1

Issues of attack in distributed systems - A Generic Attack Model

Kantzavelou I. and Patel A.
Computer Networks and Distributed Systems Research Group
Department of Computer Science, University College Dublin
Belfield, Dublin 4, Ireland, tel. +353-1-7062476, fax +353-1-2697262, ioanna@teia.ariadne-t.gr, apatel@ccvax.ucd.ie

Abstract

In the past, IT equipment consisted solely of standalone systems, whereas in recent years the trend has been towards computer networks and distributed systems. The spread of distributed information technology has increased the number of opportunities for crime and fraud in computer systems. Despite the fact that computer systems are typically protected by a number of security mechanisms (Muftic 1989) such as encryption (Denning 1983), digital signature (ISO 7498-2 1989), access control (Muftic 1993), and passwords (Pfleeger 1989), attacks continue to occur (Highland 1993). In addition, it seems infeasible to close all the known security loopholes of today's systems. Therefore, computer systems and especially distributed systems continue to envisage a number of threats. A threat is a potential violation of security (ISO 7498-2 1989). More specifically, a threat is a possibility of an attack, and an attack is an attempt (by an attacker) to damage or in some way negatively affect the working of a computer system, or to damage the interest of the organisation owning the system (Kantzavelou 1994). This paper discusses issues of attack and the construction of a generic attack model.

Keywords

Threat, attack, attacker, distributed system, security flaw, method of attack, attack diagnosis factors.

R. Posch (ed.), *Communications and Multimedia Security*

1 INTRODUCTION

The growing spread of computer networks and distributed systems has created a number of threats to the security of these systems. The main source of these threats is users who use methods of attack to damage a system (Kantzavelou 1995).

A threat is a potential activity with expected or unexpected harmful results. The problems caused by this activity may or may not be resolved. The source of a threat might be one of three factors: *physical*, *human*, and *technical*, as they are described below (Kantzavelou 1994).

Physical factor

The physical factor includes natural disasters such as fire, storm and water damage.

Human factor

The human factor is the main source of computer breaches and includes unauthorised users who wish to damage a system and authorised users of a system who misuse the system either deliberately or accidentally.

Technical factor

The technical factor is the equipment of a system which might fail to carry out its functions (equipment failure) or it might carry them out in an inappropriate way (equipment malfunction).

In general terms, the target of a threat is the computer system. In particular, the assets of a computer system, i.e. the hardware, the software, the data, etc., are subject to threats. The following list of types of threats describes the results to the above assets that might become apparent when threats have been realised (ISO 7498-2 1989, Pfleeger 1989, ECMA TR/46 1988).

Disclosure of Information

Computer networks store, process and convey large amounts of information, some of it very valuable to organisations. Disclosure of such information may cause severe problems which harm the overall activity of an organisation.

Corruption of Information

A user who has succeeded in reading unauthorised information on a computer system, may wish to alter it for his own purposes. Corrupted information may be less valuable or completely worthless. The degree of damage may be higher in this case than in the case of disclosure only.

Unauthorised Use of Resources

Unauthorised use of the resources (CPU, disk, I/O devices, etc.) of a computer system may lead to destruction, alteration or loss of integrity of the resources, and lack of availability of the resources for authorised activities.

Misuse of Resources

The intentional or accidental misuse of the system resources by authorised users may lead to corruption, destruction, disclosure, or loss of data or resources.

Unauthorised Information Flow

The major function of a computer network is the transmission of information through the network. Transmission of information must be limited to allow information flow only between authorised users and end-systems. The unauthorised flow of information is a serious threat.

Denial of Service

Denial of service includes the failure of a system to carry out one or more of its functions. The threat of denial of service in the computer network of an organisation, which is dependant to a great degree on IT for its operations, is potentially catastrophic. For this reason, this threat must be considered thoroughly as part of any security policy.

Apart from this list of threats there is also another threat, the *Repudiation of Information Flow*. The repudiation of information flow involves denial of transmission or receipt of messages. Although this is a considerable threat in a networked environment which conveys valuable information, it does no actually endanger a computer system. Repudiation is a threat by one user against another, not a threat to the system as a whole, and thus why it is not included in the above list.

