SECURITY IN A DEPENDABILITY PERSPECTIVE

Erland Jonsson
Tomas Olovsson
Department of Computer Engineering
Chalmers University of Technology
S-412 96 Göteborg
SWEDEN

email: jonsson/olovsson@ce.chalmers.se

ABSTRACT

Historically security has developed as a discipline, separate from the original dependability framework, which included reliability, availability and safety attributes. Therefore, the integration of security in this framework has not yet been fully accomplished. This paper presents a novel approach to security, intended to facilitate and improve this integration. This is achieved by taking a dependability viewpoint on traditional security and interpreting it in behavioural and preventive terms. A modified security concept, comprising only preventive characteristics is defined where confidentiality is suggested to be a behavioural dependability attribute. The outcome of this interpretation influences the integration of the other three dependability attributes. The overall objective of this approach is to arrive at a more general and clear-cut dependability framework, that would describe how (un)dependable a system is, irrespective of the reason for the (un)dependability. For example, it should be possible to treat a system failure due to an intentional intrusion or due to a hardware fault using the same methods and in parallel. Finally, the problem of interpreting concepts and terminology for security impairments in dependability terms is addressed, based on a few examples from real security breaches. It is realized that this is an area where future work is necessary.

Keywords: Security, Dependability, Vulnerability, Computers, Modelling, Terminology.

1 INTRODUCTION

The research field of security and dependability are two disciplines that describe important properties of computer systems. Security has emerged from the viewpoint of unauthorized interaction with a system, leading to disclosure or modification of information. Dependability has evolved from reliability and availability considerations. Security and dependability have traditionally been treated separately. Lately however, attempts have been made to integrate these two, e.g as suggested in [Laprie1992], where dependability is defined as an "umbrella" concept of which security is just an attribute among others. However, the consequences of this proposed integration have not yet been fully assessed. What we are facing here is the classical problem of two successful disciplines that are both evolving, resulting in a situation where an overlap occurs. Advocates for each discipline tend to incorporate the "other" into their "own" one without fully realizing the consequences that such an integration would entail. The incorporation of security as a dependability attribute has already been mentioned. Similar attempts can be found within the security community [Denning1989].

Another point of concern is that the concepts overlap and that each discipline uses a terminology that is often incompatible with that of the other discipline. The most striking example of overlap is availability. From the security viewpoint it describes the possible disruption of service delivery to the authorized user as a result of intentional interaction. However, from the reliability viewpoint the possible service disruption is normally due to a component failure, even if no restriction with respect to the cause is really made.

This paper attempts to improve this situation by taking a dependability viewpoint on traditional security and interpreting it in behavioural and preventive terms, with the intention to arrive at a unified dependability framework [Jonsson and Olovsson 1992]. We shall use the term dependability for the overall trustworthiness of the system, i.e. a concept that reflects the system's adherence to the specification of the system. Here, the *specification* is the comprehensive document that defines the behaviour of the system in a certain environment. For the purpose of this discussion we shall make the unrealistic assumption that the specification exists and is complete. The *behaviour* of the system is understood as the *service* delivered to its *user(s)*, whether authorized or unauthorized, and it is directly related to the *output* of the system. The environment also gives the input conditions under which the system is designed to operate. If the environment behaves as specified and the system operates correctly, the service delivered by the system will be correct and failure-free. The system is then dependable.

An illustration of a discrepancy in terminology is that in the dependability discipline reasons for *failures* are called *faults*, whereas security people talk about *attacks* that cause *breaches*. Whether these terms correspond directly is not clear, even if the similarity is evident. Many questions of this type could be posed. What are the relations between e.g. fault, attack, flaw, error, bug, vulnerability, defect? Do some of these terms represent identical concepts? Should we in that case look for unification of terminology, or is it justifiable to maintain separate terminologies for each discipline? These are questions which need to be answered as integration work proceeds, and even though a full answer will not be given in this paper, the suggestions made is hoped to facilitate further work in this direction.

