P.W.J. VAN ZYL
RAND AFRIKAANS UNIVERSITY
DEPT. OF COMPUTER SCIENCE
SOUTH AFRICA

MOSS: A MODEL FOR OPEN SYSTEM SECURITY *H6*

- MOSS II -

A Model for Open System Security

By * Mr PWJ van Zyl, Dr MS Olivier and Prof SH von Solms

Department Computer Science
Rand Afrikaans University
Aucklandpark Johannesburg
Republic of South Africa

Tel: +27 11 489 2847
Fax: +27 11 489 2138

(*) Information Security Department
IT Division Nedcor Bank
PO Box 782 995
Sandton 2146
South Africa

Tel: +27 11 881 4911

Abstract: The International Standards Organisations' Open Systems Interconnection (OSI) Basic Reference Model is an architecture that serves as the basis for the development of communication standards for world-wide distributed information systems. OSI is one of a few international accepted communication architectures that has survived the 1980's and is receiving growing support and attention. The standard contains a security architecture that discusses security services in its layered architecture and also considers mechanisms to implement these services. The Security Architecture does not propose a specific security model through which cohesion between the security services and mechanism can be provided, and is outside the scope of the architecture. Research has shown classical security models to be inadequate to address the security problem of networked environments and the problem is worsened by typical OSI environments where the nature of the distributed systems are heterogeneous. This paper presents a security model, called the Model for Open Systems Security II (MOSS-II), that utilises the concept of access paths, security agents and security profiles to address the security problems in these complex interconnected environments.

1. Introduction

In the early 1960's a great deal of standardisation work was done, nationally as well as internationally, on information systems and data communications standards. The standardisation bodies included prominent figures, like the International Organisation for Standardisation (ISO), the International Telegraph and Telephone Consultative Committee (CCITT) and the European Computer Manufacturers Association (ECMA).

In the late 1970s, the importance of meaningful international communication standards for heterogeneous environments were realised. Because most standardisation efforts worked satisfactorily in their own "closed" environments, but not in a "global" heterogeneous environment, a "master plan" was needed to bring existing standards together where possible.

Under the auspices of ISO's Technical Committee 97, Sub Committee 16 was established to develop a reference model that would provide an architecture to serve as a basis for all future development of standards for world-wide distributed information systems [7].

In early 1980's, the work finally resulted in the approval of ISO International Standard 7498 and CCITT Recommendation X.200, which specify the basic architecture of Open Systems Interconnection (OSI). [7]

ISO's OSI became a prominent communication model and is one of only two communication architecture's that survived the 1980's[8]. The other architecture is IBM's Systems Network Architecture (SNA), which is partially proprietary.

When the Basic Reference Model, ISO 7498, became an International Standard in the Spring of 1983, there was still outstanding architectural issues, of which security was one [11]. Security was addressed in part 2 of the standard. The latest version of the Security Architecture for the Basic Reference Model was accepted in 1989.

The security architecture defines the security services that should be provided between layers as well as the security mechanisms that can be used to build these security services. The security architecture also defines the proper placement of security services within the layered architecture. What is missing from the proposed security architecture is that no single comprehensive security model for Open System Interconnection is presented.

Another security problem with networked environments is that classical approaches to computer security, like the Bell and LaPadula security model, are restrictive or inadequate in addressing security [1].

Utilising a security model, called the Path Context Model for Security [1], the Model for Open System Security (MOSS) was designed [2]. MOSS is a security model specifically aimed at ISO's OSI Basic Reference Model and also builds on the Security Architecture [4] of OSI.

