CORBA MIDDLEWARE SERVICES – ARE THEY SECURE?

E.M.Joyce
S.M.Furnell
P.L.Reynolds
P.W.Sanders
Network Research Group,
Department of Communication and Electronic Engineering,
University of Plymouth,
United Kingdom
E-mail: sfurnell@plymouth.ac.uk; ejoyce@iol.ie

KEYWORDS
Middleware, CORBA, Security, Object Services, Trader.

ABSTRACT

Middleware provides a mechanism to allow distributed heterogeneous systems to communicate, using object technology. As such, it also provides an opportunity for hackers to gain access to these systems, and so middleware needs to be secured. While middleware systems, such as CORBA, do provide security specifications, this paper will show that there are still security holes, especially in relation to the object services. The Trader service will be studied in detail, to illustrate its vulnerabilities and how they can be countered by making the service security-aware.

1. INTRODUCTION

Distributed object systems are used everywhere – the Internet, telecommunications, banking… the list goes on. However, securing such systems is not a simple task. This can be illustrated by considering one of today's middleware choices, the Object Management Group's (OMG) Common Object Request Broker Architecture (CORBA). Although there is an existing security solution, this paper will show that it has not addressed all of the possible security threats.

In CORBA, a client is an entity that wishes to invoke an operation on a target object via the Object Request Broker (ORB). The object implementation comprises the code and data that realise the target object's behaviour. The ORB receives a request and then locates an appropriate object implementation, and transmits the request data and results between the client and the target object. There is also a set of supporting services that are used to extend the ORB functionality, without which a standardised distributed solution would not be possible. It is the security of these services that this paper will focus on.

The discussion begins by providing an overview of why security is required and the main issues involved. It then proceeds to introduce the existing CORBA Security Service and some of the issues to be considered in terms of the security requirements in other CORBA services. The paper then proceeds to focus upon a particular service, namely the Trader, and the detailed security issues associated with it. These are not addressed in the current CORBA architecture and the discussion culminates by considering the modifications necessary to create a new, security-aware trading service as an example of the work that remains to be done in achieving secure middleware.

2. THE NEED FOR SECURITY

After the publicity and damage caused by incidents such as the "Love Bug" (Hopper 2000) and numerous hacker attacks, business are taking security seriously. Organisations have suffered huge losses as a result of cyber-crime. For example, on 8 December 2000, a hacker stole 55,000 credit card numbers from CreditCard.com, and when the company refused to pay any money for extortion, the hacker posted the numbers on a web-site (Chavez 2000). According to the 5th annual "Computer Crime and Security Survey" (conducted by the Computer Security Institute (CSI) and the US Federal Bureau of Investigation) such cyber-crimes are widespread, diverse in nature and on the increase (CSI/FBI 2000). 90% of survey respondents reported computer security breaches within the last year; 74% suffered financial loss as a result of security breaches and of the 42% (i.e. 273 respondents) who were willing to quantify those losses, the financial lose was estimated to be $265,589,940.

Security for any distributed system uses five basic and partially overlapping **services** as specified by the International Standards Organisation (ISO):

- **Authentication**: The security service should be able to guarantee that the user/resource is actually who/what it claims to be. One type of threat is known as a **masquerade**; that is when an entity successfully pretends to be some other legal entity and thereby gains illegal access to a resource.
- **Access control**: Protects resources from unauthorised use. It can be used on various assets, e.g., communications, data. It provides for the various types of access to a resource, e.g. read, write, update, or execution;
- **Confidentiality**: Confidentiality means being able to guarantee the privacy and secrecy of a resource such as a data file containing personnel details.

Apart from unauthorised access to a resource, the loss of anonymity or the misappropriation of messages or data records can be considered breaches of security;

- **Integrity**: Integrity of resources ensures that they are always available and correct, no matter what corruption attempts have been made. Therefore any integrity services must guard against any threats involving illegal asset/resource modification;
- **Non-repudiation**: Repudiation is the denial of an action by an entity, e.g. a user may deny sending or receiving a message. Non-repudiation forces an entity to *own up* to its participation in some action. Denial of origin, transmission, receipt or participation are all repudiation threats.

