A LAYERED SECURITY ARCHITECTURE: DESIGN ISSUES

Heiko Tillwick, Martin S Olivier

Information and Computer Security Architectures (ICSA) Research Group

Department of Computer Science, University of Pretoria, South Africa

heikotillwick@softhome.net, molivier@cs.up.ac.za

ABSTRACT

System security is a key technology to the development and deployment of IT applications and services in a growing global network. Security is critical at various levels of the system. However, security solutions typically address a very specific vulnerability with little relation to the larger picture of secure information systems. Organisations have successfully implemented these solutions without knowing if all security requirements have been met or what impact these solutions have on other parts of the information system. The focus of this paper will be to identify the various layers that exist in large distributed systems, and to lay the groundwork for defining security requirements for each layer allowing for a mapping of security implications that each layer has on other layers. This will result in the design of a layered security architecture which could assist organisations in mapping out all required or successfully implemented security requirements at various levels of information systems.[1]

KEY WORDS

secure information systems, layered architecture, security model, system layers

[1]This material is based upon work supported by the National Research Foundation under Grant number 2054024 as well as by Telkom and IST through THRIP. Any opinion, findings and conclusions or recommendations expressed in this material are those of the authors and therefore the NRF, Telkom and IST do not accept any liability thereto.

A LAYERED SECURITY ARCHITECTURE: DESIGN ISSUES

1 INTRODUCTION

System security is a key technology to the development and deployment of IT applications and services in a growing global network. Daily patches and bug fixes of operating systems, browsers and other commonly used applications highlight the importance of security and security assurance. Technologies such as firewalls and antivirus software have become extremely popular in the security domain. They however only solve very specific problems and in no means provide security assurance. Instead, their importance has been overemphasised, or rather: other aspects of system security have been neglected or have received insufficient attention.

Security features are often implemented as a result of a direct threat. Systems are designed with functionality and efficiency in mind but often fail to make a thorough investigation of the security requirements of the application as well as the underlying system. This inevitably leads to patches or security software or hardware being used at later stages of the development cycle, often after a vulnerability has been exploited.

This lack of coordination between security requirements and security measures might lie with the fact that there are currently no clearly defined guidelines as to the requirements of the individual parts of a system. Many organisations such as financial institutions have clearly defined business requirements. However, these business requirements have little relation to the specification and the implementation of the system. This clearly indicates that some process is required to formalise the implementation of a secure system and achieve a state of security assurance.

Up to now there has only been limited research on a model for system-wide security that profits from a holistic approach. Most technologies, protocols and models concentrate on a very specific area. The Open Systems Interconnect (OSI) model, although it is not a security model, has been designed and utilised with great success within the networking domain. It enables the abstraction of the individual core functions of network communications resulting in a more modular communications approach. A similar model is required to describe the individual parts of a system.

The proposed model would attempt to describe the system as a whole: from the storage through to workflow and communications. As with the OSI model an ordering can be achieved. This can be a strict ordering or a more lax ordering, resulting in a clearly defined layered architecture.

It will be the aim of this paper to identify the possible layers of an electronic information system and define them in such a way that the facets of a system can be classified into a specific layer. Once the layers have been clearly defined it will be possible to analyse how each layer will address certain aspects of security and moreover, it will be possible to position the specific security-enhancing technology within the context of the layered security architecture. This paper will however not present the architecture in full but will lay the groundwork for future work on the layered architecture.

The paper is structured as follows. Section 2 briefly reviews the cornerstones of security and the security services as defined by ISO 7498-2 [3]. Section 3 takes a look at system entities. Section 4 presents the layered security architecture. Section 5 takes a look at some related work. Section 6 concludes the paper.

2 BACKGROUND

2.1 Security services

Various forms of security-enhancing technologies have received research attention. Security has traditionally been defined in terms of the three cornerstones of security: confidentiality, integrity and availability [5]. Confidentiality addresses privacy related issues, integrity prevents the unauthorised modification of information and availability prevents the unauthorised withholding of information. While confidentiality has been the main focus of previous technical solutions, it is the balance between all three dimensions that is required [7]. Neglecting to address either one of the cornerstones will result in inadequate security. On the other hand, a state of security assurance can be achieved if all of the cornerstones have been addressed.