In the following section 2 gives the traditional viewpoint of dependability and security and section 3 describes the terminology for dependability impairments. In section 4 a novel approach to integrate security and dependability is suggested. Section 5 briefly discusses the terminology for security impairments followed by a few examples in section 6.

2 TRADITIONAL DEPENDABILITY AND SECURITY CONCEPTS

Dependability was first introduced as an extension of *reliability* and *availability* and these were then reduced to be specific attributes of dependability together with *safety* and *security* [Laprie1985]. Reliability and availability constitute different views of a basic concept that deals with the delivery of *service*. Here, service is the system behaviour as perceived by its users [Laprie1992]. Reliability is a characteristic that reflects the probability that the system will deliver its service under specified conditions for a stated period of time, whereas availability reflects the probability that the system will be available, or ready for use, at a certain instant in time. Safety is also related to the service delivered by the system, but rather than characterizing the system during operation, it denotes the system's ability to fail in such a way that catastrophic consequences are avoided. Safety is reliability with respect to catastrophic failures.

Security was incorporated as the fourth dependability attribute. It refers to the system's ability to prevent unauthorized access or handling of information, and to its ability to withstand illegal interaction or attacks against *system assets* such as data, hardware or software. This notion of security normally assumes a hostile action from a person, the *attacker*, who often tries to gain some kind of personal benefit from his actions. Security is normally defined by three different aspects: *confidentiality, integrity* and *availability* [Hsaio1988], [ITSEC1991], [Laprie1992], [Laprie1992b], [Pfleeger1989].

Confidentiality, which is also called *secrecy*, is the ability of the computing system to prevent disclosure of information to unauthorized parties. *Integrity*, sometimes called *accuracy*, is the ability of the computer system to prevent data or other assets from being modified, deleted or destroyed by an unauthorized party. Finally, *availability*, is the system's ability to deliver its normal service to the authorized user, even in the presence of attacks. It should be noted that availability is found in two places in the present dependability model, both as a direct attribute to dependability and as one of the security aspects.

Various versions of the definition of security exist. See e.g. [ISO7498], [Garfinkel and Spafford1991], [Muftic1989]. In database systems *integrity* refers to actions taken by an authorized party and to the accuracy and validity of data, whereas *security* refers to protection of data against unauthorized interaction [Date1990].

3 TRADITIONAL DEPENDABILITY IMPAIRMENTS

3.1 Background

The discussion in this paragraph is primarily based on the ideas presented in [Jonsson 1993]. A model of a dependable system contains at least two components. The first is the *object system* (or *system* only) for dependability assessment, in the following denoted **SYS**. The second component is the environment consisting of a number of environmental subsystems.

Consider a specific system as depicted in the block diagram in figure 1 in which a circle denotes a state, an arrow an event and a square a (sub)system. The block diagram describes the dependability impairments we are going to discuss. The environment interacts with the SYS by generating inputs to it and by receiving outputs from it.

The discussion in this section starts out from the observation that a failure is normally preceded by a chain of events that leads to that failure. These events and their intermediate effects on the system are called *impairments*.

3.2 Threat

In theory, all subsystems in the environment may interact with the SYS. This interaction may be intentional in the sense that the subsystem is functionally connected to the SYS. The interaction may also be unintentional, reflecting no functional relationship. The interaction, whether intentional or unintentional, may result in undesired effects in the SYS. From this viewpoint the environmental subsystem represents a *threat* to the dependability of the SYS, thus the definition:

Definition:
A dependability **threat** is an environmental subsystem, that can possibly lead to an error or a vulnerability in the system.

The notion of threat has normally been linked to intentional faults and the security attribute. The above definition is much wider. Any subsystem in the environment may constitute a dependability threat.

3.3 Fault

A fault that can be related to a threat is called an **external fault**, since the source of the fault is found outside the SYS. **Internal faults** are faults that arise (apparently) spontaneously somewhere in the system, i.e. with no detectable relation to a threat. The following definition of fault covers both cases:

Definition:
A **fault** is an event leading to an error in the system.