By applying these concepts, a system can be made secure. However to implement security, these concepts must be realised. Security **mechanisms**, or methodologies, must be used to actually implement these security services, e.g. cryptography, digital signatures, access control lists. The ISO also defines a security policy as a set of criteria for provision of security services. It defines what is and what is not permitted in the area of security during general operation of a secured system. It must be implemented by taking the appropriate security measures. However, no security measures, no matter how ingenious they may be, will be effective unless the user understands what needs to be protected and can determine what mechanisms are used, i.e. what the policy is. Security needs a complete and usable **administration** system that will allow users to maintain and operate security on a day-to-day basis.

According to ISO, security should be provided in a modular format (ITU). A system should be able to function independently of the security service, and when the security module is introduced the same system should now operate in a functionally similar but secured fashion. This type of thinking is practical in a centralized system, such as a mainframe, where the Trusted Computing Base (TCB) (OMG SWG 1994) is contained within a single system. The security service can monitor all requests and provide the required security functionality. However, distributed systems are more complex. Distributed objects introduce complications and the TCB is no longer contained in a single system and may need to operate across multiple systems and security domains. This results in an extended set of security requirement for a distributed processing environment (DPE) such as CORBA, and therefore the modular solution may be inadequate.

3. SECURITY ISSUES FOR SUPPORTING SERVICES IN A DPE

The CORBA Security Service (CORBASec) provides a framework for distributed object security. There are two levels of security. Level 1 provides protection for applications that are "unaware" of security, by transparently calling security functions on object invocation. Level 2 security provides more facilities and allows applications themselves to control the security provided, i.e. security-aware applications.

CORBASec currently supports certain levels of authentication, access control, confidentiality, integrity and non-repudiation. Another feature of CORBA security is the use of credential delegation between objects. It allows credentials to be propagated along an object request chain. Security is implemented by a number of objects, as shown in figure 1 below. Apart from the specific security interfaces, CORBA makes use of two objects, **Current** and **Credentials**. Current, a pseudo-object initially used by the transaction service to propagate transaction context, it is now adopted by security to propagate the security context. It does so by holding a reference to Credentials. Once a user is authenticated, a Credentials object is created. It holds information such as roles, privileges and an authenticated ID.

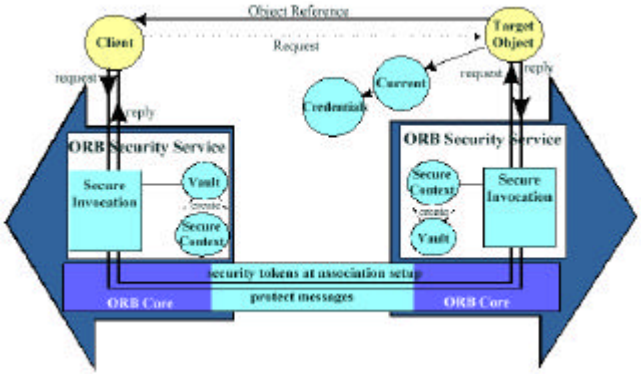

Figure 1: CORBA Security Service

Looking beyond security, to the overall CORBA standard, it currently consists of an ORB and 14 further CORBAservices (Orfali et al. 1997). Each service is implemented by a number of objects, the interfaces of which are defined in Interface Definition Language (IDL). Security is currently implemented by applying the security rules to these service objects. This means that access can be granted to a client, when requesting use of a CORBAservice object, if the client possesses the appropriate privilege attributes. However, even looking at an overview of the services some security issues become apparent. They are outlined below:

