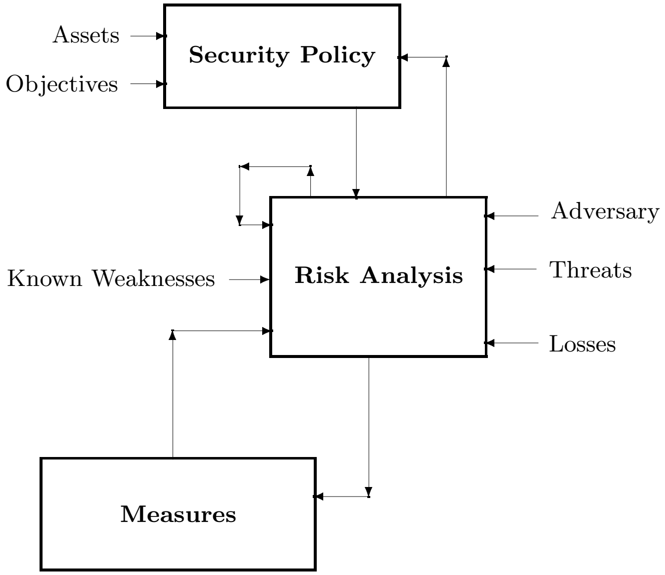


Fig. 1. The Risk Analysis Cycle

”what are the potential losses if the threat happens?” The *losses* should not only include time and money spent to undo the effects of the threat, but also financial and legal implications of a possible lawsuit, the effects of a bad press, loss of goodwill, etc. (see also tab. 2). Finally, the *risk* is calculated according to the following equation:

$$\mathbf{Risk} = \mathbf{Probability} \times \mathbf{Loss}$$

Figure 1 shows that several iterations will be necessary before a conclusion can be drawn. Also, it might be necessary to adapt the security policy. The threats with highest risk should be countered first. A set of appropriate measures will be assembled. These measures will be a mixture of **physical protection** and **technical measures**, and **procedural measures**. Obviously, the proposed measures should not cost more than the threat they counter. See also sec. 4.

The risk-analysis should be reconducted every time the policy changes or a major security incident occurs. Probabilities and risks should be revised.

2.3 Security versus User-Friendliness

There is no system that is 100% safe, except one that is switched off and kept in a bunker. Many security measures make the system less user-friendly. If the users are not convinced of the usefulness of the measure, they will subvert it, one way or the other. Humans are often the weakest link. Through continuous education, the users are kept vigilant and aware of their responsibility towards the overall security.

3 Threats

In this section, the malicious security incidents are classified, and some measures are presented.

Table 3 gives a taxonomy[10] of the major threats. The left column indicates the typical steps and modes of intended use of computer systems. The right column involves misuse.

Table 3. Taxonomy of security threats

normal intended use	**misuse**
access to the computer system	external misuse
use of the computer system	hardware misuse
apparently authorized use	masquerading
direct use	pest programs for deferred misuse
use apparently conforming with intended controls	bypass of intended controls
active use	active misuse of resources
apparently normal use	passive misuse of resources
apparently proper use	misuse resulting from inaction
proper use	use as an aid to other misuse

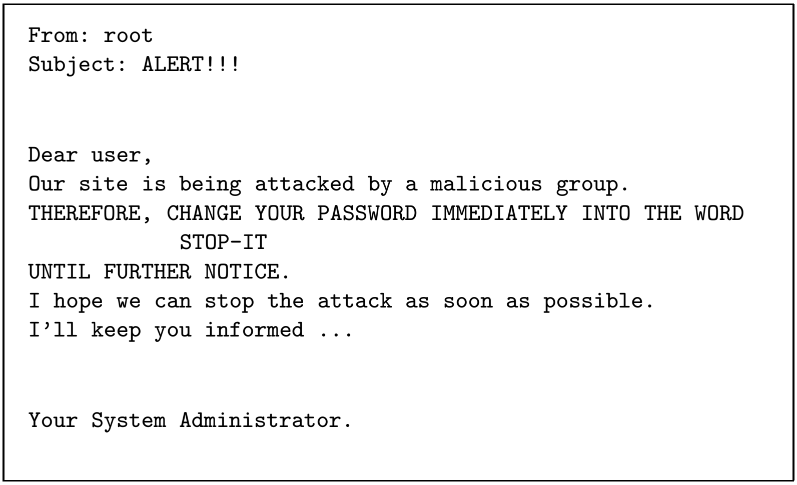
3.1 External Misuses

External misuse refers to threats that do not require physical access to the computer system or network. It is not difficult to look over one's shoulder and observe the keystrokes (for instance, when the password is being entered). How often have users posted their account-numbers and passwords on their terminal? The contents of a computer screen can be copied from a distance (e.g. in a van parked outside the building) through a device that can capture and visualize the electro-magnetic radiation of the screen. These examples can be summarized as *visual spying*.