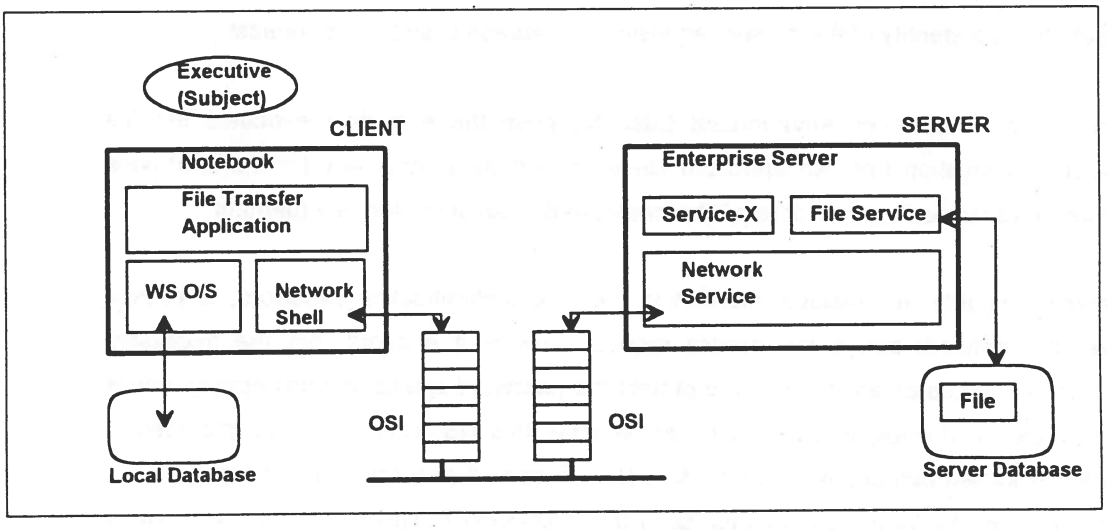
MOSS was showed to posses the capability to provide security in OSI, but is restrictive in the way security related components, like security profiles, are defined. This paper presents a revised version of MOSS, called MOSS II.

The next section starts by reviewing the basic concepts of the Path Context Model for Security [1] by using an example.

2. Path Content Model

MOSS and MOSS-II are founded on a security model called the Path Context Model for Security (PCM) [1]. The PCM model was designed to solve security problems in heterogeneous networked environments where classical security models fall short.

To highlight the typical problems, the figure below depicts a client server environment where certain data and other services are shared from an enterprise server. The client and server are connected via a network using the OSI stack. The typical security problems within such an environment are discussed.



Suppose a company executive regularly connects to the enterprise server with a notebook to download or copy information for further processing. The reason to down load the information is that the notebook will not always be connected to the network and therefore the information shared by the enterprise server will be unavailable. Because sensitive information is stored on the notebook, it is further expected that the notebook possessed a secure operating system.

To copy the information, the following steps are performed:

Step 1) The executive log on to his notebook and is authenticated by the security system (operating system) of the notebook. A successful log on will result in the start-up of the network shell to enable the initiation of server requests, like Remote Procedure Calls (RPCs) and logical disk redirection. The logical disk redirector allows the notebook to see the enterprise server's disk as an extension of its own.

Step 2) Next the executive starts an application that will perform a file transfer and copies the required file(s) from the enterprise server to his local disk.

The copy or download operation described is simple and straight forward, but it is not without security problems. These are presented below:

1) User authentication: The enterprise server protects the files with its own security system. If the executive is authenticated by his notebook, then how is he authenticated by the enterprise server? He could for example have another user account on the enterprise server which implies dual log on, or the file transfer application could identify and authenticate itself to the enterprise server. The latter is usually what happens [1] with the result that the identity of the person requesting the access is lost in the request.

2) Source and destination environment trust: Suppose the executive executes the file transfer application from an untrusted client, then there is now way for the enterprise server to recognise that the data will be transferred to an untrusted environment.

3) Network security: If sensitive information, like the authentication password, is passed over the network during the access request, how is it ensured that the necessary protection measures are activated to protect the password against eavesdropping, and is it possible for the requested party to test whether this protection has been provided or not? It is known that popular file transfer software as well as terminal emulation software that are used by large corporations, send the passwords in clear text over the network when the software authenticates itself to the enterprise server.

The security problems identified above are by no means exhaustive and only identifies a few of the concerns. For more information refer [1].

A solution to these security problems can be found in the "path" that is followed during the file transfer request. To address these security problems, the Path Context Model uses the concepts of baggage collected on this access path and of security profiles. The security profiles of the sensitive files define access paths that are valid and would allow access as well as those paths that are invalid and that results in access denial. A typical security profile for sensitive information would therefore specify that if a request is received to copy data over the network, then the request must be from a "trusted" environment, the password must be encrypted, only company executives can request the transfer and the destination of the data must be to a trusted environment. These issues are addressed in the PCM.

