International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:7, 2007

Requirements Driven Multiple View Paradigm for Developing Security Architecture

K. Chandra Sekaran

***Abstract*—**This paper describes a paradigmatic approach to develop architecture of secure systems by describing the requirements from four different points of view: that of the owner, the administrator, the user, and the network. Deriving requirements and developing architecture implies the joint elicitation and describing the problem and the structure of the solution. The view points proposed in this paper are those we consider as requirements towards their contributions as major parties in the design, implementation, usage and maintenance of secure systems. The dramatic growth of the technology of Internet and the applications deployed in World Wide Web have lead to the situation where the security has become a very important concern in the development of secure systems. Many security approaches are currently being used in organizations. In spite of the widespread use of many different security solutions, the security remains a problem. It is argued that the approach that is described in this paper for the development of secure architecture is practical by all means. The models representing these multiple points of view are termed the *requirements model (views of owner and administrator)* and the operations *model (views of user and network)*. In this paper, this multiple view paradigm is explained by first describing the specific requirements and or characteristics of secure systems (particularly in the domain of networks) and the secure architecture / system development methodology.

***Keywords*—**Multiple view paradigm, requirements model, operations model, secure system, owner, administrator, user, network.

I. Introduction

SECURITY for the early computing systems was provided by restricting the access of the physical resources. However, in this Internet era, every application is distributed in the networked environment, because the application itself being shared by multiple users. And, most of the users of these applications are the 'outsiders' (customers) of the organization which hosts this application on the web. In order to be able to share the information / application with those outside community, the restriction on physical resources access has to be removed. However, the security for the application and or information is being enabled through various security primitives such as encryption, digital signatures etc. Even though these security approaches are hard to design and implement, there are cases that these approaches are also hacked. A question that immediately comes to mind is 'why shouldn't we follow a systematic approach to provide security to the application and or information that is based on the requirements of the security policies, strategies and business processes of the organization?'. This point gives rise to the approach that is described in this paper which first identifies the requirements from four different view points in order to provide security, and then the development of a solution. This we call as '*multiple view paradigm*' on the security needs of the application.

Author is Professor in the Department of Computer Engineering, at National Institute of Technology Karnataka, Surathkal , Mangalore, India.

It is known to everyone in this present world that unbreakable security is broken all the time, often in ways its designers never imagined [1]. This suggests clearly that what we need is a way to model threats against computer and information systems. If we can understand all the different ways in which a system can be attacked, then, it is possible to design and develop counter-measures to those attacks. This paper analyses such a methodology, called *Multiple View Paradigm* for developing secure software / information systems.

In order to build a multiple view paradigm, we use a modified version of 'ConcurTaskTree'[2] notation based requirements driven task modeling approach. Task models describe how activities can be performed to reach the goals of the system when it is interacted with users. They should incorporate the requirements foreseen by all those who should be taken into consideration when designing an interactive application such as an e-commerce application, or any web based application.

II. Multiple View Paradigm

Designing secure systems with a single view point appears less promising as there are always hackers in those systems. This is more true while looking at today's larger, more distributed systems and applications over Internet. In this context, this paper utilizes a multiple view approach to the design of secure systems. The target for these systems is the environment with real-life applications running in organizations providing various critical applications.

The views of the system's security, namely, requirements model and the operations model have naturally been occurring as a result of the involvement of multiple parties in the system development and deployment process – the owner, the designer, the user, and the network with which the user accesses the application or system. These models represent the way the owner and the designer, and, the user and the network can best understand the system's security problem. This natural partitioning of the security requirements based on these two parties / views also has technical merit; it is

necessary to separate security aspects of the system into individually comprehensible packages or modules, which enables the concept which is termed as "*task tree*" in this research. This *'task tree'* is a framework that provides the operations (operations model) that are essential in order to implement the security views of the requirements model.

A secure system and the design of its architecture aims at creating a solution for the problem undertaken with the requirements identified for it; in our case, the requirements have been identified from four different view points for the secure system in an organization, and thus its architecture reflects these needs.