Other threats in this class include *deceiving users and operators*, also called *social engineering*. Unexperienced users can easily be tricked into performing foolish actions: a forged phone call or forged e-mail messages (supposedly coming from the system operator) mentioning a major break-in, and asking the recipient to change his password into a specific word; see figure 2.

Operators are often willing to respond to a phone call from a user who has forgotten his password *without any verification of the identity of the caller*. Also, they give the superuser-password over the phone to someone who mispresents himself as a maintenance person, etc.

Finally, searching waste baskets for printouts will often give an attacker a wealth of information that can be used in further attacks.



```
From: root
Subject: ALERT!!!


Dear user,
Our site is being attacked by a malicious group.
THEREFORE, CHANGE YOUR PASSWORD IMMEDIATELY INTO THE WORD
            STOP-IT
UNTIL FURTHER NOTICE.
I hope we can stop the attack as soon as possible.
I'll keep you informed ...


Your System Administrator.
```

Fig. 2. Forged e-mail can trick users into doing foolish actions

3.2 Hardware Misuse

It is often forgotten to erase disks, tapes, cassettes, ..., before they are discarded. The erasure should be done thoroughly by overwriting these media with innocuous data.

Eavesdropping on communication lines is not very difficult, especially on LANs, which are mostly broadcast media. Several public domain *network sniffers* are available. Some of them can be configured to show only the first hundred bytes of a telnet (ftp, ...) session. Since passwords are sent in cleartext, such a sniffer can capture quite a few account-password pairs in a very short time.

Electronic jamming can cause serious interference on the network, and initiate a *denial of service attack*. Furthermore, disgruntled employees may damage or modify the equipment; if the power supply or cooling is interrupted or sabotaged, the IT-infrastructure comes to a grinding halt.

Finally, since most computers and storage media are small, they are easily removable. Theft might become a serious problem, if the physical access to the building is not strictly controlled.

3.3 Masquerading

Once passwords or other authentication means have been captured, they can be used to masquerade as somebody else. Seizing passwords is not that difficult. Humans are known to pick bad passwords; several studies have shown that 25% of the passwords can be guessed easily [3]. Moreover, passwords can also be detected through visual spying or social engineering (sec. 3.1), eavesdropping on the network (sec. 3.2), installing login-spoofs (sec. 3.4) or conducting a dictionary attack (sec. 3.9). Finally, most systems come with pre-installed *universal accounts* that are protected with the same password!

In *piggy-backing attacks*, the attacker gains physical access to communication lines or workstations. Unattended logged-in terminals, with unlocked keyboards, are also a prime target. Commands entered (or inserted in the communication stream) will be executed on behalf of the logged-in user.

Playback attacks merely repeat captured conversations or messages.

In a world-wide network environment, attacks may be very difficult to trace to the originator, if the latter is clever enough to use many different systems: his physical whereabouts will be completely masked. This kind of attack is sometimes called *network weaving.*

Computers linked by LANs typically use trust to enhance the user-friendliness. That way, users are only required to authenticate to one of the trusted machines. From then on, they can access any resource on any other machine without any further authentication. *Spoofing attacks* try to exploit this trust. If the attacker succeeds in making a server-machine believe the request came from a trusted host, the server will act upon the request.

One of the latest examples of masquerading, is *Web spoofing* [5]: it allows an attacker to create a ”shadow copy” of the entire World Wide Web. Accesses to the shadow Web are funnelled through the attacker's machine, allowing the attacker to monitor all of the victim's activities including any passwords or account numbers the victim enters. Moreover, the attacker can send misleading or modified data to Web servers in the victim's name, or to the victim in the name of any Web server. See also fig. 3.

3.4 Pest Programs

By installing **pest programs**, the attacker tries to set up opportunities for further misuse. Table 4 gives short overview of the different kinds of malicious software.