Below is a condensed definition of the PCM security model [1].

PCM definition

The PCM is based on two basic concepts:

1) First, the concept of an access path which is formed by the various components that need to be activated or utilised in order for a typical user's request to be executed. The initial originator of any service request is always an individual termed a primary access say A1 (the executive). Assume that software components S1, S2,...,Sn (file transfer application, network shell, OSI stack, etc.) in a single domain D1(notebook client) are required to access some object, says a file or a block of information on a file, O1 (the file), in domain D2. Then A1,S1,S2,..,Sn,O1 is defined as an access path to O1. The notation implies that A1 initiates S1, S1 in turns activates S2 and so on. The following are relevant:
   i) n is finite for any environment.
   ii) There may be multiple access paths to O1.
   iii) The total number of access paths in any environment E is finite.
   iv) Any given Si is a software component which utilises or activates hardware or other resources. Typically Si consists of a set of tasks, processes, routines or programs. Having the ability to control access via the software is in line with software or firmware based directions in computer engineering.
   v) The access path concepts applies equally to cross domain environment. In this case a corresponding Si simply exists in another domain.

2) PCM is a model that accommodates both preventative and detective modes of security. Preventative security is performed in PCM via the principle of baggaging. Baggage is defined as the minimum amount of information that has to be collected and must accompany the access request on its route (S1,S2,...,Sn) in order to enable full preventative security. Preventative security therefore implies the ability to prevent unauthorised access by subject A1 to object O1 via access path S1,S2,..,Sn.

PCM consists of three components. They have been defined as the following distinct and mutually exclusive elements.

1) A set of restrictions which specify the security procedures to be applied, termed the security profile.

2) A mechanism which collects information against which the security profile can be applied, termed the baggage collection vehicle.

3) The mechanism which performs the actual checking, termed the validator

To summarise, PCM operates as follows; Every protected object has a security profile. As a request proceeds towards a protected object, baggage is collected about the access path taken by the request. When the request reaches the object, the offered baggage is compared against the object's security profile, and depending on the outcome, access is granted or denied. The PCM is further defined using formal grammar theory, specifically Random Context Structure Grammars (RCSGs) [1].

3. MOSS II: The security model

Although the PCM goes a long way to solve some of the security problems, the following are outstanding security issues:

1) Objects deny access: A full access path is executed before the access is denied. For example, if the password over the network was not encrypted, the request will still continue up to the file before it will be rejected.

2) Consistent security: After the file was transferred the same security must still apply. Usually information will be transferred to another secure environment, but the protection profile will change. Different accessors (subjects) are immediately allowed to access the data (object) and the original security profile rules are lost.

3) Pre request enforcing security: If for example the security profile requires the password to be encrypted over the network line, then it must not be a separate security action, but must integral to the file transfer request.

4) Post request enforcing security: If information is transferred over a network after the request was successful, then the protection usually does not end. If the information was protected at the one system, it usually will require protection when sent over the network. Some mechanism is required to enforce the protection after the access request was granted. This protection is also part and parcel of the initial file transfer request, and must therefore be automatic and transparent.

To address the above security issues, the PCM was extended and MOSS was created [2]. A revised version of MOSS, called MOSS-II, is presented here. The main reasons for the revision of MOSS are:

1) PCM profile specification: The PCM utilises formal language theory to specify the security profiles, called Random Context Structure Grammars (RCSG). RCSG were also used to specify the protection profiles in MOSS, and therefore restricts the specification of profiles to a specific method. MOSS-II does not prescribed a specific specification mechanism for the protection profiles, which opens the door for more research and the use of other technologies, like knowledge bases used in expert systems.

2) Private profiles: PCM and MOSS present the idea that security profiles for objects are defined and used by the validator. MOSS-II proposes that all objects possess a security profile that "lives" and "moves" with them. MOSS-II also proposes that all subjects possess a security profile that "lives" and "moves" with them.

The different elements of MOSS-II are now described.

1) Conformance rules: These rules ensure that access requests comply with the security requirements for the access request as well as the result. For example, the conformance rules will specify that passwords sent over the network must be encrypted and that the origin of sensitive information be proven as well as the information's confidentiality during network communications. The possible security measures and mechanisms will be expanded on later.