III. The Requirements Model

Even though researchers interested in software security have recognized the significance of the awareness factor at the implementation level (in terms of algorithms, approaches, methodologies and solutions), it is essential to note that they have failed to see its other dimensions including the one at organizational level [3]. However, recent attempts have prompted the proper identification of various kinds of requirements of security to information and or software. Security properties and behaviors of a software system are categorized into 11 classes in ISO/IEC-15408 Common Criteria.[4]. These classes are made of members, called families, based on a set of security requirements. In addition to these requirements, this research reveals the identification of the following too, to be identified as requirements for the development of complete security solutions to the software.

a) Constituent organizational units – The organizational environment may be composed of diverse interacting and collaborating constituent units that employ individual policies and governing mechanisms. In such cases it is essential to identify all such constituent units, in order to estimate the security threat from those units and develop security solutions. In other words, an organization that hosts an application or software on web, and proposes to provide security to the access of that information, must also identify and enlist the other constituent units (also called sister-concerns) of the same organization, which also would use or share the information among them. This is an essential requirement because, the security threat may emerge from an 'insider', who is working for this organizational unit but not having any access right to this information; and, may try to get the access to the secured information through another 'known' insider (colleague), who may be working in another constituent unit and having permission to access the information, which has to be monitored and prevented.
b) Constituent Systems – The infrastructure that exists in an organizational environment can have heterogeneous system components, services and applications, which may include multiple levels of databases and management systems, operating systems and networking solutions. All these systems must also be identified for the complete assessment of security requirement. It is because, through any one or more such constituent systems, it may be possible to hack the system from where the actual application is hosted, which needed to be secured from attacks.

Inherently, most of the software / information systems that require to be protected from hacking are being made available or the services of them are provided in an environment where there are multiple constituent organizational units and systems. Thus, thinking to provide security or assess the security requirement from these other constituent elements of the same organization is quite natural. The development of security in software systems must focus such a multi-constituent environment also.

As an example, Fig. 1 shows a typical requirements model, as a safeguard that can be deployed to prevent the likelihood of threats being realized or to reduce the impact of a realized threat. The (Fig. 1) requirements model caters to the need of a multi-domain or multi-constituent environment where from the software information is available to the outside world for access. That is., this model spans all of the security requirements of different domains, including communications and or network security, computer systems security, personnel security, administrative security and media security.

IV. The Operations Model

Determining the vulnerabilities and threat exposures embedded in commercial software systems and networks shall be the first step, which is very critical in practice. If we know what we have to fix and how to get the fix, a simple patch, upgrade, or configuration change could be sufficient to eliminate even the most serious vulnerability. Several security policies and models have been proposed by researchers in the past [5,6]. In order to develop a security solution to the aforesaid multi-constituent environment and its requirements model, we prefer this 'operations model'. In other words, it is not possible to trace out the place / position of threat exposure in a multi-constituent environment as it has multi-domain nature in terms of organizational units and systems. Hence we require a security model which would cater to the 'trace' of all possible security exposures in a multi-domain / multi-constituent system. The operations model performs this trace using a framework, referred to as *'task tree'* as said in section 2.

The '*task tree'* has two levels: the top level correlates to the requirements model, and the lower level incorporates the 'trace' extractions based on the requirements model that is described in the previous section. The top / upper level is to specify the essential security mechanisms or solutions in order to meet the security requirements identified in the requirements model. Each of the security mechanisms listed in the top level is being broken down (in the bottom or lower level) into a set of sub-mechanisms, which are referred to as 'tasks'. Thus the structure of the security model that is prescribed here looks like a 'task tree'.

Fig. 2 shows top level security mechanisms that correspond to the requirements model shown in Fig. 1. These security mechanisms may be required in typical application software

such as an e-commerce software, that is serving many clients from a multi-constituent organization. In this example / figure, there is a mapping between the requirements identified in the requirements model and the mechanisms supposed to be used as solutions for those requirements.

Fig. 3 shows a typical bottom level '*sub-mechanisms'* that are required in case the '*mechanism*' is "access control". A close look at these sub-mechanisms reveal that these have been captured in order to provide a complete security solution to each top level mechanism by reconsidering it – how a mechanism (security solution) is to be viewed, defined and represented in a multi-constituent / domain environment, and how to capture the processes or tasks that need to be in the 'task tree', in order to provide a complete secure solution to the mechanism in hand. Relevant issues included a need to know how to trace the threat in the constituent units and systems of the organization, assurance of the continued integrity at sub-systems level and the level of protection or guard for the messages for the chosen mechanism.

Thus, this task tree based operations model enables the complete security solution that must be based on the insertion / inclusion of different units and systems of the environment / organization, through their critical points and interfaces. It is true that these included security measures are, in addition to the normal security solutions that are visualized for the hosted software / information system.

The security solution thus provided through the multiple view paradigm described above, enables the solution's space to be split into the smallest acceptable secure functional elements or solutions at all possible levels, with all data / information access through them. This security solution only could be the tightest and operationally acceptable methodology that is required to be developed and incorporated in the software systems of today's world.

