An Outline of a Taxonomy of Computer Security Research and Development

Catherine Meadows
Naval Research Laboratory
Code 5543
Washington, D.C. 20375

1 Introduction

Research in computer security in the last decade has in general been concentrated in a few areas: highly trustworthy systems that protect data at different security levels. As people are starting to realize how the computer security problem can affect their lives in ways that may have little or nothing to do with these narrowly defined problems, research has been starting to broaden. However, this broadening of interest has been sporadic, and it is difficult to get an overall picture of where research is heading. In this paper we present the outline of a taxonomy which will allow us to place existing and possibly future computer security research in context. This taxonomy is intended to be a growing entity; as new areas of research open up we can extend the taxonomy to include them. The taxonomy also allows us to identify areas of research that are still relatively unexplored.

2 Discussion of the Taxonomy

For the purpose of the taxonomy, we define computer security to include any means for ensuring that a computer-based system performs a function in the face of an intruder or intruders who are actively trying to prevent it from doing so. In our taxonomy we divide computer security research into five orthogonal areas. One can identify subareas, not only by picking subareas of each area, but by picking subareas of one or more areas and combining them. Not all subareas identified this way will be useful, but in many cases this technique may help us discover a potentially fruitful subarea that may have been neglected. It may also help give us a better idea as to how a subarea fits into the general scheme of things.

The five areas we define are systems, policies, techniques, assurance, and interactions with other requirements. In the systems area we identify the kinds of systems we are trying to secure. In the policies area we identify the policy that a system must enforce. In the techniques area we identify the kinds of techniques (e.g. encryption) that are available to us. In the assurance area we identify the various assurance techniques that we can use to assure that a system enforces its policies. Finally, in the interactions area we identify the various tradeoffs between security and other desirable properties of a system.

2.1 Systems

We divide Systems into two subareas: components and composed systems. We define a component to be a system that is designed and built all in one piece. An example of a component would be an encryption device, a secure operating system built to Orange Book standards, or an application that is intended to run on a secure operation system. A composed system is one that is composed out of one or more secure systems that were designed and built separately. An example would be a network consisting of nodes trusted at various levels, or a system consisting of a secure application running on top of a secure operating system. We divide composed systems into two types: flat and hierarchical. A flat system is one in each component enforces its own security policy separately, such as a network. An hierarchical system is one in which one component relies on another to enforce part of its security policy, such as an application running on top of a secure operating system. A system may have both flat and hierarchical aspects.

2.2 Policies

At this point, we make a crude division of policies into three areas. This first is protection of exclusivity. This includes all policies in which the system is expected to protect various entities from unauthorized use, and includes secrecy (prevention of unauthorized

1993 ACM 0-89791-635-2

knowledge of data) as a subcase. The second is protection against unauthorized modification (integrity). The third is protection against denial of service. Other possible ways of breaking down policies would include the means by which the decision to grant or withhold authorization is made.

2.3 Techniques

In this area we include the various techniques that can be used for enforcing security policies. At this point we have decided to break techniques down into three subareas: techniques for enforcing security within a system, techniques for protecting a system against intrusion from outside, and techniques for enforcing security between systems. Most security techniques fall naturally into one of these categories, although there is occasionally some overlap.

Under techniques for enforcing security within a system we include the various techniques for enforcing access control, such as reference monitors, and techniques for guarding against inference of sensitive information, such as covert channel analysis and the various inference prevention mechanisms that have been proposed for secure databases.

We divide techniques for protecting a system against intrusion from outside into two subareas. These are techniques for detecting intrusion and other system anomalies and techniques for authentication. Under intrusion detection we include the various tools under development that are used to detect intrusion of a hostile user, as well as virus detectors that can be used to detect intrusion by hostile software. Under authentication we include passwords, biometric authentication, and cryptographic authentication.

Under techniques for enforcing security between systems we include all techniques that can be used to help systems communicate securely in a hostile environment. These include key distribution and intersystem authentication protocols, as well as secure communication devices.

2.4 Assurance

In order for a secure system to be usable, it must not only be secure, but the user must have a high degree of confidence in its security. Such confidence is difficult to obtain, since the user must trust the system to behave correctly, not only under normal operating conditions, but in the presence of individuals or programs that are actively seeking to subvert its goals.

There is some overlap between assurance techniques and security techniques. A technique that can be used to provide more assurance that a system is secure can also be used to uncover security flaws that may not have been found without the technique. Thus we can consider something like covert channel analysis as both a security technique and a verification technique.

We divide assurance techniques into four areas: formal methods, semi-formal methods, testing, evaluation. We do not intend this to be a complete description of all possible assurance techniques, but at this point such a division appears to cover the techniques available.