2) Access rules: These rules contain the conditions before information can be accessed. These rules may for example specify that the data may only be accessed from a specific place by a specific group of users using a specific program and that all communications

should be encrypted.

3) Security profiles: Two types of security profiles are identified - Subject security profiles and object security profiles. Subject security profiles are attached to subjects and contains only conformance and access rules peculiar to that subject. Object security profiles contains conformance and access rules that belong to objects. How these profiles are used will be expanded on later.

4) Security agents: At every point where baggage is collected, a security agent is activated. The security agent uses the baggage collected and the security profile of either the object or subject, to activate conformance measures or access control rules. Security agents have the capability to update the baggage as well as the security profiles. The main functions of the security agent are:

    1) Update the baggage on the current access path.

    2) Act as a validator by using the collected baggage and access rules in the security profile(s) to perform preventative security.

    3) Act as a security enforcer by using the collected baggage and the conformance rules in the security profile(s) in order to maintain the three characteristics of computer security [10] which are secrecy, integrity and availability.

    4) Act as a self enhancement agent by having the ability to update the access as well as conformance rules in the security profile(s). This is one of the most powerful features of the security agent, because it allows the agent to "learn" itself by updating its security "knowledge" of the environment. Different possibilities exists here

        i) the agent detects conflicting rules in its conformance and access rules
        ii) the agent uses the access rules to update the conformance rules, for example access are denied because a request was not performed over a specified and encrypted communication link, then the security agent denying the access could update the conformance rules to ensure that the necessary encryption would be performed next time the request is executed. The deny access result can now be changed to retry, which would increase availability.

The last important component of MOSS-II to be covered is its integration with the OSI Basic Reference Model and its utilisation of OSI Security Architecture. The reason for OSI

integration is because security cannot be effective only on an application layer. To illustrate this, the following arguments are presented

1) Path enforcement: To control the path an access requests follows over the network or the path information follows over a network, is only achievable on the network layer of OSI (Layer 3), which is the routing layer.

2) Path baggage collection: To be able to determine the different nodes through which data or a request moved over the network, baggage needs to be collected on the network layer (layer-3).

3) Point to Point encryption: End to end encryption can be performed on application level but not point to point encryption.

4) Origin / destination confidentiality: It is not possible to protect the origin / destination addresses of data / request only at application layer.

The above arguments therefore proof that to have total control over security in a network environment, application layer security only is inadequate.

The first problem to solve is to determine within the framework of the OSI architecture, where the suitable places are to locate the MOSS-II security agents. This is done in the next section.

4. Open Systems Interconnection

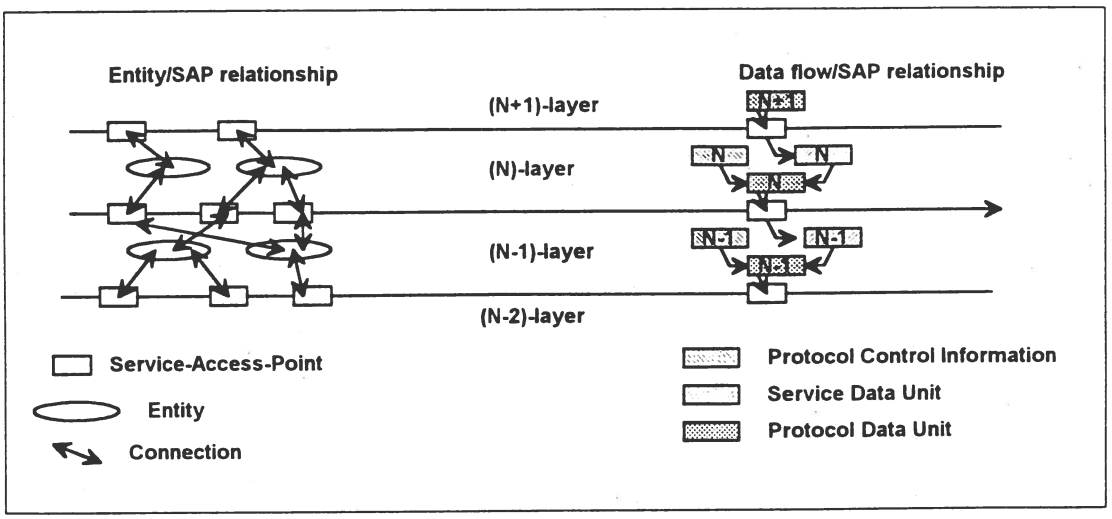
The information that is presented in this section is to motivate and support the security concepts used in MOSS-II. Some familiarity with the basic concepts of the OSI Reference Model and Security Architecture is assumed and therefore they are not discussed.