A **Trojan horse** is a program that does something unexpected. For instance, it clandestinely copies data, it reformats the disk, it disables the system, etc. If

Table 4. Pest Programs

Type	Explanation
Trojan Horse	Program that does something unexpected (and often secretly)
Mule (Spoof)	Program that mimics another program
Rabbit	Program that overloads a system
Worm	Program that replicates itself through the network
Virus	Program fragment that, when executed, attaches itself to other programs

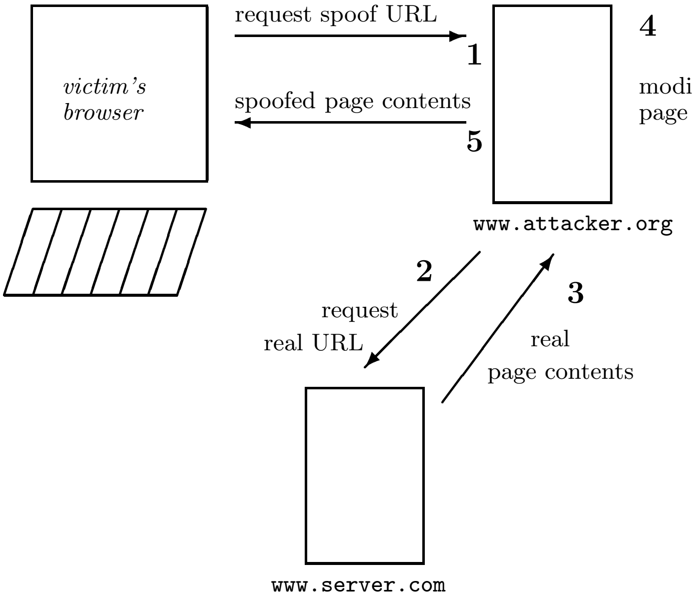


Fig. 3. Web Spoofing: An Example Web Transaction

the unexpected behaviour manifests itself only when a certain condition (date) is met (reached), the program is said to contain a *logic (time) bomb*. There are many cases known where a system administrator replaced a program by a Trojan horse, that tested the presence of the administrator's name in the password file. If the name disappeared, the program did something harmful.
It can be proved that there exist no algorithm that can decide whether a program is Trojan or not. Moreover, it is not sufficient to scrutinize the source code of a program, since a Trojan compiler may have inserted Trojan instructions in compiled program.

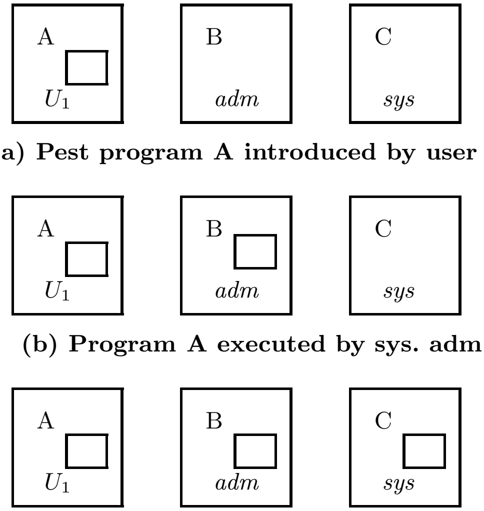
A **mule** is a program that mimics another program, but does something completely different. The classic example is a login-spoof, that behaves like the login-program (i.e. it reads an account name and password), and then prints an error message saying `"login incorrect"`. The account name and password are mailed to the attacker.

Rabbits are programs that continuously fork new processes. Hence, the system gets overloaded, and will eventually be completely locked or crash. This is a special case of a *denial of service* attack.

A **worm** is a program that replicates itself through the network. The program may be malicious or it may be used constructively to provide extensive multiprocessing.

Viruses replicate themselves by attaching their code to other programs. An infected program becomes a Trojan horse, since when it is executed, it will try to infect other programs, and possibly do something harmful. A virus does not necessarily consist of machine-executable code. Many programs (such as spreadsheets, word processors) can execute macros included in their documents. Since these macro-languages allow for reading and writing files, an infectious macro is easily developed, and inserted in a document. MIME[1] is great to send infected documents to the whole world!

The setting up of pest programs may employ other misuses. An insider may easily install such a program (possibly unknowingly). Although most systems provide some sort of access control to their resources, this limited access does not prevent the spreading of malicious software. For instance, if the system administrator executes an infected program (e.g. game), he will first infect his own programs. Later on, when the administrator executes one of his (now infected) programs with super-user privileges, he will infect the whole system. See fig. 4: the small box inside each file represents the viral code.