When a threat has been realised an *attack* is said to take place. Attacks are categorised into *accidental* and *intentional* attacks according to the attacker's intentions and into *passive* and *active* attacks according to their effects on the system. Descriptions of these attack categories follow (ISO 7498-2 1989).

Accidental Attacks

Accidental attacks are those that occur with no premeditated intent. Such attacks occur as a result of system malfunctions, operational blunders, software bugs, and user mistakes.

Intentional Attacks

Intentional attacks are those that occur with premeditated intent, and may range from casual data and system examination using easily available monitoring tools to sophisticated attacks using special system knowledge.

Passive Attacks

Passive attacks refer to unauthorised disclosure of information without modification. For example, the use of passive wiretapping (Christmas 1992) to observe information being transmitted over a communication line is a passive attack.

Active Attacks

Active attacks include the alteration of information contained in a system and changes to the state or the operation of a system. For example, a malicious modification to a file by an unauthorised user is an active attack.

Given that accidental attacks does not include any premeditated intent, the "attacker" in such a case attacks the computer system while he is using it.

In addition, the effects of a passive attack usually are not observable by the users of the computer system. For example, the contents of a file might be disclosed by an attacker to an unauthorised person and this might pass unobserved.

2 A GENERIC ATTACK MODEL

The main concept of the generic attack model discussed in this paper is based on the fact that the *Target System (TS) contains* a number of *flaws* (Gritzalis 1995). These flaws might be *exploited* by an *attacker* who *uses* a *method of attack* to *attack* the target system. In addition, an attacker might introduce additional flaws in order to achieve his own purposes. Figure 1 depicts the described generic model of an attack that can be mounted over a TS.

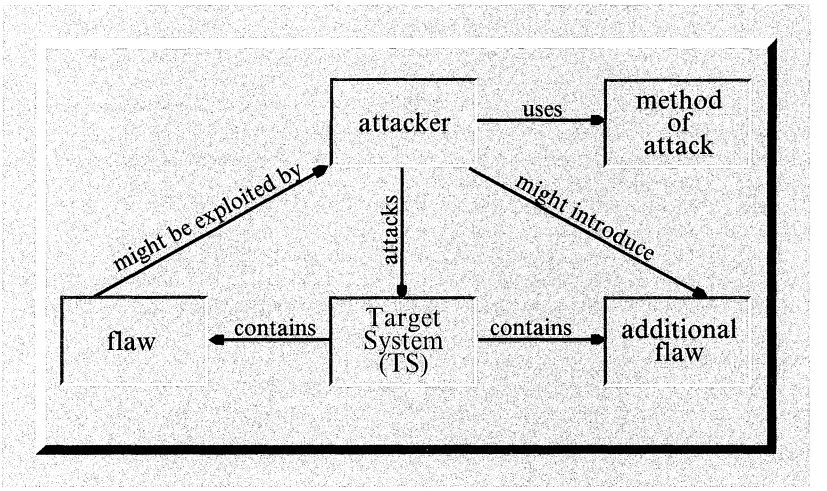

Figure 1 A generic attack model

The following sub-sections describe in detail all the entities involved in this generic attack model.

2.1 Attackers

An *attacker* might fall into one of the following three categories (Kantzavelou 1994, Anderson 1980):

- **external penetrators** are unauthorised users who wish to damage a system, or the interest of the organisation owning the system
- **internal penetrators** are authorised users of a system who are not authorised for the use of resources accessed. This category also includes masquerades who operate under another user's identity, and clandestine users who evade auditing and access controls
- **misfeasors** are authorised users of a computer system and of the resources they access, but who misuse their privileges.

2.2 Target System (TS)

The *Target System* (TS) of an organisation may vary widely. In general terms, a TS might be a standalone or networked portable/laptop, a standalone or networked PC, a Local Area Network (LAN) or a Wide Area Network (WAN), or a minicomputer/mainframe (Patel 1994).

A set of distributed machines connected over a Local Area Network (LAN) is the *Target System* of the *generic attack model* discussed in this section. The use of distributed machines connected over a LAN was decided because of three facts: first, that such a system was available for our research work; second, that there are a number of reasons why LANs need to be more secure (Muftic 1993); and third, that LANs extend to Wide Area Networks (WANs) (Harshall 1992).