In addition to confidentiality, integrity and availability, ISO 7498-2 [3] identifies a range of types of security services and mechanisms. The standard defines five security services: identification and authentication, authorisation, confidentiality, integrity and non-repudiation. The standard thus adds the dimension of access control as well as the notion of non-repudiation. Access control has always received ample interest in the security domain. Non-repudiation has become increasingly important as electronic transaction systems have become wide-spread in use.

For a system to be secure it is important to address the cornerstones of security as well as provide for the five security services. This should ideally be done during the design phase of the system as well as during the implementation and maintenance phases.

Numerous technologies exist that provide for usually one or two of the services. Elsewhere [6] studies are made to determine the usefulness of some of the existing technologies with respect to their system domain. These technologies would be evaluated and compared against each other to determine which would best address the needs for a security service within the context of a specific information technology system. An example of such a case would be the implementation of encryption algorithms. Encryption algorithms address the confidentiality security service. The choice system designers need to make is whether they would need symmetric or asymmetric encryption or both. Once this has been established a choice has to be made between the specific encryption algorithms available (for example DES, Rijndael, RC2, RSA,)

In large distributed systems, however, the required security services become almost unmanageable. These complex systems have many points of failure and are exposed to a multitude of different vulnerabilities. To address the security issues of large, distributed systems, the notion of vulnerability scanning and intrusion detection has somewhat superseded the notion of security assurance through proper system design. Systems are designed in response to possible vulnerability incidents as opposed to directly addressing the security services.

To address the security services from the design process right through to implementation and maintenance one requires a better understanding of the individual parts of a system.

2.2 System entities from a workflow perspective

An in-depth knowledge of the system entities will be a vital component of the layered security architecture. Research in the field of workflow has required the identification and classification of workflow related entities, and could assist in defining information system entities. The nature of workflow systems bear sufficient resemblance to information systems purely by means of the functional and process related nature of workflow and information systems. This resemblance will become clearer when analysing the workflow entities.

Workflow can be defined as a set of tasks and dependencies that control the coordination requirements among tasks [1]. Elsewhere [2] it has been suggested that workflow systems consist of tasks, events or task dependencies, agent privileges and documents or data. These entities present a very basic overview of workflow and allow for modifications or improvements on various aspects e.g. access control [4].

With the knowledge of security services and some related work done in the field of workflow one can start laying the groundwork for modelling a layered security architecture. The first step will be to define the information system entities.

3 SYSTEM ENTITIES

In order to successfully model a security architecture one needs to know the information system entities. We will adopt some of the entities used for workflow and expand on them, enabling us to compose an abstract view of an information system.

In order for us to create such a view we will take a holistic approach. A system consist of a set of tasks, which represent the work to be done. These tasks are coordinated by a set of events, which we will refer to as task dependencies. An agent is the initiator or requester of one or several tasks. The successful completion of a task will depend on the privileges of the agent as well as the task dependencies.

An example should help in clarifying the difference between privileges and task dependencies. In the medical field a doctor may have sufficient privileges to prescribe certain medicines but a medical or credit report is required before he can do so. The task of generating a medical or credit report is required before a prescription can be issued by the doctor i.e. the task of prescribing depends on the task of generating a report. To illustrate that task dependencies are not exclusively related to workflow we will consider an example of trying to transmit over a network. Successful transmission depends on the existence and status of a physical link.

It is important to note that privileges need not be associated with an agent. Privileges can be assigned to tasks i.e. a task might have certain privileges to perform some action regardless of who the agent is. This will have a significant impact on the design of our architecture. It is often the cause for security loopholes in current systems - when privileges have been set for the one and neglected for the other.

Other important system entities are data, communication channels as well as agent roles. Data and the communication channels are well-known and well-studied aspects of an information system and have received plenty of attention in the realm of information security. A role is a semantic construct useful in access control. It can be argued whether or not it is a system entity. For the purpose of modelling a security architecture, we will consider it a property directly linked to privileges and system agents. It will feature extensively in the security services relating to access control.

The diagram in Figure 1 is a basic model of the identified system entities. Note that the communications channel has been left out and that the diagram does indeed bear close relation to a workflow model. It could be argued that all arrows in the diagram represent a communications channel. However, it is not the aim of this paper to create a system entity framework. It should be noted that this diagram will be central in the design of a layered security architecture. The diagram itself will not have any further significance in the layered architecture.