4.1 The Basic Reference Model

The OSI Basic Reference model, is divided into two parts. The first part describes the elements of the reference model and the second part the services and functions of each of the seven layers.

The elements of the basic reference model which are key to MOSS are the relationship between entities within layers as well as the flow of data between layers. On the left hand

side of the figure below, the entity / service access point (SAP) relationship is depicted and on the right hand side, the data flow / service access point relationship is described.



(1) Entity / SAP relationship

An entity in an (N)-layer can make use of services provided through one or more (N-1)-SAP(s) to provide services to the (N+1)-layer via one or more (N)-SAP(s). Entities can be connected to more than one (N)-SAP in the (N)-layer as well as to more than one (N-1)-SAP in the (N-1)-layer. An entity can be disconnected from a SAP and reconnected to another SAP. An (N)-SAP can only be connected to one (N)-entity and one (N+1)-entity. (N)-peer-entities communicate via an (N-1)-connection made through (N-1)-SAPs.

The essence of the above which is also the crux of the MOSS-II security agent placement in OSI, is that all services are provided through SAPs and that by OSI definition, entities within layers communicate by connections made via SAPs.

(2) Data flow / SAP relationships

The right side of the figure above depicts the flow of information through the SAPs in the OSI layers. On the (N+1)-layer, a protocol data unit (PDU) is made up and is sent through services provided in a SAP down to the (N)-layer. In the (N)-layer, the data is received as an (N)-service-data-unit that combines the data unit with (N)-protocol-control-information. The resultant data unit is called an (N)-protcol-data-unit that is sent via services in an (N-1)-SAP to the (N-1)-layer. Recursively the above is defined by:

N-PDU = N-PCI + N-SDU

N-SDU = (N-1)-PDU

Again the observation that is made, is that all control information as well as data units are passed via Service Access Points between layers.

(3) Conclusion

It is therefore safe to intuitively decide that to gain utmost control over the interaction between layers, without involvement in the abstract modelling of the layered architecture of OSI, it will be best to perform or activate the needed security agents at or above SAPs.

MOSS therefore extends the scope of a SAP so that at every layer within the OSI basic reference model where SAPs are defined, they may be extended to include Security Agents. Before any information is sent through that SAP, the Security Agent will take control and perform the necessary security measures.

Extending the SAPs is much the same as inserting security layers between each of the seven communication layers. The concept of inserting entities between communicating layers, (N) and (N-1), for security purposes is also proposed by other researchers in this field [5,6].

4.2 ISO 7498-2: Security Architecture

As has been said, the security architecture defines the security services that should be provided between layers as well as the security mechanisms that can be used to build these security services. The security architecture also defines the proper placement of security services within the layered architecture.

The table below, taken from ISO 7498-2, is used to describe which services are appropriate for which layers of OSI.

	OSI Layer						
Security Service	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Peer Entity Authentication			y	y			y
Data Origin Authentication			y	y			y
Access Control Service			y	y			y
Connection Confidentiality	y	y	y	y		y	y
Connectionless Confidentiality		y	y	y		y	y
Selective Filed Confidentiality						y	y
Traffic Flow Confidentiality	y		y				y
Connection Integrity with Recovery				y			y
Connection Integrity without Recovery			y	y			y
Selective Field Connection Integrity							y
Connectionless Integrity			y	y			y
Selective Filed Connectionless Integrity							y
Non-repudiation, Origin							y
Non-repudiation, Delivery							y

The main objective of MOSS-II is to activate and perform the above security services, using security agents, transparent to the user of the service.

5. MOSS-II Example.

In this section, the example in section 2 is taken to show how MOSS-II can actually provide security in this environment totally transparent to the user.