- **Persistence State Service (PSS):** The PSS stores components persistently on a variety of storage servers. Although access to the persistent storage objects are controlled, the stored data is not secured – the security service has no control over this; it would be an implementation level detail, i.e. if the data was stored in a database, the implementer would enable database security.
- **Naming Service:** The Naming Service (NS) locates components by name. Once an object can access the NS, it can access all names in the service, as there are no security restrictions. Also NSs can be federated, i.e. two naming services are linked together to operate like a single service. If the federation exists across different security domains

the client is unaware that he is crossing a domain boundary and security controls could be by-passed

- **Event Service:** This service allows "consumers" to register/unregister interest in specific events. The "suppliers" then generate information about this event and send it to the consumers via an event channel. It is a basic publish/subscribe or notification service. Security has not been defined for the event channels, i.e. access control is not available for specific events on a single channel, and there is no indication whether the channel requires encryption. Also the event service demands a certain amount of Quality of Service (QoS), i.e. guaranteed delivery, persistence of event data in the event of an event channel failure and use of logging facility. If the event channel was subject to encryption then the supporting QoS mechanisms, would also need to ensure security, e.g. the persisted data would have to be protected.
- **Query Service:** This allows a client to use query operations for attributes associated with objects, in much the same way SQL can be used to query a database of records by querying the fields in the records. It provides for asynchronous query, so that the query can be issued and the client does not have to block while waiting for a response. No security precautions have been added and so there is no way to identify what attributes a client can perform queries on, e.g. does the client have the security clearance to query a payroll attribute on an employee database. Another problem is Denial of Service, e.g. a rogue client can flood the query service with too many asynchronous or long running synchronous queries thereby causing the services to halt or crash.
- **Trader Service:** Similar in function to the NS, the Trader allows an importer to locate an object, published by an exporter, but this time is does so by identifying a set of required properties, e.g. like the Yellow Pages. A security problem could arise if some of the services offered by the trader require higher security clearance than others; there is no way of controlling access to particular offers in a single Trader.

From this it is clear that there are security issues in CORBAservices that are not currently handled by CORBASec. The above descriptions are just high-level overviews of such problems, and each case demands further detailed investigation. Therefore, a single service, namely the Trader, has been selected and examined in detail in order to illustrate the point.

4. THE TRADER SERVICE

A Trader facilitates the dynamic offering and discovery of service instances of particular types within a distributed environment. As such, it allows clients to advertise their available services and to also match their needs against other advertised services. The OMG / CORBA Trader (OMG 1996) provides the ability to match a service request, against a list of supported services provided by potential servers, as illustrated in figure 2. The exporter will advertise its available services, by notifying the Trader. The Trader keeps a Registry of such advertisements. An importer makes a request on the Trader for a particular service, specifying any conditions that need to be met. The Trader checks its Registry to find a matching service type, with corresponding conditions. The Trader then notifies the importer of the exporter and the service.

If a Trader cannot find a matching service, it will then pass the request onto another linked (or federated) Trader. The linked Trader can then check its Registry to see if it can match the original request. Therefore trading allows an importer access to multiple Trading domains. The second Trading data store is the Service Type Repository. It stores, retrieves, manages and names service types (service types are associated with a traded service and are used to describe the service. They comprise an interface type and zero or more named property types [7]) that are used in the Registry. Importers, Exporters and the Traders are all part of the Trading Community, i.e. all objects that interact to import/export services.

Figure 2: Trader Interactions

Each Trader also has Attributes. These define a Trader's characteristics, e.g. policies for scoping the extent of a search.
Traders have an important role to play in future Internet and telecommunications networks. It can perform its basic 'yellow pages' function in the world of e-commerce by providing access to internet services, e.g. a financial Trader may provide lists of financial services that a user may wish to buy over the Internet, everything from car loans to share brokerage services. The user can decide which Trader to advertise its services in, and which Trader to import services from. The Traders can be structured to provide a greater degree of choice, e.g. a financial services Trader, may be linked to a car loans Trader and a stock brokerage Trader (and many other such traders) as opposed to having the services registered directly in its own registry.