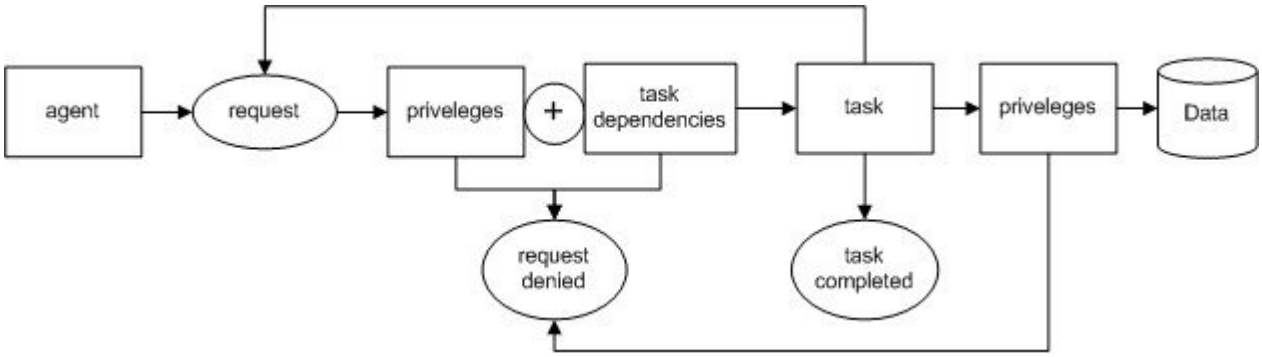

Figure 1: Relationships between system entities

4 THE LAYERED SECURITY ARCHITECTURE

4.1 Identifying the layers

Now that one has a formalised notion of the system entities one can focus on defining a layered security architecture. Since we are looking at constructing a security-centred model, we will look at the different dimensions inherent to information systems. Different dimensions require different security services. This will become clear as the discussion on defining the layers progresses.

Our usage of the term dimension, within the context of information systems, will refer to aspects relating to time, location as well as the change of location with time. These three aspects are not derived from the three physical dimensions (with time being the fourth dimension), but reflect important properties in the processes of an information system.

Workflow can best be ascribed to the dimension of time since workflow addresses sequence issues. The notion of time plays an important role in workflow - a task can only be performed after another task has been completed. Data can best be ascribed to the dimension of location while a communication channel effectively accommodates for the distribution of data i.e. the change of location with time.

We therefore continue to define some of our layers according to the system dimension. These layers will be called the temporal layer, the distribution layer and the data layer. The temporal layer will address time-based security and will feature workflow related solutions. The distribution layer will address communication-based security. The Open Systems Interconnect (OSI) model will best fit into the distribution layer. The data layer addresses secure data storage. An obvious technology for the data layer would be a database management systems (DBMS).

In addition to the temporal, distribution and data layers we introduce the resource, application and application domain layers.The resource layer is used to include factors that are instrumental in the functioning of a system, but which operate independently from the rest of the system. The resource layer should not be confused with human resources or any other organisational resources. Although the human factor plays a vital role in information security management, it will not be considered any further for the layered security architecture. This is because the human factor does not assist in defining a technology-oriented framework of an information system. The layered security architecture will take a technological perspective and not a management nor an organisational perspective.

To complete the definition of our layers we still need to define the application and the application domain layers. The application layer will provide for an external view of the system. It will address user and user group roles, policies as well as any inter-organizational issues. Finally, the

application domain refers to the context in which a system operates. Examples of an application domain would be the internet.

Now that the layers have been identified we will consider an ordering of the layers.

4.2 On the order of the layers

The resource layer was said to operate independently from the rest of the system and should as such be the lowest or first layer. The functional operation of all other layers will depend on the availability of the resource layer. The distribution layer facilitates the communication of data and should therefore be assigned a higher ordering than the data layer. The temporal layer controls the flow of information. We will therefore order the temporal layer above the distribution layer. The application domain layer will be the top-most layer with the application layer just beneath it.

While it can be argued that system designers have limited control over their system resources or over the application domain, we claim that these layers operate independently. These layers have been added to complete the ordering of the layers but will not be considered any further in the layered security architecture.

We can therefore focus on the four remaining layers. We will refer to the basic ordering of the layering as the layered framework. The framework is depicted in Figure 2.