First the following security profiles are defined. The rules are defined using normal English and are not formally specified. The reason is that MOSS-II does not yet prescribe a specific profile specification format.

Subject (Executive X) security profile - Conformance rules

SC1) When a password is sent over the network, then it must be encrypted by activating the Selective Field Confidentiality Service on layer 7 (application layer).

SC2) If files are transferred from the enterprise server then always activate the Origin No-repudiation service on layer 7 (application layer).

Subject (Executive X) security profile - Access rules

SA1) None

Object (Sensitive File-F) security profile - Conformance rules

OC1) If file is moved over network then the a Connection Confidentiality service must be activated on the application layer (layer 7).

OC2) If file is moved over network then activate the Traffic Flow Confidentiality measure on the network layer (layer 3).

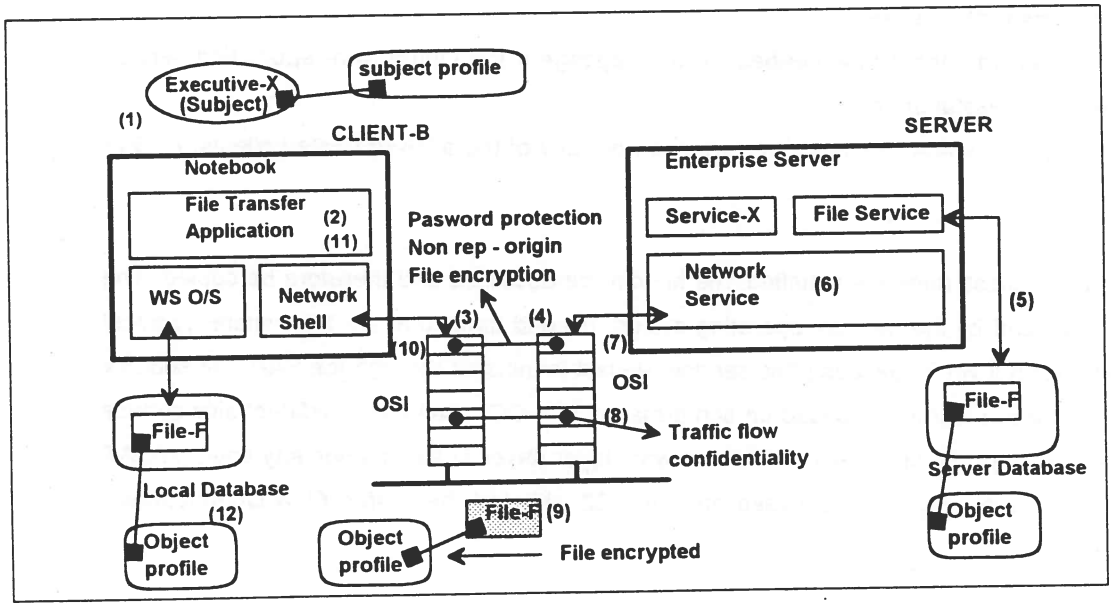
Object (Sensitive File-F) security profile - Access rules

OA1) This file may only be accessed by executives.

OA2) If file accessed over a network then a Origin Non-repudiation service need to have been executed successfully.

OA3) Use origin identified in rule OA2, and ensure that it is a trusted client, A, B or C.

With reference to the figure below, the functioning of MOSS-II is now described.



When the executive log on and executes the file transfer application, the following happens: First the executive's identity is collected in the baggage (1). Now the executive executes the file transfer program (2). The file transfer application executes (3) a FTP (File Transfer

Protocol) on the application layer in its local OSI stack. Next the security agent at the application layer SAP will be activated. This security agent will now perform the following:

SC1) Encrypt the password by activating the appropriate Selective Field Confidentiality service (see previous table) in layer 7.

SC2) Activate the Origin Non-repudiation service (see previous table), because the FTP is activated.

The initial access request now continues to filter down the OSI stack. At each security agent where a rule is performed, the required baggage will be collected.

At the enterprise server side, the request filters up the stack and reached the application layer (layer 7). Because Origin Non-repudiation was specified, a security agent is activated (4) and will verify the origin. If this security measure is successful, it is registered in the baggage.