Under formal methods we include techniques that require a formal mathematical model of the property of interest, a formal specification of the system in terms of the model, and mathematical techniques for proving that the specification satisfies its requirements. Proofs may be done by hand or by machine. Semi-formal methods include those that allow a designer to perform part of the task in a formal manner, but leave part to be done informally. An example of a semi-formal method would be covert channel analsysis tools, which identify potential covert channels and then require the user to figure out which are channels can actually be exploited.

Testing for computer security is in general as not as well understood as the formal and semi-formal methods. Most testing of secure computer systems relies on informal "tiger team" approaches in which testers attempt to break into a computer system. This is because most conventional testing techniques require some assumptions to be made about "normal" behavior of the system, while security deals with worst-case scenarios. As testing techniques improve, however, we may find approaches that are more adaptable to security.

Finally, we include evaluation, which we consider to be a "meta" assurance technique. Evaluation is the task of determining what are the appropriate assurance techniques that should be applied to a system and in what proportion, and of determining how one decides whether or not they have been applied appropriateley. Evaluation asks the question: "What do we have to do to a system before we believe that it is secure?"

2.5 Interaction with other System Requirements

Under interaction with other system requirements we include various desirable features of a system that are usually at odds with security, and techniques for determining when we should emphasize one feature over another, as well as techniques for achieving a fea-

ture without sacrificing the other, if possible. This is an area that has been neglected until recently. However, as we move beyond protection of classified information in operating systems to other security problems, we face these issues more and more. Thus we see this as a growing area.

One of the first requirements that is considered to cause problems when security is brought up is that of performance. Just about any security feature has the effect of slowing a system down. What do we do when a system must perform a function in a given amount of time? Are there ways of meeting security requirements without sacrificing performance? Are there ways of relaxing security requirements so that performance goals can be met so that an acceptable degree of security is still achieved?

Another requirement that has often cropped up in the database security area is that of consistency. A system that manages data should give a consistent view of the world. How do we do this when some data is not available to some users?

Dependability is another system requirement of concern. Some security techniques require that a system cease operation in part or in whole when a violation is detected. How can we employ such techniques and still have assurance that the system will operate correctly when needed?

Finally, we need to deal with human factors. Secure systems are notoriously "user-hostile". This can not only affect the usability of a system, but can also affect its security, since users may attempt to "work around" the security features in order make their lives easier. Thus it is necessary to make sure that security features do not affect human factors too adversely.

3 Conclusion

In this paper we set forth an outline of a taxonomy of computer security research. The purpose of this taxonomy is, not only to provide a picture of where research stands at the moment, but to identify neglected areas of research and possible new directions. One way of using the taxonomy to do this is to pick topics from two or more of the five orthogonal areas and ask if there is any research being done on this combination of topics, and if not, should there be. For example, consider the combination of intrusion detection programs and assurance. Little work has been done on providing assurance that such programs do their job in identifying intruders. What would be the best way of doing so? Formal methods, at least by themselves, do not seem appropriate here, since it is difficult to provide a formal description of an intruder. Thus some kind of testing would probably be appropriate. But what kind, and how should it be applied?

To give another example, consider the combination of policy and interaction with human factors. The encoding of a paper policy in a computer system can have unexpected effects. Flexibility may be lost, and informal interprations of the policy may not be captured. Thus it is necessary to develop ways of capturing the way the policy is enforced and translating that into a language acceptable by the system.

These are only a few of the issues that we can raise by examining the taxonomy. It is hoped that as this taxonomy matures, we can use it to identify other such issues in computer security research.

TAXONOMY

1. Systems
    - 1.1 Components
    - 1.2 Composed Systems
        - 1.2.1 Hierarchical Composition
        - 1.2.2 Flat Composition
2. Policies
    - 2.1 Exclusivity
    - 2.2 Integrity
    - 2.3 Assured Service
3. Techniques
    - 3.1 Within System
        - 3.1.1 Access Control
        - 3.1.2 Inference Prevention
    - 3.2 Without System
        - 3.2.1 Intrusion Detection
        - 3.2.2 Authentication
    - 3.3 Between Systems
        - 3.3.1 Secure Communication Between Systems
4. Assurance
    - 4.1 Formal and Semi-Formal Methods
    - 4.2 Testing
    - 4.3 Evaluation
5. Interactions with other System Requirement
    - 5.1 Interactions with Performance
    - 5.2 Interactions with Consistency
    - 5.3 Interactions with Dependability
    - 5.4 Interactions with Human Factors