(a) Pest program A introduced by user U_1

(b) Program A executed by sys. adm.

(c) Prog. B executed by sys. adm. with special priv.

Fig. 4. The proliferation of a virus

3.5 Bypass-Attacks

By using existing flaws in programs, authentication can be avoided.

A *trapdoor* is an entry path that is not normally expected to be used. In a few cases, it was implemented for debugging purposes, and afterwards completely forgotten. E.g. the `sendmail`-program hid a `DEBUG` command, which was exploited by the Internet worm [11].

[1] Multipurpose Internet Mail Extensions.

Some trapdoors are software bugs. There are plenty examples: the finger-daemon (exploited by the same worm), the talk daemon, etc. Many servers do not check their inputs. An attacker can send 'unexpected' data (e.g. of the wrong format or oversized) or execute the server-program in a specially made environment (e.g. with different 'space'-characters, or different dynamically linkable libraries, . . .). The oversized data will provoke an overflow of the buffer, which is allocated on the stack. Hence the stack will become garbled (important book-keeping information, such as the return addresses of the procedure calls, will be modified). That way, the attacker can make the server-program do whatever he likes. See also fig. 5.

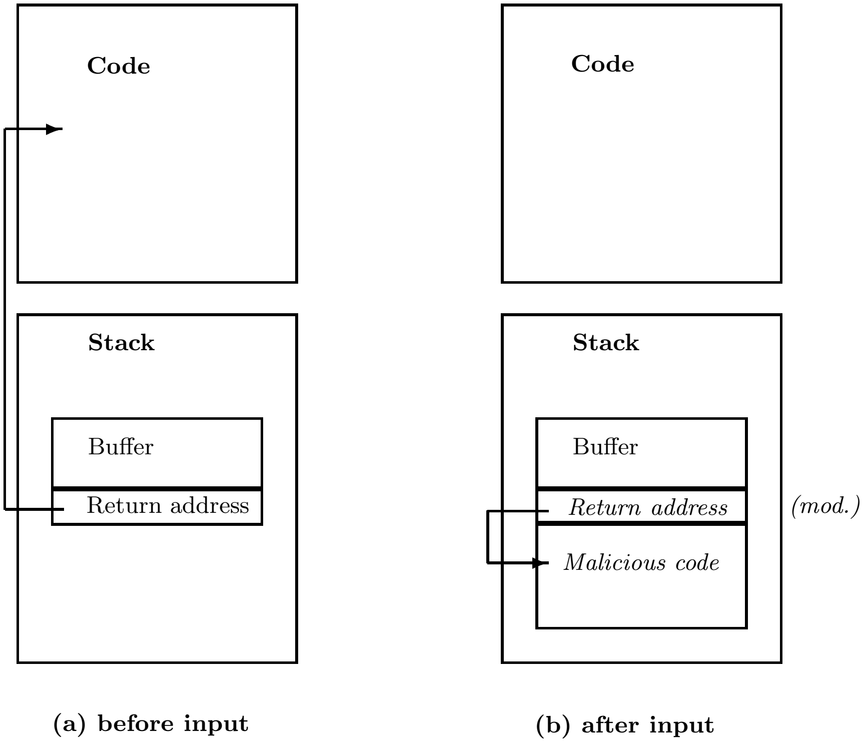


Fig. 5. Buffer overflow may cause malicious code to be executed

3.6 Active Misuse

When the attacker has access to the system with the proper privileges, he can change the state of the system or modify data. Any kind of malicious actions for which the attacker has enough authority can be performed, such as: creating or deleting data, denying or delaying service to other users, entering false or misleading data.

A special form of active misuse is the so called *salami attack*, in which numerous small pieces (e.g. a few BEFs on every transaction), are collected for personal or corporate gain.

Denial of service is one of the most difficult attacks to deal with. Usually, the attacker tries to saturate a communication line or a server (by sending millions of packets) or tries to exhaust a resource (e.g. by sending thousands of huge e-mail messages, that fill the local disk).

3.7 Passive Misuse

Passive misuse involves reading information with apparent proper authorization. It is clear that confidential information can fall into the wrong hands.