This described Target System has been analysed and the following processes have been identified as the functions that the TS carries out:

1. Manage the system
   This process includes mainly the superuser's activities and all the security sensitive user activities. It might be divided into the following nine sub-processes:
   1.1. log in
   1.2. log out
   1.3. invoke other user's privileges
   1.4. invoke remote shell
   1.5. change password
   1.6. change date
   1.7. change host name
   1.8. consume a resource
   1.9. disclose system information
2. Manage a file
   This process includes all the user activities that deal with files. It might be divided into the following four sub-processes:
   2.1. disclose information of a file
   2.2. modify a file
   2.3. remove a file

2.4. pack - Unpack a file

3. Manage a directory
   This process includes all the user activities that deal with directories. It might be divided into the following four sub-processes:
   3.1. disclose information of a directory
   3.2. modify a directory
   3.3. remove a directory
   3.4. change the working directory
4. Manage an e-mail message
   This process includes all the user activities that deal with e-mail messages. It might be divided into the following four sub-processes:
   4.1. send an e-mail message
   4.2. receive an e-mail message
   4.3. accept an e-mail message
   4.4. deny an e-mail message
5. Manage a job
   This process includes all the user activities that deal with jobs. It might be divided into the following four sub-processes:
   5.1. display the job queue
   5.2. remove a job from the queue
   5.3. run a job
   5.4. stop a job

Figure 2 presents the Target System main processes on a top level (zero level) data flow diagram (DFD).

The investigation of the Target System processes in lower levels made clear its weaknesses and security loopholes that can be exploited by an attacker. Analysis and classification of security flaws that are contained or might be mounted on the Target System are described in the next subsections.

2.3 Security flaws

A *flaw* is a condition or a circumstance that can result in denial of service, disclosure, destruction or modification of information (Landwehr 1981). Landwehr et al (Landwehr 1993) have proposed a taxonomy scheme which can be used to classify a security flaw, distinguishing the nature of a flaw from the nature of its exploitation. Following this taxonomy scheme, a security flaw is classified according to how, when, and where it was introduced into an automated information system, i.e., according to *genesis*, *time of introduction*, and *location*. Short descriptions of these main flaw classifications, as well as of their divisions and subdivisions follow (Gritzalis 1995)

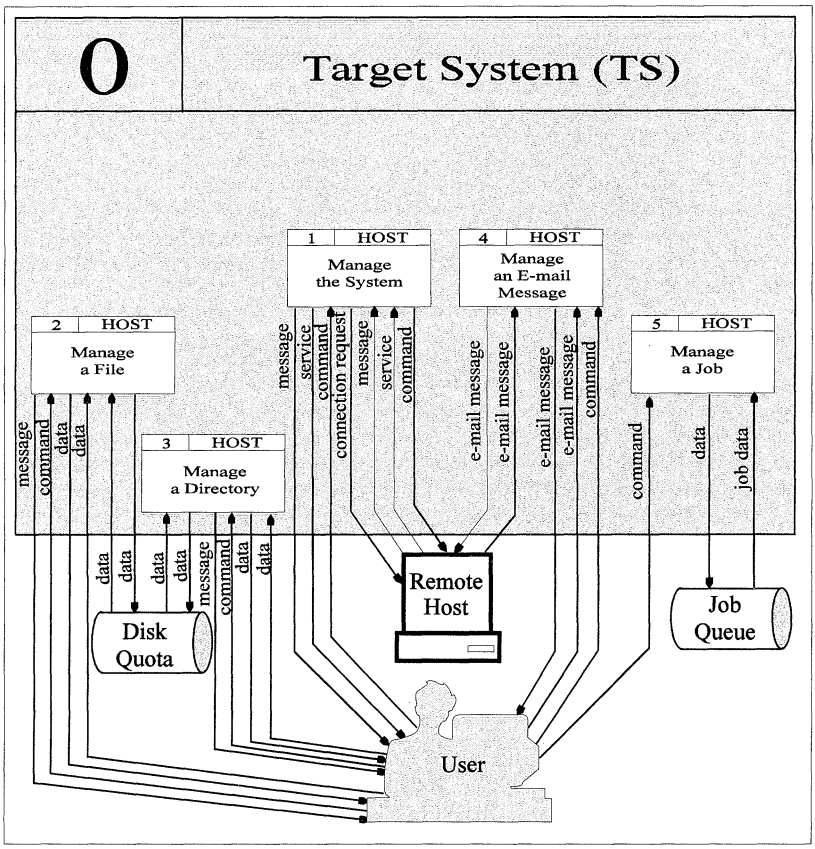

Figure 2 Target System (TS) Data Flow Diagram (DFD)