A fault is an *event* or *system state change* and is regarded as an atomic phenomenon. Thus, a fault is never permanent but an inherently transient phenomenon. Neither is a fault intermittent. An “intermittent fault” is regarded as a number of repeated transient incidents. The fault is the direct reason for the error occurrence in the system and will unavoidably lead to an error.

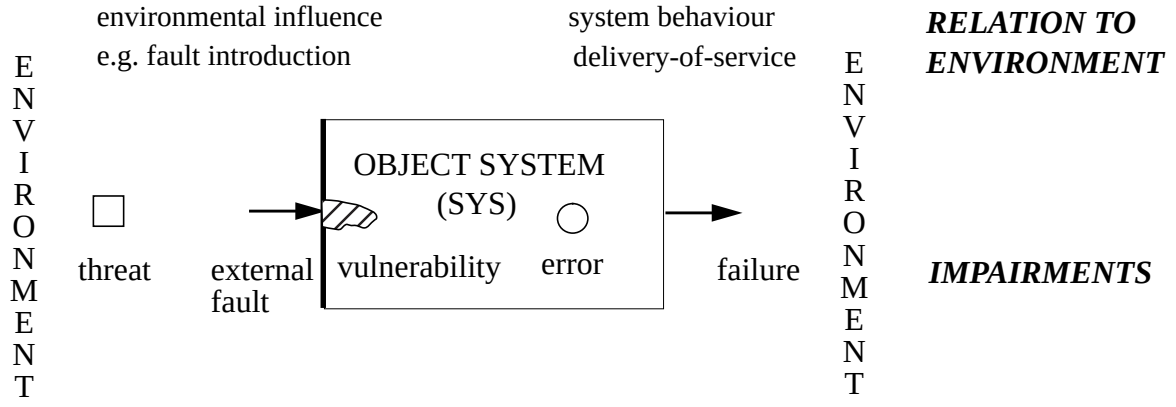

Figure 1. Dependability impairments

3.4 Error

Definition:
An **error** is a system state that may lead to a system failure during normal system operation.

Since this definition is very general and intended to be applicable to many different types of systems, the word *state* has to be understood in a broad sense. Thus we will avoid giving an exact definition or interpretation of the word that would be valid for all cases. Once an error has occurred in the system, the system is *erroneous*. The error may propagate to the system boundary, and lead to a failure.

3.5 Failure and failed state

Definition:
A **failure** is the *event* when a deviation first occurs between the service delivered by the system and the expected service, as defined in the system specification.

A failure or failure transition is an event that represents a state-change of the total system with respect to the service it delivers. Before the failure occurs, the service delivered by the system is correct and in accordance with the specification. After the failure, the service deviates from what is specified. Thus the following definition applies:

Definition:
A system that exhibits a deviation between the delivered service and the specified service, is said to be in a **failed state**.

A fault will lead to an error by definition. The resulting error may or may not start propagating depending on the operational circumstances, *error propagation*. Therefore, an error may not necessarily lead to a failure, and even if it does, the failure may manifest itself only after a considerable delay. The fundamental observation here is that *it is not until a failure has occurred that any harm is done* as seen from the user. As a consequence, a fault or an error *will not affect the dependability* of the system *if it never leads to a failure*. Thus a system may be subjected to faults and contain errors and still be completely dependable.

It should be noted that the error propagation model is not always applicable. This is especially so in some *collapsed* cases, where the failure emerges virtually directly, with no significant delay from the fault event. It may even be hard to define or distinguish the corresponding fault and error(s). Typical examples are failures that are the result of violent action towards hardware, e.g. crashing the screen, but also many of the confidentiality failures, such as the overhearing of a message, whether acoustic, visual or electronic.

3.6 Vulnerability

The critical points for a dependable system are the places where faults are injected, which for external faults is at the boundary between the environment and the SYS. The environment contains the dependability threats, whose behaviour represent a risk for fault injection into the SYS. This risk can never be completely eliminated, and there will always be a remaining probability for external fault injection into the system. This is the reason why it is worthwhile to improve the system in such a way that it can better withstand the threats. This is the same thing as to reduce the probability that a dependability threat creates a fault. Here we need to define the term vulnerability:

Definition:
A **vulnerability** is a place where the probability for fault injection exceeds a predefined threshold.