The request reaches the file (object) and again a security agent is activated (5) to perform the following:

OA1) Check if the access request was done by an executive. Because the identity of the initial access requester is contained within the baggage, the security agent can decide to allow further access.

OA2) The security agent now verifies via the baggage if the Origin Non-repudiation service was successful or not.

OA3) The origin is now verified to ensure it is from any of the allowed trusted clients, A, B or C.

If all these access rules are satisfied, the file can be accessed and therefore be copied. The file is now read by the network operating system (6) and handed to the file transfer protocol on application layer 7. Because this service request is initiated through the SAP, the security agent (7) takes control and based on conformance rule OC1 starts the confidentiality service (typically encryption) on layer 7. On the network layer (layer 3) again a security agent at SAP level will be activated (8) and based on rule OC2 will start the Traffic Flow Confidentiality service.

The sensitive file is now sent over the network encrypted (9) and its origin and direction will be kept confidential. At the client, the file is decrypted (10) and handed to the file transfer application (11). What is important to note is that the file's security profile, was copied with the file and therefore all the access and conformance rules will still apply (12). This satisfies the requirement of consistent security mentioned in paragraph 3.

6. Further research

The Model for Open System Security II lays the foundation for a wide range of future research projects. Below is a summary of areas identified by the authors that could warrant further research:

- ☐ Security profiles should be deduced / inferred or generated from the security policy. Research in this area is currently being done [9].

- ☐ MOSS-II Component protection. In order for MOSS-II to be an acceptable security model, it is important to ensure the security of its elements. The most important of which is the protection of the Security profiles, the security agents and the baggage that are collected.

- ☐ Security profile specification: Using formal language theory a interesting field of research is the specification of the conformance and access rules in the security profiles using formal languages and automata.

- ☐ Security agent modelling: Another area of research is the model that are used to depict the security agent, examples are Procedural modelling, Object oriented, Expert systems / knowledge base and Artificial intelligence like Neural networks, etc.

- ☐ Mapping of security models: The mapping of other security models, like DCE into the framework of MOSS-II would be valuable research to show differences and weakness, if any, with the security models.

Conclusion

This paper highlighted the security problems within interconnected environments. The basic concepts of the Path Context Model (PCM) for Security was presented and showed to address certain of the security problems within interconnected environments.
The next step was to extend the PCM for security to the Model for Open Systems Security II (MOSS-II) to address security issues that the PCM could not address. Finally, because of the importance and significance of the Open Systems Interconnection (OSI) Basic Reference Model, for networked environments, the Model for Open Systems Security II was also integrated into OSI. By way of example MOSS-II's power and capability to provide security in a OSI interconnected environment in an easy and transparent way, was illustrated.
The paper closes by looking at possible areas for future research.

References

[1] WH Boshoff and SH von Solms, "A Path Context Model for Addressing Security in Potentially Non-secure Environments", Computers & Security, v8, 1989.

[2] PWJ van Zyl, SH von Solms and MS Olivier, "A Model for Open System Security", Proceedings of UniForum Asia, 1992.

[3] ISO 7498-1, "Information processing systems - Open Systems Interconnection - Basic Reference Model", ISO, 1984

[4] ISO 7498-2, "Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture", ISO, 1989

[5] A Patel & PW Sanders, "A Comprehensive, Integrated, Security System", International Symposium Organised Jointly by the South African Institute of Electrical Engineers and The Computer Society of South Africa, 1991.

[6] W Fumy & M Leclerc, "Placement of cryptographic key distribution within OSI: design alternatives and assessment", Computer Networks and ISDN Systems, v26, 1993.

[7] Proceedings of the IEEE, "Scanning the Issue" The special issue on Open Systems Interconnection (OSI) - New International Standards Architecture and Protocols for Distributed Information systems.

[8] GJ Knutt, "Open Systems: Integration, Interpretability and Portability", Prentice Hall Series in Innovative Technology, 1992

[9] D Pottas & SH von Solms, "MAPS - Model for Automated Profile Specification", IFIP/Sec '92, Singapore, May 1992.

[10] CP Phleeger, "Security in Computing", Prentice-Hall International Editions, 1989.

[11] JD Day & H Zimmermann, "The OSI Reference Model", Proceedings of the IEEE, v71 n12, 1983.

-oOo-