Security flaws classified by genesis

Flaw classification by genesis aims to provide the method a security flaw finds its way into a program.

- Intentional
  Flaws that have been intentionally introduced into a program.
  - Malicious
    Malicious flaws include Trojan horse, Trapdoor, and logic/time bomb.

- Trojan Horse
  A Trojan generally refers to a program that masquerades as a useful service but exploits rights of the program's user in a way the user does not intend.
  - Non-Replicating
    A Trojan horse that exploits rights of the program's user but does not replicate itself into the program.
  - Replicating (virus)
    A Trojan horse that replicates itself by copying its code into other program files is commonly referred to as a *virus*. One that replicates itself by creating new processes or files to contain its code, instead of modifying existing storage entities, is often called a *worm*.
- Trapdoor
  A trapdoor is a hidden piece of code that responds to a special unit, allowing its user access to resources without passing through the normal security enforcement mechanism.
- Logic/Time Bomb
  A logic-bomb or a time-bomb is a piece of code that remains dormant in the host system until a certain "detonation" time or event occurs. When triggered, a time-bomb may deny service by crashing the system, deleting files, or degrading system response-time. A time-bomb might be placed within either a replicating or a non-replicating Trojan horse.

- Non-Malicious
  Non-malicious intentional flaws include storage and timing covert channels as well as other kinds of such security flaws.
  - Covert Channel
    A covert channel is simply a path used to transfer information in a way not intended by the system's designers. Covert channels are frequently classified as either *storage* or *timing* channels. The distinction between storage and timing channels is not sharp. In practice, covert channels are often distinguished on the basis of how they can be detected: those detectable by information flow analysis of specifications or code are considered storage channels. Descriptions of these two types of channels follow:
    - Storage
      A storage channel transfers information through the setting of bits by one program and the reading of those bits by another. What distinguishes this case from that of ordinary operation is that the bits are used to convey encoded information. Examples would include using a file intended to hold only audit information to convey user passwords - using the name of a file or perhaps status bits associated with it that can be read by all users to signal the of the file.
    - Timing
      Timing channels convey information by modulating some aspect of system behaviour over time, so that the program receiving the information can observe system behaviour (e.g. the system's paging rate, the time a certain transaction requires to execute, the time it takes to gain access to a shared bus) and infer protected information.

  - Other
    Functional requirements that are written without regard to security requirements can lead to such flaws.
- Inadvertent
  Flaws that have been accidentally introduced into a program. They may occur in requirements; they may also find their way into software during specification and coding.
  - Validation Error
    Validation flaws occur when a program fails to check that the parameters supplied or returned to it conform to its assumptions about them. These assumptions may include the number of parameters provided, the type of each, the location or maximum length of a buffer, or the access permissions on a file.
  - Domain Error
    Domain flaws occur when the intended boundaries between protection environments have holes. For example, a user who creates a new file and discovers that it contains information from a file deleted by a different user has discovered a domain flaw.
  - Serialisation/aliasing
    A serialisation flaw permits the asynchronous behaviour of different system components to be exploited to cause a security violation.
  - Identification/Authentication Inadequate
    An identification/authentication flaw is one that permits a protected operation to be invoked without sufficiently checking the identity and authority of the invoking agent. These flaws could perhaps be counted as validation flaws, since presumably some routine is failing to validate authorisation properly.
  - Boundary Condition Violation
    Boundary condition flaws typically reflect omission of checks to assure constraints (e.g. on table size, file allocation, or other resource consumption) are not exceeded. These flaws may lead to system crashes or degraded service, or they may cause unpredictable behaviour.
  - Other Exploitable Logic Error
    Bugs that can be invoked by users to cause system crashes, but that don't involve boundary conditions.