The vulnerability (deficiency, weakness, flaw) concept is well-known from the security domain. A security attack may be aimed at planting a vulnerability in the system, a vulnerability that can later be exploited by further successful attacks to cause loss or harm. The term vulnerability is also applicable for non-intentional interaction. For example, a hardware vulnerability could typically be an unshielded cable, which is inclined to pick up external noise.

The significant difference between a vulnerability and an error is that an error will propagate under normal operating conditions when the erroneous part of the SYS is activated (used). Vulnerabilities will not propagate during normal operation but only function as a channel for external faults into the system.

4 INTEGRATING SECURITY AND DEPENDABILITY ATTRIBUTES

4.1 Background

We now ask ourselves how the traditional security concept could be readily integrated with reliability/availability. The three security aspects, confidentiality, integrity and availability are, according to [ITSEC1991], given a *preventive* definition: the *prevention of unauthorized disclosure, modification and withholding* respectively, i.e. related to the prevention of faults from being introduced into the system. We shall see in the following that by means of giving them a *behavioural* meaning, i.e. related to the behaviour of the system, or *preventive* meaning they are, to a large extent, already covered by existing concepts in the dependability discipline.

A closer look closer at the system behaviour shows that we need to distinguish between two different receivers of the output delivered by the system: the authorized user and the unauthorized user. See figure 2. The authorized users are the users that are the intended receivers of the service that the system delivers, as specified in the system specification. In the following we shall call the authorized user(s) the **User**. A user is any system in the environment that is a potential consumer of the output delivered by the system. It may be human or object: a person, a computer, a program etc. All potential users except the authorized users are unauthorized users. Unauthorized users, or authorized users who are abusing their authority are called **Non-users.** For a more detailed discussion please refer to [Jonsson and Olovsson 1992].

4.2 Availability

In traditional security availability is defined as the unauthorized withholding of information and resources. The traditional dependability interpretation is: readiness for usage, i.e., the ability of the system to deliver its service to the User. Therefore, availability could in both cases be regarded as a behavioural concept, denoting the extent to which system service is available to the User. See figure 2.

