ABSTRACT

This paper provides a pragmatic overview of design requirements that are dictated in official documents regarding the security in standardized Open Computer Systems. The main focus is on publicized military and commercial security requirements. However, some commonly discussed security models meeting these requirements will be placed into context.

1. INTRODUCTION

In the development of computers the trend is moving towards Open Systems based upon well understood operating systems (i.e. UNIX\textsuperscript{TM}, DOS\textsuperscript{TM}, etc.) and well defined communication architectures (OSI, TCP-IP, etc.). The purpose of Open Systems is openness and user friendliness something that is contradictory with a secure system. However, there is a need for computer security because of the importance that computers are playing in our daily life. Computers are the backbone of the financial world, governments, educational systems, scientific research and military systems. The ease of access to computers and their connectability provides added dimensions to threats and vulnerability of computer systems.

The era of distributed computing provides a new computing environment as a set of systems connected together via a local area network (LAN) or a wide area network (WAN). The systems are workstations, network terminals, tightly coupled systems, etc. The openness of these systems is provided by their adherence to standard communication protocols and use of programmatic interfaces and other mechanisms that allow the systems to share logical or physical devices.

The era of distributed computing provides a new computing environment that provides greater flexibility through openness via the many available applications and the connectability using the new communications technologies being developed. In reality there are no physical walls surrounding the information in an Open Computer System. The distributed computer environment is a veritable unlimited collection of interconnected machines in multiple domains serving a multitude of different applications. Current computer technology evolves around the traditional mainframe computer being replaced by combinations of tightly coupled systems and loosely coupled or networked systems that provide greater flexibility than ever before achieved.

2. SECURITY REQUIREMENTS

A quantitative measure of the cost of not having a secure system can only be provided when the monetary loss is directly measurable. In many cases a computer system is a Pandora's box of intellectual property, proprietary data, or secret information that has little direct value until combined into a whole. Our judicial system is not always ready for computer crimes that are performed due to the lack of a secure system. For example, is it a crime to peek into someone's car or house? Is it a crime if someone performs passive monitoring of a computer link, or lists the files or the contents of the files of a computer system that they are logged into? Is it a crime if someone accidentally introduces a computer virus into a network that affects hundreds or even thousands of computers? The answer depends upon the contents of the information that is disclosed, the secrecy level it was assigned, and the offenders' intentions.

Recently several incidents have received much attention in the media. A few years ago a worm planted by a student infected hundreds of university and military computers. This would probably not have happened to any of the affected systems if they had been securely designed and administered. The cost incurred included the downtime of each of the infected computers, the work that was not performed during this time, and the time that the system administrators used to restore these systems with backups, that had been taken prior to the infection.
In addition the cost incurred included the time it took each individual user to restore their work from the time of the last backup to the time of the worm or infection (the original integrity of the computer files and programs).

The film "Wargames" introduces another, hopefully, fictitious situation where the user (a young boy) gains unauthorized access to a system and, unintentionally, almost initiates the third world war by triggering an unmanned nuclear weapon system. The value of this type of computer crime is quite obvious to anyone.

The above examples together with many other disclosed and undisclosed incidents are proof for the argument that there is a cost related to not having a secure system, and that there is a need for this security requirement in Open Systems. One way of expressing the cost of a lack of security is

\[ \text{COST} = U \times V \times D \]

where

- \( U \) = Number of affected users
- \( V \) = Mean hourly value of the users services/work
- \( D \) = Mean downtime and time to restore the system

Many people claim that the threat to security is not the system itself (a computer performs the tasks for which it is designed and programmed), but the user of the system, the computer's design, and the method of design verification. Theodore M. P. Lee [9] uses probability theory to give an informal view of statistical models of Trusted Computer Systems versus people. The computer provides the right level of security in a given environment when all combinations of the following three risks are low enough:

1. The risk of error in determining that the system is secure.
2. The risk of error in the personnel security practices.
3. The risk that sensitive information will be misused.

Of the three factors, error determination is the sole factor under control of the computer vendor.

The military uses the clearance process while the commercial industry often uses similar levels of "need to know" or "trusted to know": "A person is to be granted a clearance of a given level if the expected distribution of damage, computed over all people granted that clearance, during their tenure, assuming they are granted access to information of that level, falls below some threshold." [9]

Assuming a quantitative measure of \( x \) units for the damage using the military classifications TS (Top Secret), S (Secret), and C (Classified), the probability density function \( D(x) \) for the damage is shown in the figure "Distribution Function of Harm." [9:14]

UNIX\textsuperscript{TM} places labels on programs and files utilizing the concept of superuser, etc. ... The above model could also be used to estimate the consequences of a malicious program running at different classification levels in a UNIX\textsuperscript{TM} computer.