Security flaws classified by time of introduction

Classifying identified security flaws, according to the phase of the system life cycle in which they were introduced.

- During Development
  Flaws introduced during development of the software can originate in:
  - Requirement/Specification Design
    Software requirements describe what a particular program or system of programs must do. How the program or system is organised to meet those requirements is typically recorded in a variety of documents, referred to collectively as specifications. This category includes flaws introduced during the requirements and specification design phase.

  - Source Code
    The source code implements the design of the software system given by the specifications. This category includes flaws introduced during the programming phase.
  - Object Code
    Object code programs are generated by compilers or assemblers and represent the machine-readable form of the source code. This category includes flaws into the compilers or assemblers that cause problems to the object code programs.
- During Maintenance
  Flaws introduced during maintenance of a system mainly due to programmer's failure to understand the system as a whole.
- During Operation
  Flaws introduced during the operation use of a system.

Security flaws classified by location

A security flaw can be classified according to where in the system it is introduced or found.

- Software
  The taxonomy for the area of software suggests particular system functions that should be scrutinised closely. Software flaws can occur in operating system programs, support software, or application (user) software.
  - Operating System
    Flaws introduced into the operating system programs.
    - System Initialisation
      Flaws in the system initialisation functions can occur either because the operating system fails to establish the initial protection domains as specified (for example, it may setup ownership or access control information improperly) or because the system administrator has not specified a secure initial configuration for the system.
    - Memory Management
      A memory management function provides control of the storage space. Errors in this function may permit one process to gain access to another improperly, or to deny service to others.
    - Process/Management/Scheduling
      A process/management/scheduling function provides control of the CPU time. Errors in this function may permit one process to gain access to another improperly, or to deny service to others.
    - Device Management
      Device management often includes complex programs that operate in parallel with the CPU. These factors make the writing of device handling programs both challenging and prone to subtle errors that can lead to security flaws. Often, these errors occur when the I/O routines fail to respect parameters provided them or they validate parameters provided in storage locations that can be altered, directly or indirectly, by user programs after checks are made.

- File Management
  File systems typically use the process, memory, and device management functions to create long-term storage structures. With few exceptions, the operating system boundary includes the file system, which often implements access controls to permit users to share and protect their files. Errors in these controls, or in the management of the underlying files, can easily result in security flaws.
- Identification/Authentication
  The identification and authentication functions of the operating system usually maintain special files for user IDs and passwords and provide functions to check and update those files as appropriate.
- Other/Unknown
  Flaws to the operating system that cannot be classified into the above categories.

- Support
  Support software comprises compilers, editors, debuggers, subroutine or macro libraries, database management systems, and any other programs not properly within the operating system boundary that many users share.
  - Privileged Utilities
    The operating system may grant special privileges to some support programs; these are known as privileged utilities.
  - Unprivileged Utilities
    Support programs that have no special privileges.
- Application
  Programs that have no special system privileges and are not widely shared as application software.

- Hardware
  Issues of concern at the hardware level include the design and implementation of processor hardware, micro programs, and supporting chips, and any other hardware or firmware functions used to realise the machine's instruction set architecture.

2.4 Methods of Attack

In the preparation phase of an attack (Heberlein 1990), the attacker gathers information: generic and specific about the Target System. Then he decides which method of attack to apply in order to achieve his goals. Some *methods of attack* that could be used against the data transferred over the TS are (Kantzavelou 1994):

Impersonating/Masquerading/Mimicking

An unauthorised user gains access to a system by posing as an authorised user. Example: using another person's password to log on (Christmas 1992).

Active Wiretapping

Connection of an unauthorised device to a communication link for the purpose of obtaining access to and modifying data (Davies 1992). This method of attack may include the following attacks categorised according to the method of modifying data.

1. False Messages
   The attacker generates false messages or control signals (Christmas 1992).
2. Protocol Control Information Modification
   A user modifies the protocol control information in the message frames in order to send them to a wrong destination or to a destination of his preference.
3. Bogus Frame Insert
   A user inserts bogus frames into the message stream either synthesised or saved from a previous connection.
4. Data Portion Modification
   A user modifies the data portion of a message to achieve his own purposes.
5. Sequencing Information Modification
   A user attacks the ordering of a message by modifying the sequencing information in the protocol frame control portion.