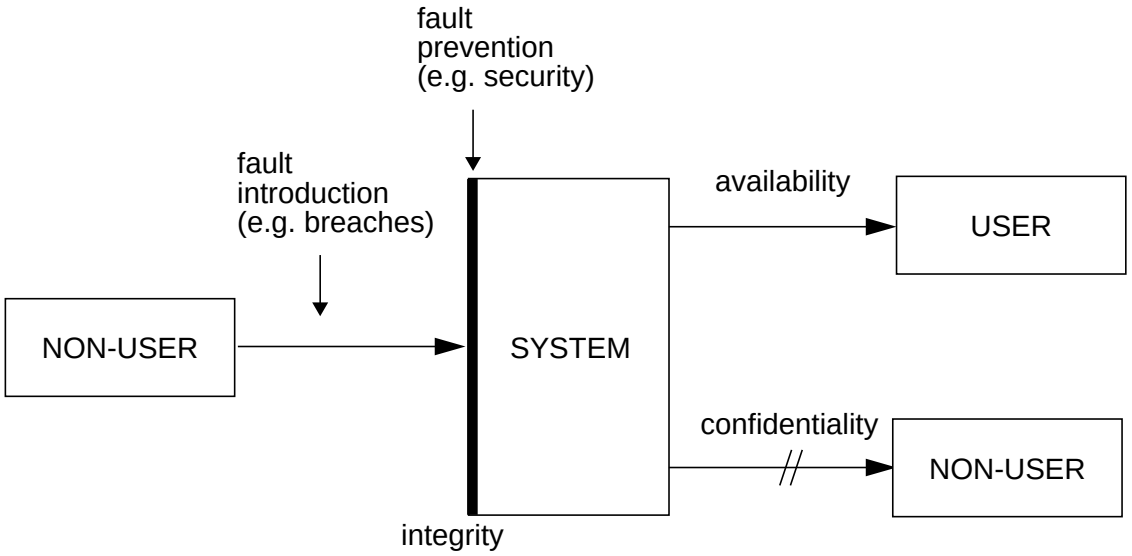

Figure 2: Understanding security in dependability-related terms

4.3 Integrity

Integrity is the prevention of unauthorized modification, deletion or destruction of system assets. Integrity is violated by means of an attack, performed by a Non-user. Thus, integrity is a preventive quality of a system and characterizes the system's ability to withstand certain type of attacks. Therefore, integrity is one specific type of dependability fault prevention.

4.4 Confidentiality

Confidentiality is the ability of the system to prevent unauthorized disclosure of information (for IT systems). It also includes unauthorized use of system assets and resources. Thus, confidentiality means restricting the availability of the service delivered by the system to the Non-users. It is thus a behavioural concept which defines certain characteristics of the system behaviour, but unlike availability attribute it defines *system behaviour with respect to a Non-user.* It actually defines to what extent information and other assets should be accessible, or rather not accessible, to Non-users. With this interpretation confidentiality can be regarded as a new behavioural attribute in the dependability discipline.

4.5 Security

In view of the discussion in the previous sections we suggest a modified definition of the security concept, so that security is simply regarded as prevention with respect to the introduction of intentional faults or intentional vulnerabilities. Thus, security is a purely preventive concept, which is not at all related to the behaviour of the system but only to its ability to protect itself against certain types of faults and attacks. This is a rather restricted interpretation of security and one could discuss whether there would be some more appropriate word to use for this concept. We will, however, use the term security in this paper. Consequently, *security mechanisms* are fault prevention mechanisms.

Recently, there has been some attempts to find quantitative measures for security in this fault preventive interpretation. See [Olovsson and Jonsson1994], [Brocklehurst et al 1994] and references therein. In these, it is suggested that the *effort* expended by an attacker to make a security breach could be used as a measure of the security of the system. The basic idea is that the more effort an attacker has to use to accomplish a security breach, in a statistical sense, the more secure the system is. Thus, the effort parameter would reflect the system's degree of security, i.e., its fault prevention ability with respect to intentional faults.

4.6 Modified dependability attributes

The discussion in the preceding subsections suggested that confidentiality should be treated as a separate dependability attribute that would describe the relation of the system to the Non-user. Reliability and availability, on the other hand, are both attributes describing the relation of the system to the User. They could therefore be regarded as views on the same composite attribute: reliability/availability.

Where does that leave safety? Following the proposed terminology safety must be regarded as a "sub-attribute" to either one of reliability/availability or confidentiality. Therefore, dependability can be understood in terms of only two attributes: reliability/availability (related to the User) and confidentiality (related to the Non-user), leaving safety to describe certain types of failures, i.e. catastrophic failures, for both of these. See figure 3.