In this stage, the attacker hunts for interesting data through random browsing or selective searching. The curiosity of humans is often so strong, that *bait systems*[2] can lure a computer cracker long enough for the operator to trace the felon.

Some databases do not answer queries that pertain to one specific case or person. However, by selecting the right set of queries, it is often possible to infer the wanted information. Also, traffic analysis of possibly encrypted conversations can often reveal interesting information.

Finally, *covert channels* can be used to subvert a system that disables the flow of information from a privileged user to an unprivileged user. For instance, one bit of information can be signalled to an unprivileged user on the basis of whether or not a shared resource (e.g. disk) is exhausted or not.

3.8 Inactive Misuse

Inactive misuse is a typical incident where an insider does not perform a task for which he/she is in charge. Some examples:

- the backup is not or only partially taken,
- the account of a former employee is not removed,
- accounts that come pre-installed on a system, are not disabled,
- old disks, tapes, cassettes are not erased before being disposed of,
- the logs are not checked regularly,
- a terminal, on which a user is logged in, is left unattended, while the keyboard is unlocked,
- ...

[2] A bait system is a computer that contains fake top-secret information.

3.9 Indirect Misuse

Indirect misuse relates to all actions that are performed for later misuse. Some typical examples include:

- pre-encrypting of data (in order to be able to break a ciphertext),
- dictionary attack on a captured password file,
- scanning telephone numbers of computers by using an autodialler,
- ...

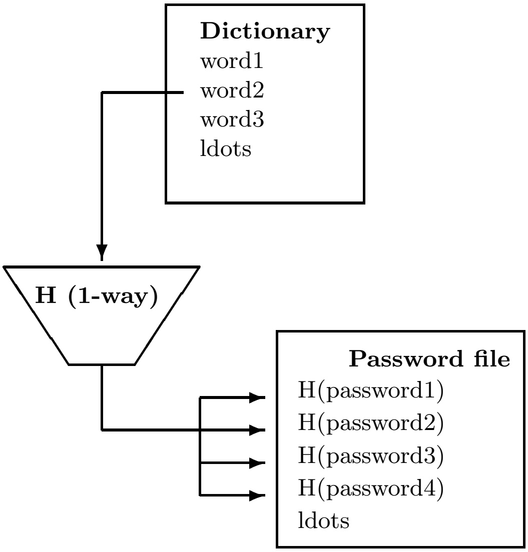


Fig. 6. The dictionary attack

In a *dictionary attack*, one attempts to identify which dictionary words are used as passwords. Usually, passwords are not stored in readable form in the password file, but are transformed through a one-way function. It is not difficult to apply the same function to every word of the dictionary and compare it with all the values found in the password file (see fig. 6). The same attack is also possible in all situations where passwords are used as cryptographic key, as long as the encrypted plaintext is recognizable (e.g. contains readable text, a timestamp, a network address, ...). For instance, in the Kerberos authentication system [9,4], the ticket granting ticket (TGT) is sent to the user encrypted with a key derived from a password. Since the TGT contains IP-numbers, it is susceptible to a dictionary attack.

4 Measures

It is difficult to give a complete overview of measures that can be taken. Often, one measure will not suffice to counter a threat. On the other hand, some measures have an impact on many threats.

Security measures will in general reduce the probability that certain threats occur, and/or limit the possible losses. They can be preventive, detective or corrective. Also, losses can be insured with an insurance company.

The measures can be categorized in three different classes:

- physical protection,
- technical (logical) measures,
- organizational procedures.

Table 5 gives an brief overview of the different kinds of measures. The following subsections illustrate the different classes.

Table 5. Classification of Measures

	Physical Protection	**Technical Measures**	**Organizational Measures**
Preventive	... guard at entrance ...	... firewalls ...	... education/training ...
Detective	... motion detector ...	... Logs ...	... call-back procedure ...
Corrective	... UPS ...	... anti-virus monitor ...	... backup ...

4.1 Physical Protection

Physical protection deals with the physical access to buildings, hardware and media:

- **protection of the building**
  measures against natural disasters, assaults, unwanted visitors, ...
- **protection of the hardware**
  measures against theft, vandalism, sabotage; measures limiting physical access to certain rooms; provision of spare parts (computers, ...) and backup-site, etc.
- **protection of the data (media)**
  measures for the protection of removable media (disks, tapes, CD-ROMs, ...), and —often forgotten— the backup media, etc.