Passive Wiretapping
Monitoring or recording of data while the data is being transmitted over a communication link (Christmas 1992). This method is also called *eavesdropping*.

Traffic Flow Analysis
Examining the flow of messages across a network. The frequency, length and addresses (both source and destination) of messages are analysed.

Replay
"Playing back" a recording of a previous legitimate message.

Message Deletion
A user discards messages passing on a communication link.

Denial of sending a message or its contents
A user denies the fact of sending a message or its original content.

Denial of receiving a message or its contents
A user denies the fact of receiving a message or its original content.

Jamming
A user misuses the resources of the system by swamping a communication line with bogus or dummy traffic so that real messages may not be transmitted.

In conclusion, the generic attack model presented in this section involves four entities; first, the attacker who carries out an attack; second, the Target System which is the target of an attack; third, the security flaws that might be exploited by an attacker in order to attack the TS; and fourth, the method of attack used by an attacker.

3 BASIC FACTORS FOR ATTACK DETECTION

When a user requests the execution of a command (or a program) from the Target System (TS), an event is taking place on the TS. The decision whether this event is an attack or not (Denault 1994) depends upon three factors: *volume*, *failure*, and *period*, as they are described below:

Volume factor

The volume factor is the number of the occurrences of the event, i.e., how many times the user has tried the same action during the time specified by the period factor (e.g. reading files).

Failure factor

The failure factor is the number of failures of the occurrences of the event within the given event volume, i.e. how many times the user failed to carry out an action.

Period factor

The period factor is the time within which the event volume has been calculated.

The above three factors influence our decision when detecting attacks on the Target System. More specifically, when the failure is high the user is considered as suspicious, while when the failure is low the user is either non suspicious or an expert attacker who evades detection. Similarly, when the period is long, the user becomes non-suspicious, while for short periods we decide that the user is quickly trying to attack the TS. Finally, when the volume is high we decide that the user is non-suspicious, while when the volume is low the user is suspicious.

The above analysis of the attack factors behaviour permits us to define a function the value of which will give us a hint on whether a specific user is suspicious or not:

$$f(s) = \frac{failure}{volume * period}. \quad (1)$$

The behaviour of the defined function (1) was examined and this examination produced the following results which are presented in Figure 3.

(a) For given volume and failure, while the period is being increased the value of $f(s)$ will being decreased as shown in (a) (See Figure 3).

(b) For given volume and period, while the failure is being increased the value of $f(s)$ will being increased as shown in (b) (See Figure 3).

(c) For given failure and period, while the volume is being increased the value of $f(s)$ will being decreased as shown in (c) (See Figure 3).

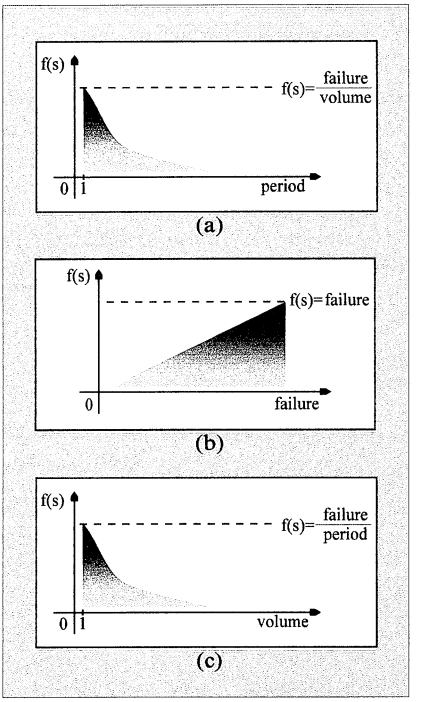

Figure 3 $f(s)$ behaviour

4 CONCLUSION

The growing spread of computer networks and distributed systems has generated a number of threats to the security of these systems. Users may use various methods of attack to damage a system. This paper has described a generic attack model which involves four entities; first, the attacker who carries out an attack; second, the Target System which is the target of an attack; third, the security flaws that might be exploited by an attacker in order to attack the TS; and fourth, the method of attack used by an attacker. In addition, it has introduced in a high level a mathematical approach in deciding whether a user is suspicious or not when using a computer system.