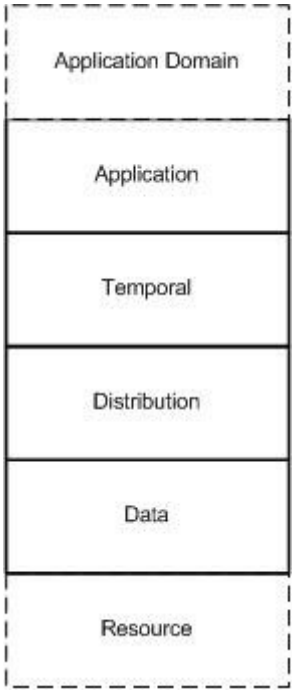

Figure 2: The layered framework

4.3 Mapping the layers to security service requirements of system entities

The security architecture will look at the aspects of identification, authentication, authorisation, confidentiality, integrity and non-repudiation. This will be done at each individual layer. We will choose one system entity, as defined in Section 3, from each layer and ask the question: which layers are responsible for providing the specific security service for a particular system entity?

Preliminary results indicate towards the mapping depicted in Figure 3. A discussion on Figure 3 follows.

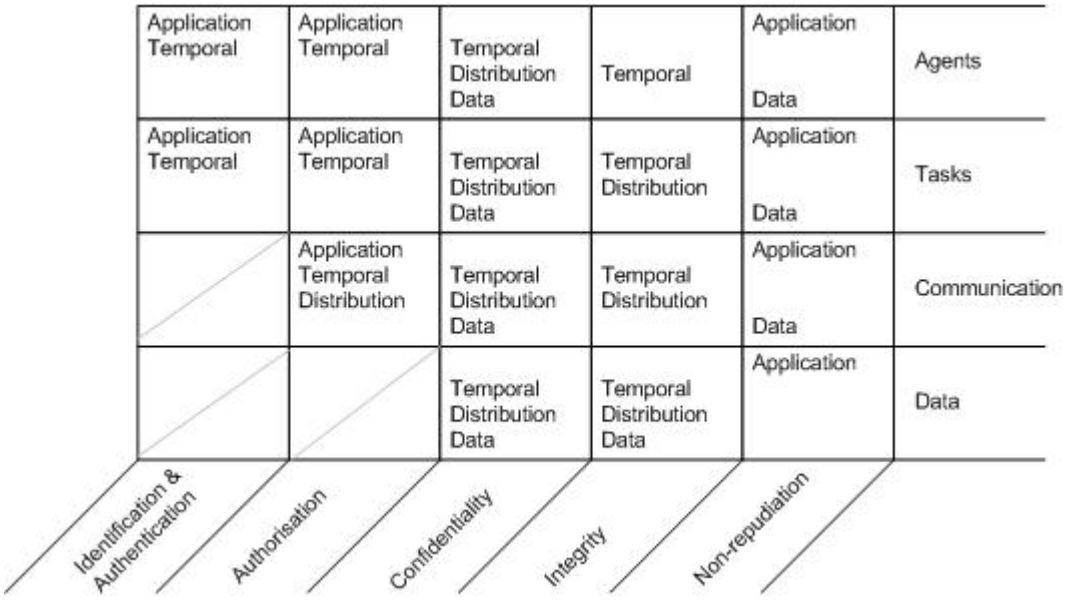

Figure 3: Mapping of layers to security service requirements of system entities

We will start with the data system entity and work our way upwards, considering possible layers for each security service. Data itself does not need to perform access control. The identification, authentication and authorisation services are therefore not relevant services required for data. These services are most often associated with agents or more specifically, users or user groups. The identification, authentication and authorisation of data will not be considered.

The confidentiality of data can be addressed in the data, distribution and temporal layers. It should be clear how the storage and distribution aspects need to be sensitive to confidentiality requirements. This is often, but not exclusively, achieved through cryptography. The temporal layer needs to assure that access control has completed successfully before the data is disclosed. A failure of any three of these layers to ensure confidentiality can result in the confidentiality of data being compromised.

The application layer plays no part in ensuring confidentiality. An application, classified as a system agent in our definition of system entities, can request or provide information but cannot assure its confidentiality. One might consider the scenario whereby two applications share information. Could one argue that at an application level, one application can ensure that the shared information has been kept confidential? We argue that this is a trust issue and not a case whereby the application layer is instrumental in providing confidentiality. The same argument applies to providing confidentiality to the communication, tasks or agent entities. The application layer will therefore not be considered as an option for providing the confidentiality service to any of the system entities.