Lee also deduces a model for the failure distribution for a system using the three highest security levels of the military TCB (Trusted Computer Bases) evaluation class (B2 - A1). [9:17] The probability function \( Re(t) \) infers the probability that a TCB of Class \( e \) has a security flaw or the possibility of introducing a flaw in its mandatory access control. Table 1 gives explanation of the criteria for the Classes.

Based upon the above probability density functions and examples there is definite value in designing secure systems as well as providing policies for the people using the system. Since the system is not more secure than the design and the verified design of the system, policies alone are not enough to provide a Secure Open System. The design of the computer is inherently connected to the requirements of the user. User requirements are explicitly documented in literature from the Department of Defense [3], ISO [7], ECMA [4], etc. The following sections give a plausible overview of the requirements for Secure Open Systems.

2.1 TRUSTED COMPUTER SYSTEM EVALUATION CRITERIA

In 1985 the Department of Defense (DOD) published the "Trusted Computer System Evaluation Criteria" [3] (also known as the "Orange Book" due to it's bright orange color). The Orange book allocates a division of classes for the security of a system ranging from D (Minimal protection) to A (Verified protection). The orange book is not an implementation guide nor is it an architectural document. It simply gives a set of requirements for the Trusted Computer Bases (TCBs) that are used by the DOD. However, since this document was the first document from a major institution to specify requirements of this type it has become a major document in the evolution of any Secure Open System. The terminology in this document has become common design terminology for computer and for software vendors, as well as for security requirements from other Government institutions.
The different levels of classification in Table 1 provide a vehicle for increased levels of security. In the discussion of security levels, the orange book uses four building blocks:

- Security Policy
- Accountability
- Assurance
- Documentation

The Security Policy is the formal set of laws, regulations, rules and practices that any organization applies when managing, protecting, and distributing sensitive information. The Security Policy will inherently provide a policy for the use of a computer in this organization. Dependent upon the usage of computers in this organization, tailored policies for the use of these in the organization exists. The orange book provides a set of requirements linked with a class of security that aids in the definition and interpretation of the security policy. However, any organization's security threat is a function of the organization's goals (i.e., credit bureau vs. research) and its exposure to these threats.

The accountability need specifies the need for identification of the subjects utilizing the system. The identification also specifies the need for audit information providing a capability to identify a responsible party or to identify the security relevant events, (i.e., change to superuser in UNIX™, etc.) The authorization and audit data must be protected from unauthorized modifications and destruction.

The assurance requirement objectives ensures that hardware, firmware, and software will be evaluated and verified to be secure. The secure system being a system that adheres to a security policy and an accountability measure. There is a need to document the functionality of the mechanisms providing the trust. (Later we will see that a security model is an important mechanism in this verification and documentation.)

To comply with the criteria there is also a need for security documentation supporting the system. This need increases when targeting for higher levels of security. The documentation covers the areas of usage, and administration of the system through design and test documentation. At level B1 and higher the requirement for an informal model of the requirements is specified, and at the highest level A1 formal security specifications and verification plans are required. These requirements are summarized in Table 2. [3:109]

The orange book provided a formal requirements for a single system that is not networked. However, in the world of Open Systems, the Network is a key component. The Department of Defense has prepared additional documents [12] to provide guidelines in these environments; however, most work in this area is found in the frameworks of ISO [7] and ECMA [4]. Also, recently (Draft May 90) the Information Technology Security Evaluation Criteria (ITSEC) specifies the 'Harmonized Criteria of France, Germany, the Netherlands, and the United Kingdom [6]. The ITSEC is first formal evaluation definition from Europe threaded with the 'Orange book' and the TNI.

### 2.2 NETWORKING REQUIREMENTS

The International Organization for Standardization (ISO) maybe the most renowned standardization body for communication architecture standards. ISO is a voluntary organization with members from national standardization bodies coming from its respective member countries. ISO receives input from working members including organizations such as ANSI (the American National Standards Institute), ECMA (the European Computer Manufacturing Organization), IEEE (Institute of Electrical and Electronics Engineers), etc. Through ISO the work of a security standard for the Open Systems Interconnect (OSI) model is evolving. The ISO/TC97/SC21/WG1 and the ISO Special Interest Group on Security (ISO SIG-SEC) is providing corner stones for the security architecture found in OSI. The security framework [7] defines a set of OSI security categories and services as well as it suggests in
2.3 SUMMARY OF THE REQUIREMENTS

The above documents provide a wide range of specifications for a secure computer system. Three major computer security policies that can be extracted are:

- **CONFIDENTIALITY**: The protection of information from unauthorized disclosure.
- **INTEGRITY**: The protection of information from unauthorized alteration.
- **DEFIANCE**: The protection against service attacks.