5 ACKNOWLEDGEMENTS

I would like to thank Dr. Dimitris Gritzalis for all the technical discussion we had for the production of this paper, and Assoc. Professor Costas Coyas for his advice in mathematics when needed.

6 REFERENCES

Anderson, J P *Computer Security Threat Monitoring and Surveillance*, Technical report, James P. Anderson Co., Fort Washington, Pennsylvania (1980).

Christmas, P *Network Security Manager*, Elsevier Advanced Technology, UK (1992).

Davies, D W and Price, W L *Security for Computer Networks: An Introduction to Network Security in Teleprocessing and Electronic Funds Transfer*, John Wiley & Sons Ltd., UK (1992).

Denault, M, Gritzalis, D, Karagiannis, D and Spyrakis, P, *'Intrusion-Detection: Evaluation and Performance Issues of the SECURENET System'*, Computer and Security, Vol. 13, No 6, pp 495-508, October 1994.

Denning, D E *Cryptography and Data Security*, Addison - Wesley Publishing Company (1983).

ECMA TR/46, *Security in Open Systems - A Security Framework*, European Computer Manufacturers Association (1988).

Gritzalis, D, Kantzavelou, I, Katsikas, S, Patel, A *'A Classification of Health Care Information System Security Flaws', Proc. of the 11th International Information Security Conference (IFIP SEC '95)*, Ellof J., et all. (Eds), Chapman and Hall, May 1995, Capetown, South Africa (to appear).

Harshall, F *Data Communications, Computer Networks and Open Systems*, Addison-Wesley Publishing Company, Third Edition (1992).

Heberlein, L, Dias, G, Levitt, K, Mukherjee, B, Wood, J and Wolber, D *'A Network Security Monitor' Proc. of the 1990 IEEE Symposium on Research in Security and Privacy*, USA (1990).

Highland, H J *'Virus Reports' Computer & Security* Vol. 12 No 4 (June 1993) pp 322-333.

ISO 7498-2, *Information processing systems - Open Systems Interconnection: Basic Reference Model - Security Architecture*, ISO (1989).

Kantzavelou I, Patel A *'Implementing Network Security Guidelines in Health Care Information Systems'*, Proc. of the *8th World Congress on Medical Informatics*, July 1995, Vancouver, Canada, (to appear).

Kantzavelou, I *An Attack Detection System for Secure Computer Systems*, M.Sc. Thesis, 1994.

Landwehr, C *'Formal Models for Computer Security', ACM Computing Surveys*, Vol. 13, no. 3, pp. 247-278, September 1981.

Landwehr, C, Bull, A, McDermott, J and Choi, W *'A Taxonomy of Computer Program Security Flaws with Examples'*, US Naval Research Laboratory, NRL/FR/5542-93-9591, November 19, 1993.

Muftic, S, Christoffersson, P, Ekberg, J, Heijnsdijk, J W J, Law-Min, F, Maroulis, D, Patel, A, Sanders, P and Varadharajan, V *Security Mechanisms for Computer Systems*, Ellis Horwood Limited (1989).

Muftic, S, Patel, A, Sanders, P, Colon, R, Heijnsdijk, J W J and Pulkkinen, U *Security Architecture for Open Distributed Systems*, Wiley Series in Communication and Distributed Systems, UK (1993).

Patel, A, Kantzavelou, I *'Issues of Security and Network Security in Health Care Information Systems' Proc. of the 12th International Congress of the European Federation for Medical Informatics*, May 1994, Lisbon, pp. 493-498.

Pfleeger, C *Security in Computing*,, Prentice-Hall International Editions (1989).

Ioanna Kantzavelou is a member of the Computer Networks and Distributed Systems Research Group held in the Computer Science Department of University College Dublin (Ireland). She received an M.Sc. by research (security in computer networks) degree at UCD in 1994, and she has worked on Secure Environment for Information Systems in MEDicine (SEISMED) project in UCD, and on a numerous other projects in the industry and other universities. Her interests are security in information systems and especially in medical information systems.