The definition of integrity implies the prevention of unauthorised modification. When considering data integrity one needs to determine which layers prevent the unauthorised modification of data. Data integrity can be provided by the data and the distribution layers. The data layer should ensure proper storage by preventing any data corruption. The distribution layer has to ensure that for each transmission, the sent and received data are identical.

Preventing data corruption is a more difficult task and often depends on the correct functioning of the hardware. Because hardware is part of the resource layer, we take note of the fact

that the resource layer does indeed feature in the layered security architecture. As stated before, we will not include the resource layer in the layered security architecture. This is because system designers often have no control over it. Instead, we note that the data and distribution layer could implement integrity by means of detecting and possibly correcting data corruption. Parity checking and cyclic redundancy checks [10, p. 228-240] are technologies used in error detection. Hamming codes [10, p. 241-244] are able to perform error correction provided that only minimal corruption has occurred.

The temporal and the application layer still need to be considered as possible options for providing data integrity. The temporal layer will have to ensure that identification and authentication precedes authorisation and that authorisation precedes any modification of data. The discussion on data confidentiality with respect to the application layer is also relevant to data integrity. In both cases the application layer will not be considered for either data confidentiality or data integrity. An agent can not assure the integrity of a communications channel, a tasks or another agent. An attempt of an agent to verify the correctness of data, will result in the data, communication and temporal layer being accessed i.e. the application layer plays no part in information integrity but relies on the other layers to do so. The application layer will thus not be considered as an option for providing the integrity service to any of the system entities. We therefore conclude that the data, distribution and temporal layers are responsible for ensuring data integrity.

Finally, we will consider non-repudiation of data i.e. preventing any situation where the validity of data is questioned. Non-repudiation is a security service that cannot be addressed by a data or distribution layer. This is because the data and distribution layers have no knowledge of the agent or task that initiated a request for data manipulation or data retrieval.

The application layer can assist in the non-repudiation of data. This is done by involving a trusted third party (i.e. an agent) who can verify the validity of the data. Digital signatures are a prime example of a non-repudiation technology. Furthermore, we contend that a trusted third party can not only be used to verify the validity of data, but can also verify the transfer of data or the order of tasks. It could also play a role in confirming the authenticity of agent identities. The application layer will therefore be responsible for providing non-repudiation services for all four system entities.

The question of whether the order of events (i.e. the temporal layer) can provide non-repudiation services still remains. This seems to be a complex issue that is dependant on numerous other factors. Although the order of events can ensure the validity of data to some extent, there remains the issue of trusting such an ordering. If an organisation controls an information system then the order of events will also be in the control of the organisation. It is unlikely that any dispute between the organisation and another party can rely on temporal aspects for assurance. This is a trust issue. In some cases an electronic process or an information system might be accepted and trusted by general consensus. Claiming that an automatic transfer machine (ATM) has paid out an incorrect sum upon a withdrawal might prove to be difficult. This issue does not only apply to the data system entity but also to the communication, task and agent entities. We will therefore not consider the temporal layer as a possible non-repudiation service provider. The mapping of security services to the data system entity is complete. We will now consider the remaining three entities starting with a discussion of the security services required for an electronic communication.

* * *

A communication is a system entity that can best be ascribed to the distribution layer. The mapping of Figure 3 indicates that identification and authorisation is not applicable to the com-

munication entity whilst authorisation is. A communication or transmission can be initiated by agents, but they cannot assume identities and can therefore not be authenticated. Authorisation of a communication is possible without needing specific identification. An example would be the case where a web server accepts incoming connections regardless of who initiates the connection - the only criteria being the port and protocol used. In this example communication channel properties are used to authorise a transmission i.e. authentication occurs at the distribution layer.

The authorisation of a communication might also be task or agent dependant. The order of communication-related tasks could disallow or authorise the establishment of a communication channel. Authorisation can also occur at the application layer because authorisation could depend on the initiator or requestor of a communication. A suitable example is caller identification for telephones. The choice of accepting a call (i.e. authorising a communication) can be based on the identity of the caller.

The security implications of the data layer are similar for the communication and task system entities. We will therefore consider the relevance of the data layer with respect to both entities.