- **protection of the utilities**
  measures for the protection of power supply through *UPS*[3] and for the protection of communication channels, etc.

4.2 Technical Measures

Technical (or logical) measures are usually software solutions (or a combination of hard- and software):

- **logical access control**
  protects the internal resources, limits the user's capabilities;
- **authenticated login-session**
  protects the access to the system; it may involve cryptographic protocols, a call-back mechanisms, etc.;
- **logs**
  can provide evidence for security incidents;
- **anti-virus programs**
  scan files for known viruses, check the integrity of files, or alert when suspicious actions are about to be executed;
- **cryptography**
  protects the confidentiality, the integrity and the authenticity of data and messages; an important aspect is the key management (see organizational measures, sec. 4.3);
- **firewalls**
  filter packets and shield the internal network against certain hostile actions; they can also impose a strict route inside and outside the internal network.
- ...

4.3 Organizational Measures

Most measures will fail if no strict procedures are developed:

- **account management**
  includes specific rules for the creation/deletion of accounts, rules for well-chosen passwords, ...
- **automatic backup**
  consists of a backup scheme, a restoration scheme, a number of safe vaults, etc.
- **auditing, monitoring**
  are important instruments in the detection of security breaches, and when applied properly can stop these incidents early.
- **education/training**
  has as main target keeping the users aware of the importance of the security measures and alert for symptoms of incidents, etc.

[3] Uninteruptable Power Supply.

- **an incident plan**
  involves a detailed (tested!) procedure, the appointment of a contact person, the elaboration of juridical steps, etc.
- **key management**
  determines how and when new keys are chosen, ...
- ...

5 Conclusions

Computer security is more than implementing a few measures. It should be derived from an explicitly stated security policy. No system can be secured for 100%. However, by selecting a good mixture of preventive, detective and corrective measures, the security incident rate can be drastically reduced.

Resources spent on securing the IT-infrastructure, should not be considered as 'unproductive overhead'. In fact a good security creates added value:

- it reduces the probability of fraud,
- it avoids hours of overtime,
- it increases the reliability of the services,
- it may be the only guarantee for the survival of the organization.

References

1. E. Amoroso. *Fundamentals of Computer Security Technology.* Prentice Hall Inc., 1994.
2. William R. Cheswick and Steven M. Bellovin. *Firewalls and Internet Security: Repelling the Wily Hacker.* Addison-Wesley professional computing series. Addison-Wesley, Reading, MA, USA, 1994.
3. D. Curry. Improving the security of your unix system. Technical Report ITSTD-721-FR-90-21, SRI International, Apr 1990.
4. Bart De Decker. Unix security & kerberos. In B. Preneel, R. Govaerts, and J. Vandewalle, editors, *Computer security and industrial cryptography: state of the art and evolution: ESAT course — May 1991, Leuven, Belgium*, number 741 in Lecture Notes in Computer Science, pages 257–274, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 1993. Springer-Verlag.
5. E. Felten, D. Balfans, D. Dean, , and D. Wallach. Web spoofing: An internet con game. Technical Report 560-96 (revised Feb. 1997, Dep. of Computer Science, Princeton University, 1996.
6. R. Focardi and R. Gorrieri. A classification of security properties. *Journal of Computer Security*, 3(1), 1995.
7. Simson Garfinkel and Gene Spafford. *Practical UNIX and Internet security.* Computer security (Sebastopol, Calif.). O'Reilly & Associates, Inc., 981 Chestnut Street, Newton, MA 02164, USA, second (completely rewritten and expanded to include Internet security) edition, 1996.
8. R. Hauser, P. Jansen, R. Molva, G. Tsudik, and E. van Herreweghen. Robust and Secure Password and Key Change Method. In Dieter Gollmann, editor, *Computer Security—ESORICS '94*, number 875 in Lecture Notes in Computer Science, pages 107–122. Springer, 1994.

9. J. Kohl and C. Neumann. The kerberos network authentication service. Technical Report RFC #4, MIT, dec 1990.
10. Peter E. Neumann. *Computer Related Risks*. Addison-Wesley, Reading MA, California, NY, etc., 1995.
11. Eugene H. Spafford. The internet worm program: An analysis. Technical Report CSD-TR-823, Purdue University, November 1989.
12. William Stallings. *Mecklermedia's official Internet world Internet security handbook*. IDG Books, San Mateo, CA, USA, 1995.