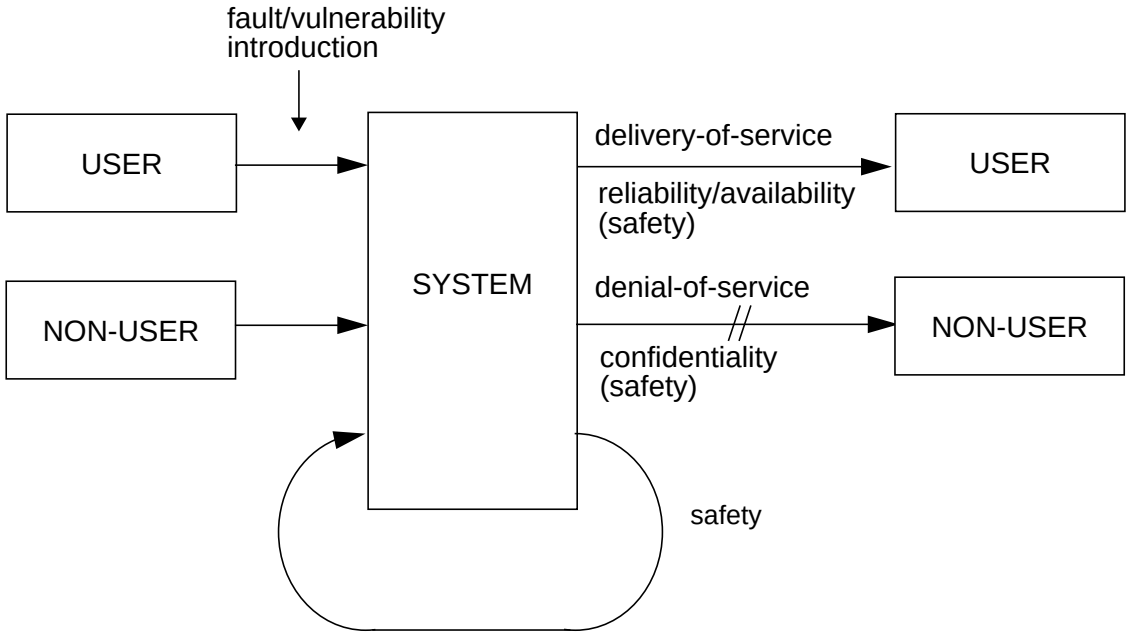

Figure 3: Modified dependability attributes

The modified definitions of dependability and its attributes can be summarized as follows:

reliability/availability: refers to the system's ability of delivery-of-service to the authorized users, called Users.

confidentiality: refers to the system's ability of denial-of-service to unauthorized users, called Non-users. All users but those explicitly specified as authorized users are Non-users.

safety: refers to the system's ability to avoid unintended catastrophic consequences, whether due reliability failures or to confidentiality failures. These consequences may affect the environment, including Users and Non-users or the system itself.

dependability: is the trustworthiness of a computer system such that reliance can be justifiably placed on the service it delivers to its Users, on the confidentiality it maintains with respect to its Non-users and on the absence of unintended catastrophic consequences.

In the definition of safety, the word "unintended" is included since many systems are intentionally constructed to cause catastrophic consequences on the environment, an obvious example being warheads.

5 SECURITY IMPAIRMENTS

In view of the behavioural system model discussed in section 4 and the suggested integration of security and dependability attributes, we shall in this section suggest definitions and/or interpretations of security impairments from a dependability viewpoint. As we shall see, system impairments due to security violations can in many cases be directly interpreted in dependability terms, e.g as fault - error - failure and vulnerability. However, in some cases this interpretation is not evident, and it is quite clear that there are some problems in this context that need to be further addressed. It shall also be pointed out that the definition of system boundaries can pose some problems, especially when defining the boundaries between the attacker and the user.

Threat

The security threat concept agrees with that of dependability. It is a subsystem in the environment that interacts with the object system that can possibly create a fault or create a vulnerability in the system.

Attack

An attack is an attempt to perform a breach in the system.

Breach

A breach is an event that creates an error in the system, i.e., an intentional fault, or an event that creates a vulnerability in the system. It is the result of a successful attack. A breach normally represents a violation of the (system) security policy.

Error

An error can be the result of a breach or of a “normal” fault, i.e., a fault that is not due to some intentional human interaction. In the first case we refer to the error as a “security” error and in the latter as a “reliability” error.

Failure

A failure that occurs is the result of a “security” error or “reliability” error that has propagated to the system boundary. We refer to these failure as a “security” failures or as “reliability” failures.

6 EXAMPLES

6.1 Background

The previous discussion was aimed to show the difference between the behavioural and preventive view of the system and its attributes and to distinguish between faults and failures. In particular it was suggested that security is purely preventive, i.e., related to faults and vulnerabilities and that the behavioural attributes of dependability are reliability/availability and confidentiality. In this section we shall give a few examples of some security attacks and discuss those and their consequences from a dependability viewpoint. We also show how security faults can lead to reliability as well as confidentiality failures. Some of the examples are taken from a security intrusion experiment that was performed at the department of Computer Engineering at Chalmers University of Technology [Olovsson and Jonsson1994].