Resnick (Resnick 1997) suggested that the Trader could be used to standardise World Wide Web (WWW) facilities. There are a dizzying array of choice of search engines, web crawlers and white pages such as Yahoo, HotBot, and Alta Vista. However, these facilities, especially the search engines, lack a programmatic interface and differ not just in implementation but also in how they are accessed, how

predicates are formed and how Uniform Resource Locators (URLs) are registered. Therefore a synergy between the CORBA Trader and the Internet facilities would offer a solution. Search engines would benefit from a standardised programmatic API, represented in CORBA IDL.

It is also important to remember that CORBA is not just for Internet use. It is designed to work on any heterogeneous distributed object environment. Therefore some other possible uses of the Trader have been suggested by the Distributed Systems Technology Centre (DSTC) research group in University of Canberra, Australia [Bearman 1996]:

- real-time trading, e.g. dynamic configuration of services within telecommunications switches (combining bandwidth from local and trunk carriers to provide an end-to-end service);
- large scale trading, e.g. using trading to access network elements from network management applications for a national telephone system.

It is clear that the intense interest in security in web-based (Rogers 1998) and other distributed systems security (Australian 1998, Leahy 1999) means that Traders will have to incorporate security if they are to be included in this future. Even though Traders can make use of CORBASec to counteract threats, there are still some security holes. These Trader-Security issues are addressed below, after describing how CORBASec and the Trader operate.

5. SECURITY ISSUES RELATING TO TRADERS & TRADING

Traders, in a distributed environment like the Internet, are open to attack, just like any part of a distributed system. The following outlines the areas most vulnerable to security breaches and the security services that must be used to counteract them.

5.1 Authentication

Traders receive requests for imports/exports from members of the trading community. Like any system resource, they are susceptible to masquerade. Authentication is the service required to deal with this threat. It is a two-way process; traders, as well as importers and exporters should be identifiable and authenticatable. One possible way of achieving this is the use of certification by Trusted Third Parties (TTP). The ISO's X.509 (CCITT 1989), an authentication framework using public-key certificates, could be used. It is a hierarchy of Certification Authorities (CA) which issue signed certificates. Authentication is accomplished through the presentation of a certificate signed by a trusted CA.

5.2 Access Control

Access Control needs to be handled at two levels. Firstly, access control of the Trader itself should be considered, i.e. who has access to the Trader. Secondly, access control of service offers must be dealt with, i.e. which service offers an importer can see.

Unauthorised Trader Access
Traders should have security attributes. Two trading community objects, e.g. Trader and exporter, have access to the security domain Access Control Manager – in CORBA this would be the AccessDecision object. Therefore, AccessDecision can make decisions relating to who can have access to which Trader, using the domain's access control mechanisms and working in accordance with the access control policies.

Unauthorised Service Offer Access
Even if an importer has access to a Trader it may not have access to all the service offers the Trader holds. Some of the service offers may be of a higher security classification. Therefore, a Trader will have to hold an associated security attribute with each service offer held in the Registry.

Current Access Control Limitations
Although access control of the Trader can currently be handled by CORBA's AccessDecision object, the access control of the service offers within the Registry cannot. It would require the storage of a security attribute in the Registry itself. The reason for this is that such an attribute would be used to sort and make selections when providing service offer lists to importers. This problem is also linked to Delegation, as the security attribute would have to be set and would probably be delegated from the exporter, e.g. use the exporter's security level.

5.3 Integrity and Confidentiality

Integrity and confidentiality of data, stored or in transit, must be guaranteed in a distributed system; this has to include trading-related data.

Stored Data
Details of service offers, including an object reference, are stored in the Registry. Therefore it must be protected, as an intruder may try to gain unauthorised access to a service, by gaining illegal access to the object. Similarly, details of the Service Type held in the Repository, should be protected to ensure that intruders do not have knowledge of "how" to use the service type, i.e. interface details, parameters, etc.