From our definition of the data layer we know that it involves the storage of data. It is not concerned about time or distribution related information. As already discussed, the data layer does not perform access control and can therefore not be considered for the identification, authentication and authorisation services. Also, the data layer can not be responsible for the integrity of a communication, task or agent. In our architecture communication properties and tasks dependencies will not be classified as system data. The safe storage of data cannot prevent unauthorized modification of communication or task related information. The question remains whether ownership of data has any ramifications on confidentiality or non-repudiation requirements of the communication, task or agent system entities. Does a failure to secure data at a data level compromise the confidentiality of these entities? Log files are examples of how data can leave a trail, compromising the confidentiality of system entities. Safeguarding log files is therefore essential to confidentiality. Similarly, log files can be used for non-repudiation of task, communication or agent specific information. We conclude that the data layer can provide confidentiality and non-repudiation services for the communication, task and agent system entities.

Next we need to consider additional confidentiality, integrity and non-repudiation requirements of a communication. The distribution layer needs to ensure confidentiality of a communication channel. This can be done by preventing eavesdropping. As with the case of confidentiality of data, the temporal layer could require access control to precede any privilege-dependant tasks (such as joining an existing communication).

The unauthorised modification of a communication or a communication channel can only be prevented by the distribution and temporal layers. The application and data layers have already been ruled out as possibilities. As with confidentiality, the temporal layer ensures that access control and any other task dependencies have occurred before a communication channel properties can be altered. The question remains if the integrity of a communication can be provided at a distribution layer. We argue that this is possible. Rerouting of a communications channel is a communication-related process and needs to be performed by the distribution layer.

An example of a security-enhancing technology at the distribution layer is IP Security (IPSec). IPSec is a set of protocols developed by the Internet Engineering Task Force (IETF) to support a secure exchange of packets. IPSec claims to address authentication, confidentiality and integrity at the network layer.

The application and data layers have already been identified as possible non-repudiation service providers. The temporal layer has been ruled out as a possibility. The distribution layer is concerned with the transfer of data and is unsuitable for providing non-repudiation. It will not be considered in the mapping of non-repudiation requirements to communication, task and agent

entities.

The mapping of security services to the communication system entity is complete. We will now consider the task system entity.

* * *

Tasks, and more specifically the order of tasks, are part of the temporal layer. In the temporal layer security services need to address security issues relating the order of tasks.

We will summarise previously identified mappings for the task entity. The mapping of non-repudiation services to system tasks is complete: only the application and data layers have been considered. In addition to non-repudiation, the data layer will only be considered for providing confidentiality. The application layer has been ruled out as a possibility for the confidentiality and integrity services. Only a few possible combinations still exist for a mapping to system tasks.

The distribution layer cannot identify, authenticate or authorise a task. It can, however, ensure confidentiality and integrity of tasks and task dependencies. A request for a remote task needs to rely on the distribution layer for transmitting the request.

The temporal layer cannot identify or authenticate tasks based on task dependencies. However, the order of task execution can be sufficient for authorising a task. Unauthorised disclosure or unauthorised modification of tasks and their order of execution can be prevented by the temporal layer. Finally, the application layer can identify, authenticate and authorise tasks.

These possibilities present the mappings of security services to the task system entity. Agent system entities will be considered next.

* * *

Agents perform the co-ordination processes required in the application layer. In the application layer security services need to address security issues relating to the ownership of and privileges on system data as well as the co-ordination of system tasks. The application layer provides an interface to the system.

Access control for agents can be provided by other agents or by the successful completion of one or more tasks. Neither the location nor the transmission of data can contribute to access control. Therefore the application and temporal layers will be considered for the identification, authentication and authorisation services.

Agent confidentiality is ensured by the data, temporal and distribution layers. The relevance of the data layer with respect to agent confidentiality has already been discussed. The temporal layer ensures that access control is performed before agent information is revealed. Similarly, the distribution layer can hide or reveal agent information such as the agent identity.

As in the case of confidentiality, the temporal layer can prevent the unauthorised modification of agent identities and their allocated privileges. No other layers are suitable for providing integrity services for system agents. Previous discussion on non-repudiation have already identified the application and data layers as possible non-repudiation service providers for system agents.

We therefore conclude the discussion on a mapping of the layers to security service requirements of system entities. The mapping of Figure 3 will be an important step in determining security service responsibilities of each layer. A further assessment of the mapping from security services to system entities will be done by the layered security architecture. It was the aim of this paper to lay the groundwork for future work on the layered architecture. The layered framework and the results of the mapping will be central to this process.