The confidentiality requirement comes from the military requirement for multilevel security. The need for confidentiality is a requirement for protection of information from unauthorized disclosure. In businesses other than the military this is the protection of an individual's privacy by not providing access to confidential information such as medical records, credit records, etc.

The integrity requirement is the protection from unauthorized modification of information. This has been a primary concern of the commercial industry for quite some time. While the formulation of the confidentiality requirement is well defined in the multilevel security policy, integrity lacks a similarly well defined policy.

The defiance question is the protection of a system from service attacks. The requirements are straightforward, however, they are not as easily understood or implemented.

The next section talks about some models that are used in the understanding of the secrecy and confidentiality need as well as the integrity need. The defiance need is not discussed here since this is indirectly inherent to confidentiality and integrity. The denial of service can be a matter of hindering unauthorized access to a system which is covered by access control models.

### 3. SECURITY MODELS

The security model is a concept that is used for further defining requirements in an abstract fashion. The security model is needed when designing a secure system, as it is a requirement when designing and verifying governmental computer systems. In the U.S. Trusted Computer System Evaluation Criteria an informal model of the computer security policy must be provided when submitting a system for an evaluation of rating of B1, and a formal model is required for B2 and higher. Also, "The German Criteria for the Evaluation of Trustworthiness of Information Technology Systems" requires a formal model at the Q3 level (comparable to B3) [13]. Since the framework of the Orange book is used in most evolving standards for security it is inherent that the concept of security models will evolve. The security model provides an abstract expression of the requirements and provides a more precise specification of the security requirements defining the properties of the system without defining the system's functions. The model may be formally written using mathematics and set theory, a natural language, or an algorithmic computer language. Traditional security models are commonly modeled with a mathematical language. The traditional models furthermore are built for a set of subjects or agents operating on a set of objects using a well defined set of rules.

In the world of security models there are some common denominators that should be discussed when analyzing existing models. First, the security models demonstrate some sort of access control (preferably mandatory access controls). The access control is in a limited scope the access seen when subjects operate on an object or a set of objects, the traditional view of access control. On the other hand, when defining the requirements in the Distributed Computer Environment the scope possibly be broader. In a computer network as well as in a TCB there is a second dimension. There is a need for information flow control in the access control mechanism. The Information flow control provides a concept for understanding the integrity in a system and the analysis of covert channels.

Before discussing the security models further and their fulfillment of the requirements that were abstracted in the previous section, I will elaborate the purpose of a security model. Morrie Gasser identifies the following properties of a security model: [5,130]

1. It is precise and unambiguous.
2. It is simple and abstract, and therefore easy to comprehend.
3. It is generic. It deals with security properties only and does not unduly constrain the functions of the system and its implementation.
4. It is an obvious representation of the security policy.
The first criteria is the primary reason why a security model is usually specified in a mathematical language which is the most precise and unambiguous language that can be used. The second criteria calls for abstract models where the properties, but not the functions of the system are modeled. The purpose of the model is not to provide a specification but to provide a close to authentic abstraction of the real world. By not dealing with the functions of the system the model will not constrain the design; hence the model should be useful in all cases where it represents a security policy. The model will provide a tool for understanding and interpreting the security requirements and the implementation of these.

In the analysis of a security model it is important to consider the functional building blocks of a model. The concept of a state transition is commonly used in the modeling of computers, and it is a fundamental building block of many models. The concept of a security state model is that the transformation from one state to another follows well defined rules. Also, each state can be represented with a fixed number of sets of variables. As shown in tables 4 and 5, the states can be represented in an access matrix where the rows or columns represent the subjects while the counterpart of the matrix's columns or rows represents the manipulated objects.

To further understand the state machine model consider the building blocks of a model:

1. **STATE VARIABLES**. The state variables are extracted from the security policy and provide a mathematical abstraction of this policy (i.e. Subjects could be the set of all subjects $S = \{x\}$, while objects could be the set of all the objects $O = \{y\}$).

2. **SECURE STATE**. The definition of a secure state is obtained by taking properties of the policy in their mathematical translation and constructing an invariant. Or, equivalently taking the properties of the policy and constructing a table or array of these states.

3. **TRANSITION RULES**. The transition function provides the rules that can change the state of a combined subject and object into another state. These rules are mathematical or computerized rules.