It is not wise to assume that the Trader's backend data, i.e. the data stored in the *Registry* and *Repository*, is hidden behind object interfaces and, therefore, is not as vulnerable to attack as object references that are exported through the interface. Intruders do not always use legitimate access mechanisms and, therefore, the 'backdoor' entry must be considered. Such data will usually be held in persistent storage, such as a database, or flat file. Therefore the Trader, if operating as a security-aware service, should be able to guarantee that the data is secure, even when it is in storage. Cryptographic mechanisms are used to ensure that the confidentiality and integrity of the data is preserved.

However, these types of solutions are product dependent and so the only way to ensure a truly generic solution would be to use the Persistent State Service (PSS) in a secure fashion.

Inter-Community Communications
Since a Trader is operating in a distributed environment, this provides an intruder with ample access to intercept any communications between members of a trading community. From such interceptions, one may be able to re-construct Registry/Repository information. In addition, replay attacks have to be considered.

All communications between trading community members should be encrypted to ensure the confidentiality of any intercepted messages. Another form of communications security is a digital signature. The Digital Signature Standard (DSS) (NIST 1991) uses a public key to verify to a recipient the integrity of data and the identity of the sender of the data. The DSS can also be used by a third party to ascertain the authenticity of a signature and its associated data. Finally replay attacks can be dealt with by using sequencing data.

Use could again be made here of security-aware CORBAservices. In this case it would also be necessary for the Query service to be security-aware. This would allow the Trader or other trading community members to interrogate the Registry/Repository, in a secure manner.

Current Integrity and Confidentiality Limitations
Securing trader data, such as that held in the Registry and Repository, needs to be addressed. Currently these databases are not encrypted. Also trading community communications should be secured. The level of security would depend on the objects involved and their security level, as well as the level of the service offers being exported/imported.

5.4 Non-Repudiation

The trading community is made up of distributed objects, which are less predictable, due to their flexible and granular nature. There are two problems. Firstly, if the intruder is an authorised user, or is successfully masquerading as an authorised user, how can their actions be discovered? For example, an intruder can masquerade as an importer, and query Traders to find useful service offers. The processing of a monitoring database may help, by providing clues to an intruder's activities. Secondly, if adhoc interactions are taking place, how can it be proven that a specific interaction took place, if one party wishes to deny the event, i.e. accountability? Irrefutable evidence is required, i.e. a non-repudiation service.

Monitoring
All security related events should be monitored. These events are defined by the security policy. Apart from notifying an administrator, via an alarm, that an illegal action has be taken, monitoring could also provide clues to a previously unknown intruder, e.g. an importer making multiple unauthorised import requests on several Traders. However this requires data filtering to find trends that can be used to raise a system administrator's suspicions, i.e. intrusion detection.

Irrefutable Evidence
Non-repudiation is used to provide irrefutable evidence that certain events took place. For example, digital signatures can be used with audit logs to record events. Just as other system resources are subject to a non-repudiation policy, so too are all the trading community members.

Current Non-Repudiation Limitations
There are two issues relating to non-repudiation. Firstly, the current CORBASec non-repudiation service is not complete. It deals with evidence generation and verification, but does not address delivery and evidence storage. Secondly, non-repudiation is considered to be an optional service. It is available, but only to security-aware applications. It should be made available to security-unaware applications.

6. MODIFICATIONS REQUIERD FOR SECURITY-AWARE TRADERS

Both the Trader and the Security Service require modification if they are to provide a Security-Aware Trader.

6.1 Security-Aware Trader Attributes

Attributes are already used in the Trader specification to provide a framework for describing the behaviour of any OMG Trader. It is proposed that ***Security Attributes*** be added for use by the Trader. They will control the security behaviour of a Trader, by specifying which security services the Trader uses, i.e. just how security-aware the Trader is. The suggested security attributes are defined in Table 1 below.