5 RELATED WORK

With today's increasingly complex information systems, information security has become the focus of many research papers. A need for information security clearly exists. The approaches and methodologies, however, vary considerably even for solutions to the same problem. Some approaches are invariably better than others for certain situations or problem statements.

Following is a discussion of some of the approaches that are more closely related to the layered security architecture. One example of a security model is made by the paper on distributed system security [11]. Here the authors attempt to solve the security problems inherent to distributed systems by creating an illusion of a single virtual machine. Although such a monolithic approach to information security does have merit, there are numerous advantages of using a layered approach instead.

Serpanos and Voyiatzis [9] have already researched the benefits of developing a layered approach to information security. Their solution, however, only applies to networks and more specifically addresses the OSI model. They have acknowledged the fact that numerous protocols address security at various levels of the OSI layers but fail to ensure security from a holistic point of view. A lack of methodologies that define clear and easy-to-adopt rules and steps has been noted as one of the shortcomings of current approaches to network security.

Instead they propose a Secure Network Reference Model that contains four layers. This model should complement the OSI model and provide a frame of reference for network security solutions. Their adoption of the layered approach benefits from the advantages of modularity, flexibility, ease-of-use and standardisation. These advantages will inevitably apply just as well to the layered security architecture.

Another paper that proposes a layered architecture is presented by Olivier [8]. Again the paper notes that technologies exist that address specific issues but that little has been done to structure the notion - in this case the notion to structure privacy-enhancing technologies. The paper identifies four layers and proves a strict ordering between them.

Although the papers [8, 9] bear limited relation to the idea of a security architecture for information systems, they are prime examples of how a layered approach has been ideal in the design of an architecture.

6 CONCLUSION AND FUTURE WORK

In this paper we have explored ways of modelling a system using a layered approach. The identification and classification of core system entities has resulted in a model of information system entities and their relationships to one another. This, together with the notion of system dimensions, lead to the definition of the data, distribution and the temporal layers. The application layer was added later and together with the other three layer formed the layered framework.

The layered framework indicated that it will be central to future work on a layered security architecture. The layered framework, together with the mapping of layers to the security requirements of system entities, will allow for a specification of the architecture.

Future research is required to fully explore the implications of these mappings. This could lead to the discovery of possible interdependencies of the layers. An interesting aspect will be the impact a successful implementation of a security service has on the rest of the system. Moreover, how will a failure to address a specific security service at a specific layer impact on the other layers. A discussion on these relationships will allow for a clearer understanding of where security services are required and how a failure to address this requirement will impact on the system as a whole.

References

[1] Vijayalakshmi Atluri, Soon Ae Chun, and Pietro Mazzoleni. A chinese wall security model for decentralized workflow systems. In *Proceedings of the 8th ACM conference on Computer and Communications Security*, pages 48–57. ACM Press, 2001.

[2] Patrick C. K. Hung and Kamalakar Karlapalem. A secure workflow model. In *Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003*, pages 33–41. Australian Computer Society, Inc., 2003.

[3] Iso/iec 7498, October 1999.

[4] Myong H. Kang, Joon S. Park, and Judith N. Froscher. Access control mechanisms for inter-organizational workflow. In *Proceedings of the sixth ACM symposium on Access control models and technologies*, pages 66–74. ACM Press, 2001.

[5] Michael McEvilley. The essence of information assurance and its implications for the ada community. In *Proceedings of the 2002 annual ACM SIGAda international conference on Ada*, pages 35–39. ACM Press, 2002.

[6] Chris J. Mitchell. Security techniques. *IEE*, August 1995.

[7] Martin S. Olivier. Database privacy: balancing confidentiality, integrity and availability. *SIGKDD Explor. Newsl.*, 4(2):20–27, 2002.

[8] Martin S. Olivier. A layered archtecture for privacy-enhancing technologies. In *Proceedins of the Third Annual Information Security South Africa Conference (ISSA2003)*, pages 113–126, July 2003.

[9] Dimitrios N. Serpanos and Artemios G. Voyiatzis, editors. *Secure Network Design: A Layered Approach*, July 2002.

[10] William A. Shay. *Understanding Data communications and Networks*. Brooks/Cole, 2nd edition, 1999.

[11] Wm A. Wulf, Chenxi Wang, and Darrell Kienzle. A new model of security for distributed systems. In *Proceedings of the 1996 workshop on New security paradigms*, pages 34–43. ACM Press, 1996.