6.2 Security fault leads to reliability failure

The “kbd_mode” command is intended to reset the keyboard of a Sun UNIX-system to a well-defined state. However, it has turned out that it is possible to execute the command remotely on another machine, in which case the keyboard of that machine gets locked, i.e., it becomes unavailable to the User.

The fact that the execution of the command leads to something else than intended, is an error in the software, since it could be expected that executing a command remotely should lead to the same result as executing it locally, or, if not leading to the same result, should be disregarded by the system. Therefore, the programmer has made a fault in the design process that lead to this error. The error is activated by the attacker, who makes it propagate to cause a reliability failure, so that the User can no longer use his machine.

It is also quite clear that a pure hardware (component) fault, which is not a security issue, could lead to very same result, i.e. a disabling of the User keyboard.

6.3 Reliability fault leads to confidentiality failure

It is quite obvious that “normal” hardware or software faults can lead to confidentiality failures, e.g., an erroneous authorization program could lead to leakage of information to a Non-user, or a pure hardware fault entail that an intended encryption of a message did not occur.

6.4 Trojan Horse

A User has left his login file world readable and writable. This vulnerability can be exploited by an attacker to plant a Trojan Horse. The Trojan Horse is activated for a certain set of conditions, that could e.g. involve time, certain actions by the User etc. The planted Trojan Horse is an error in the system, since it will be activated during normal operation. When the conditions are fulfilled the Trojan Horse will be activated and perform its task. This task may be deleting all the User’s files on the hard disk, which is a *reliability* failure.

Another example would be if the Trojan Horse was activated when the User sent email to a certain receiver. The action in this case could be copying the email to the Non-user, which is a *confidentiality* failure.

7 CONCLUSIONS

A novel approach to the integration of security and dependability has been proposed. It is based on the observation that the dependability of a computer system could be described in behavioural and preventive terms. A behavioural viewpoint is related to the behaviour of the system, i.e. to how the system influences its environment. A preventive viewpoint describes the measures to be taken to prevent faults from being introduced into the system, i.e. how to prevent unwanted environmental influence on the system.

Using this approach, we have shown how the various aspects of traditional security could either be mapped onto existing dependability concepts or be understood as a new dependability attribute, which we call confidentiality. Confidentiality is different from the traditional dependability attributes in that it describes the system’s relation to an unauthorized user and not to the authorized user. Safety describes the system’s ability to avoid catastrophic failures whether reliability or confidentiality failures. Security is redefined as a concept for fault prevention with respect to intentional external faults or attacks against the system. In this way (preventive) security has no *direct* relation to behavioural attributes confidentiality and reliability/availability.

8 REFERENCES

[Anderson1989] **T. Anderson** (editor)**:** *Safe & Secure Computing Systems*, Blackwell Scientific Publications, ISBN 0-632-01819-4, 1989.

[Baker1991] **R. H. Baker:** *Computer Security Handbook, 2nd Edition,* TAB Professional and Reference Books, McGraw-Hill Inc, ISBN 0-8306-7592-2, 1991.

[Brocklehurst et al 1994] **S. Brocklehurst, B. Littlewood, T. Olovsson, E. Jonsson:** "On measurement of Operational Security", pp. 257-266 in Proceeding of the Ninth Annual IEEE Conference on Computer Assurance, COMPASS'94, Gaithersburg, Maryland, USA, June 29 -July 1, 1994.

[Date1990] **C. J. Date:** *An Introduction to Database Systems, vol. 1, 5th edition,* pp. 429ff. Addison-Wesley 1990, ISBN 0-201-51381-1.

[Denning1989] **D.E. Denning:** "Secure Databases and Safety: Some unexpected conflicts," pp. 101-111 in T. Anderson (editor): *Safe & Secure Computing Systems*, Blackwell Scientific Publications, ISBN 0-632-01819-4, 1989.