Table 1: Trader Security Attributes

Security Policy-Attributes	Indicated function
Security-aware	All other policies are checked as the Trader is using security (at some level)
Access_trader	Includes Trader in ACL and uses authentication with trading community members, etc.
Access _service_offers	Provides access control on the service offers listed in a query
Encrypt_stores	Encrypts Registry and Repository
Encrypt_comms	Encrypts communications
Integrity_check_stores	Integrity checks Registry and Repository
Integrity_check_comms	Integrity checks communications
NR_trade	Non-repudiation of Trading related events

Security Policy-Attributes	Indicated function
Audit_trade	Audit Trading related events

For example, a Trader could be a ***Public Trader***. This means that everyone would have access to it and it would have no security applied, i.e. the *Security-aware* attribute would be set to off, indicating that all other attributes were also turned off. Alternatively a Trader may be a ***Secured Trader***. It would be *Security-aware* and have *all* other attributes turned on, i.e. it would use all the available security services. Another option is to make a Trader a ***Security-Aware Trader***. In this case the security-aware attribute would be on, and *some* of the other attributes would be on, e.g., *Encrypt_stores* and *Interity_check_stores*, but not *NR_trader* or *Audit_trader*, thereby providing a specified level of security.

6.2 Security-Aware Trader Data Structures

The two Trader data structures are the Repository and the Registry. The Repository should not have to be modified, as it will hold the security attributes in the same manner as it currently holds any other properties.
The Registry will not have to be modified either. It holds details of the instances of service offers. This includes the service type, an object reference and a set of properties held as name-value pairs. A new security property that defines the security level of a service offer will now be held in the Registry so that access controls can be applied to the offer. The exporter will specify the security level.

6.3 Security-Aware Trader Interfaces

There are eight interfaces defined for a CORBA Trader. However only one of these interfaces should have to be modified, namely the **Admin** interface. The Admin interface allows the administrator to configure the Trader, by using ***Set*** methods on the Trader's ***Attributes.*** These methods will now have to deal with the additional security attributes specified in table 1 above, to control the Trader's security behaviour. If *Security-aware* is set to on, then at least one other security attribute must be set to on also; otherwise an error will be returned on the ***Set*** method. If *Security-aware* is set to off, then all other security attributes must also be set to off; otherwise an error will be returned on the method.

6.4 An Enhanced CORBA Security Service

The CORBA security service is itself incomplete. There are certain facilities missing or incomplete. Firstly non-repudiation is only supports evidence generation and verification. It does not deal with delivery, storage or adjudication issues. Secondly, the audit facility is a simple one and does not address the needs of today's Intrusion Detection Systems. Thirdly, Secure Interoperability is also limited between security domains. Both domains must possess the same mechanisms and policies. Such limitations would mean that if two federated traders existed in different security domains, they may not be able to communicated if they have to do so securely. Finally, security administration is another problem area. Most ORB security product vendors promote the fact that they have gone beyond the CORBA Level 2 specification and provide administration services, but adequate security administration should be part of the overall standards to allow integration between products. By enhancing CORBASec to make these facilities available, it would provide better security for ORB operations. However, this is a complete topic in itself and outside the scope of this paper.

6.5 Security-Aware CORBAservice

As was mentioned earlier, if other CORBAservices were secured then a more generic security solution could be applied. If services such as the PSS, Query and Collection services were security-aware they would able to guarantee security of the data they were accessing. Then other CORBAservices, such as the Trader, could make use of them. For example, if the PSS was secure, the Trader could use it to access its Registry and Repository.

6.6 Modification Summary

Figure 3 (based on the OMG Trader), summarises the modifications that have to be made to the CORBA Trader to create a Security-aware Trader. The modifications are as follows:

- New Trader Security Attributes;
- New Registry Security Property;
- Modified Admin interface;
- Use of the Enhanced Security Service (including Enhanced Secure Interoperability Service);
- Use of security-aware CORBAservices