V. An Example

One of the problems with gauging the true business impact of security breaches is that many companies are understandably reluctant to publicly admit that they have suffered significant losses due to the failed security. It is not merely a network problem or an information technology problem. The development and implementation of a sound security policy must start with strategic business assessment from the owners of that business, then its design and development issues will have to be thoroughly studied and implemented by the designer; the requirements at the genuine users and the threats from the bogus users also must be taken care; the problems associated at the network – either at the protocols of networks or applications, and the users, thus must be the totality of the concerns that are to be considered before the security is deployed in that company. These views, which have been discussed in the above sections are being studied and experimented for this research purpose, with many organizations on a trial basis. A typical example of an e-business company is presented in this section, in brief.

Table I gives the analysis (requirements model) carried out for a business house that has its applications in the web. For this e-business to be a highly secured one, there will be different requirements from the four parties of the system. These have been illustrated in Table I. For example, the transaction processing requirements for the owner of the application is making the system 'fully alert'; that of the designer is to enable all the possible methodologies using suitable algorithms or approaches; that of the user is to exploit the system for its fullest potential; and, for the network side, it is to monitor all the transaction based messages and provide security based information access through network. The operations model has also been developed using task tree representation. This task tree gives out the details of lower level security measures that need to be incorporated in the system.

VI. Conclusion

The growing dependence of the society on computers and software systems require that serious efforts are to be put by the research community to meet the growing IT security. Due to interdependence and interconnections of constituent units and systems in any organization, the usual parochial and short term focus of IT security is no longer adequate. We must consider the possible impacts of security threats from all these sub-systems and shall provide a sustainable security model as long term solution. An attempt towards this task is the multiple view paradigm that is presented in this paper. Without careful attention to these issues discussed in this model, it is easy to understand that, a complete security solution cannot be developed.

References

[1] Anderson, R, Why Cryptosystems Fail, Communications of the ACM, No.37, pp.32-40, 1994.
[2] F. Paterno, Model based Design and Evaluation of Interactive Applications, Springer-Verlag, 1999, ISBN 1-85233-155-0.
[3] Theus V., and Ray H., Intrusion Detection Techniques and approaches, Computer Communications, Elsevier, No.25, pp.1356-1365, 2002.
[4] ISO/IEC-15408, Common Criteria for Information Technology Security Evaluation, v2.0, National Institute of Standards & Technology, Washington, DC, June 1999.
[5] Anthony H., and Roderick C., Correctness by Construction: Developing a Commercial Secure System, IEEE Software, pp.18-25, Jan-Feb 2002.
[6] Khaled M.K., and Jun Han, Composing Security-aware Software, IEEE Software, pp.34-41, Jan-Feb 2002.

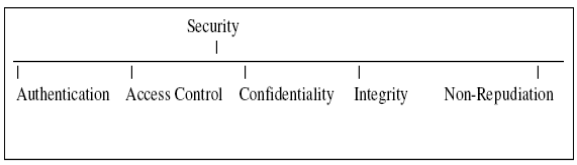

Fig. 1 A typical Requirements Model Task Tree

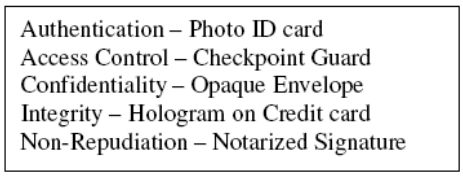

Fig. 2 Security Mechanisms – Top Level of Operations Model

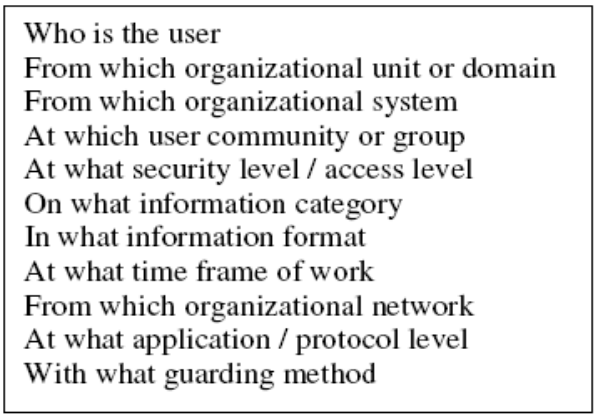

Fig. 3 Sub-mechanisms of Access Control-the Tasks- the bottom level of the Task Tree

TABLE I
VIEWS AND REQUIREMENTS ANALYSIS FOR AN E-COMMERCE APPLICATION

Views / Require-ments	Owner	Designer	User	Network
Business-related Security issues	List Security holes	List all threats & vulnerabilities	Methods to access	Protocols to support the access
Transaction Processing	Full-Alert Policy	Algorithms & Methodology for complete secure system	Full usage	Monitoring messages & secure them