Initially the need for linking the computer security policy together with an informal or formal model was identified. The formal model being a mathematical or algorithmic model of the security policy versus the informal model being a less restrictive representation of the security model. Due to the requirement of a model one needs to understand the commonly used models that currently exist. Previously, I argued that there is a need for understanding not only the confidentiality policy but also the integrity policy. The difference being that the confidentiality policy aims towards a "need to know policy" versus the integrity policy aims towards a "need to change policy". During the past twenty years important work has been done in the development of computer security policy models. The most visible work has probably been performed by Bell and LaPadula [10], Clark Wilson [2], and Cohen [11].

**Bell-LaPadula** is a lattice model utilizing the state transition concept to describe subjects and objects transition in a computer policy adhering to the confidentiality policy. The model evolves around sets of subjects and objects. A set of transition rules are described and proven for the transition from one secure state to another. The subjects are specified with a clearance level while the objects are specified with some classification. The clearance and classification scheme is specified in accordance to a lattice where each pair of subject and objects have a greatest lower bound and a least upper bound.

**Clark Wilson** is an information flow model utilizing transaction rules to describe the flow of data, separation of duties and logging of transactions for the purpose of auditing in a computer policy adhering to the integrity policy. There are two data elements in the Clark Wilson model. The Constrained Data Items (CDIs) are labeled since they are used by the integrity model. The Unconstrained Data Items (UDIs) are not identified and labeled. The rules of the model are defined into Integration Verification Procedures (IVPs) and Transformation Procedures (TPs). The IVP being the integrity verification tool, and the TP being the well defined transaction.

**Cohens POset** is a simple information flow model combining confidentiality and integrity lattices adhering to both the confidentiality and integrity policy. The model utilizes partially ordered information sets (POsets) in combination with information domains. The transitive POset modeled is "Flow Control POset" (FCP). The POset is modeled to flow in an information domain which is the entity that information flows to and from. The mechanism of a "Time Flow Configuration" (TFC) is used to maintain the information flow set. These three models are summarized in Table 6.

<table>
<thead>
<tr>
<th>Subject/Object Access Matrix Model</th>
<th>Subject/Object Access Matrix Model</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LEGEND:</strong></td>
<td><strong>LEGEND:</strong></td>
</tr>
<tr>
<td>R -&gt; Read</td>
<td>S -&gt; Send</td>
</tr>
<tr>
<td>W -&gt; Write</td>
<td>R -&gt; Receive</td>
</tr>
<tr>
<td>X -&gt; Execute</td>
<td></td>
</tr>
<tr>
<td><strong>TABLE 4.</strong> Subject/Object Access Matrix Model</td>
<td><strong>TABLE 5.</strong> Subject Access Matrix Model</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MODEL</th>
<th>POLICY</th>
<th>TYPE/PROPERTIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bell-LaPadula</td>
<td>Confidentiality</td>
<td>State Transition</td>
</tr>
<tr>
<td>Clark Wilson</td>
<td>Integrity</td>
<td>Information Flow</td>
</tr>
<tr>
<td>Cohens POset</td>
<td>Confidentiality/Integrity</td>
<td>Information Flows</td>
</tr>
</tbody>
</table>

**TABLE 6. Security Model Summary**
4. CONCLUSIONS AND RECOMMENDATIONS

The Information age is moving towards new dimensions where the functions traditionally provided by mainframe computers are being distributed across a network into an environment consisting of Open Systems. The traditional view of security in a computer environment evolved around two distinct policies, "the need to know" and the "need to modify." It is important to realize that the paradox is here. Computer security policies that were evolving towards different goals should now merge towards a common target. The target is a mixture of the Integrity and the Confidentiality policy. The discussion should not focus on the computer policies and the models that support these. The decision for any development organization will be to determine if the integrity requirement should be modeled separately from the confidentiality requirement, or if they should be combined into one model. The design will require mechanisms to meet both these policies.

The Bell-LaPadula model appears to meet only the confidentiality policy, the Clark Wilson model meets the integrity policy and Cohen's POset model may meet the combined requirements. Whatever the designer and the developer of a secure computer system does, it is important to stay focused. There is much more work than just the modeling to be done. The model only expresses an abstract view of the policy and is far from being a specification and/or implementation of the security in a computer system. However, by following the rules and recommendations that can be deduced from these models, one will find some useful mechanisms for understanding the prevention of unauthorized information disclosure and modification, for exposing covert channels and for fighting Trojan horses, computer viruses, malicious programs and other threats. A further discussion of the design and implementation of security in distributed computer systems can be found in Kolstad [8].

5. REFERENCES