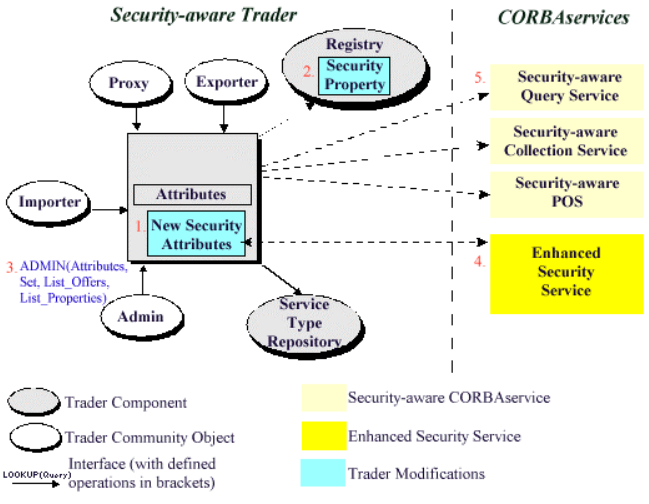

Figure 3: Modifications for Security-Aware Trader

7. CONCLUSION

In a distributed object system such as the Internet, services could be built using objects. Therefore finding the objects required, local or remote, is pivotal to the success of such an environment. A Trader can do this. However, the Trader

provides a very vulnerable point for attack, providing an intruder with access to a multitude of services. Therefore it should be made security-aware. It should be able to ensure that only authorised clients can access it, and that clients can only view the service offers which they are authorised to see. To provide a Security-Aware Trader, modifications have to be made to the CORBA Trader and Security services.

It should be noted, however, that the Trader example was only provided to act as a proof of concept. Other CORBAservices also need to be secured, and be part of the TCB, if the OMG is to provide a secure environment, where security administration does not become fragmented and, therefore, impossible to manage. The bottom line is that security cannot be completely treated as an "add-on" facility. Within CORBA, each CORBAservice has to be "aware" of security and able to interact with comprehensive security service.

REFERENCES

Bearman M. 1996, "Tutorial on ODP Trading Function", DSTC, University of Canberra, Australia

CCITT 1989, *Recommendation X.509 "The Directory-Authentication Framework"*, Consultation Committee, International Telephone and Telegraph, International Telecommunications Union, Geneva.

Chavez P., 2000 "55,000 credit card numbers stolen, posted by hacker", Nandotimes.

CSI/FBI, 2000, "2000 Computer Crime and Security Survey", CSI/FBI.

Hopper D.I., 2000 "Destructive ILOVEYOU virus strikes worldwide", CNN.

ITU, *"ITU X.700 Series – System Management"*, ITU.

Leahy E. 1999, *"Ericsson Fraud Management Solution – FraudOffice"*, Ericcson, Business Evolution and Components Seminar.

National Institute of Standards and Technology (NIST) 1991, *"Proposed Federal Information Processing for Digital Signature Standard (DSS)"*, Federal Register, v. 56, n. 169.

OMG 1996, "OMG RFP5 Submission: Trading Object Service", OMG Document orbos/96-05-06, Version 1.0.0.

OMG Security Working Group 1994, "OMG White Paper on Security", Issue 1.0.

Orfali R. et al. 1997, "Instant CORBA", J. Wiley & Sons.

Resnick R. 1997, "Intergalactic Distributed Objects", Dr.Dobb's SourceBook.

Rodgers D. 1998, "Developing Secure, Web-Based Applications", Software Development Journal.

The Australian 1998, "Mobile fraud runs riot", The Australian.

BIOGRAPHY

ELIZABETH JOYCE was born in Dublin, Ireland, and after graduating Univerisity College Dublin, worked as a Business Intelligence consultant in Europe. While working on her PhD thesis on DPE security, she joined IONA Technologies, as a security architect in the US. She is currently working in Symantec in the UK as a security solutions architect.

DR STEVEN FURNELL is the research co-ordinator of the Network Research Group at the University of Plymouth (UK), a post-graduate and post-doctoral team encompassing thirteen full-time and part-time researchers. Dr Furnell holds a first class honours degree in Computing & Informatics and a PhD in the area of IT security. His current research interests include security, Internet and WWW technologies and mobile systems. Research within the Network Research Group encompasses a range of industrial and European projects and details can be found online at http://ted.see.plym.ac.uk/nrg.