[DOD1983] **Department of Defence:** *Trusted Computer System Evaluation Criteria* ("orange book"), CSC-STD-001-83.

[Garfinkel and Spafford1991] **S.Garfinkel, G.Spafford:** "Practical UNIX Security", p. 11ff, O'Reilly and Associates Inc., ISBN 0-937175-72-2, 1991.

[Hsaio1988] **D. K. Hsiao:** "Database Security Course Module," pp. 269-301 in *Database Security: Status and Prospects,* Elsevier Science Publishers B.V, Holland, IFIP WG 11.3, ISBN 0-444- 70479-5, 1988

[ITSEC1991] **Information Technology Security Evaluation Criteria (ITSEC)**: *Provisional Harmonized Criteria, December 1993*. ISBN 92-826-7024-4.

[ITSEM1993] **Information Technology Security Evaluation Manual (ITSEM)**: *Provisional Harmonized Methodology, September 1993.* ISBN 92-826-7087-2.

[ISO1983] **International Standards Organization:** *Data Processing - Open Systems Interconnection - Basic Reference Model,* ISO/IS 7498, Geneva 1983.

[ISO7498] **International Standards Organization:** *Information processing systems - Open Systems Interconnection - Basic Reference Model, part 2: Security Architecture 7498/2*.

[Jonsson and Olovsson 1992] **E. Jonsson, T. Olovsson,** "On the Integration of Security and Dependability in Computer Systems", IASTED International Conference on Reliability, Quality Control and Risk Assessment, Washington, Nov. 4-6, 1992. ISBN 0-88986-171-4, pp. 93-97

[Jonsson 1993] **E. Jonsson,** "A Unified Approach to Dependability Impairments in Computer Systems", IASTED International Conference on Reliability, Quality Control and Risk Assessment, Cambridge, MA, Oct. 18-20 1993, ISBN 0-88986-181-1, pp. 173-178.

[Joseph1991] **M.K. Joseph:** "Integration problems in Fault-tolerant, secure computer design," pp. 347-364 in A.Avizienis. J.C. Laprie (editors): *Dependable Computing for Critical Applications,* Springer-Verlag, N.Y., ISBN 3-211-82249-6, 1991.

[Laprie1985] **J.C. Laprie**: "Dependable Computing and Fault Tolerance: Concepts and Terminology," in *Proc. 15th IEEE International Symposium on Fault-Tolerant Computing (FTCS-15), June 1985.*

[Laprie1992] **J.C. Laprie et al.:** *Dependability: Basic Concepts and Terminology*, Springer-Verlag, ISBN 3-211-82296-8, 1992.

[Laprie1992b] **J.C. Laprie:** Dependability: a unifying concept for reliable, safe secure computing, Proc. of 12th IFIP World Computer Congress Madrid, Spain, September 1992, vol. 1, pp. 585-593.

[Louw1992] **E. Louw, N. Duffy**, "Managing computer viruses", Oxford University Press, ISBN 0-19-853973-8, 1992.

[Muftic1989] **S. Muftic:** *Security Mechanisms for Computer Networks,* Ellis Horwood Ltd, England, ISBN 0-7458-0613-9, 1989.

[Nessett1986] **D. M. Nessett:** "Factors Affecting Distributed System Security," *IEEE Symp. on Security & Privacy*, 1986, pp. 204 - 222.

[Olovsson and Jonsson1994] **T. Olovsson, E. Jonsson, S. Brocklehurst, B. Littlewood:** Investigation of Quantitative Assessment of Operational Security Based on Intrusion Experiments, Technical Report No 192, Department of Computer Engineering, Chalmers and Compendium from the Nordic Seminar on Dependable Computing Systems 1994 (NSDCS'94), Lyngby, Denmark, Aug. 24-26, 1994.

[Pfleeger1989] **C. P. Pfleeger:** *Security In Computing*, Prentice Hall International, Inc. ISBN 0-13-799016-2, 1989.