# Managing Complexity in Secure Networks

David Bailey
Galaxy Computer Services
3419 La Sala del Oeste N.E.
Albuquerque, New Mexico 87111

## Abstract

The "Security Island" physical security paradigm, on which we base our concepts of protecting computer systems, derives from notions of centralized control and isolation. Implicit in this view is the need for "global understanding" of the system being protected: it must, in principal, be possible for a single person to know about all of the data paths and security controls within the system. Otherwise, it is not possible to analyze adequately the protection afforded by the system. As a system grows in size and complexity, maintaining global understanding becomes increasingly difficult, and ultimately it is impossible. We describe two alternative paradigms—"Secure Telephone" and "VIP Protection"—that may be able to survive the complexity threshold at which the security island paradigm collapses.

## 1 Do We Need a New Paradigm?

A trusted network poses many difficult security problems. The hardest of these seem to stem directly from the complexity of the network. The trouble is that our ideas of how to protect computer systems and networks* are derived directly from our past experiences with physical security systems. The physical security paradigm, which we might call "security islands," is based on isolation and hierarchical control. As network size and complexity grow, hierarchical control becomes increasingly difficult to manage because it is no longer possible to "understand" what the system really does, it is no longer possible to analyze the implications of change, and it is no longer possible to determine whether development is being performed correctly.

*For the purposes of this paper, any distinction between the ideas of "computer system" and "computer network" are irrelevant. In the remainder of the paper we will use the terms "system" and "network" interchangeably.

In this paper we explore the *security island paradigm* and where it leads. We then present two "new" paradigms (neither of which is at all new), one based on rejecting hierarchical control and the other based on rejecting isolation.

## 2 Security Islands

The security island is commonly used for designing physical security systems for fixed plant sites. A sensitive operation is located in a building somewhere. To protect it, we establish a security perimeter, build a fence, put guards at the gates, and control who can enter and exit the perimeter.

Variants on this scheme are used when the requirements vary. Sometimes the perimeter is not made obvious with a fence and the guards are less visible. Sometimes the perimeter is arranged so that the public can enter part of the facility. Sometimes it is necessary to segregate part of the site population from other parts. The basic design, however, remains the same. One person is in overall control of facility security. That person knows what assets are being protected and how the protection is being accomplished. He or she is in a position to analyze the effects of changes and to establish whether security changes have been made correctly when the operation changes.

When it first became necessary to protect computers, the job was assigned to the physical security manager for whom the techniques to be applied were obvious. In the early days, when the system was a single machine that ate cards and produced listings or tapes, the techniques applied by the physical security manager worked very well. Over the first decade of computer security experience a large body of knowledge was accumulated and became well entrenched. The techniques that developed had only one minor problem-they were inadequate for the remote access time-sharing systems that were coming into vogue at the end of the period.

Now, another fifteen years later, we are left with a legacy of physical security attitudes and practices that have been gradually bent and stretched to accommodate new technology until they can be stretched no more. Computer networks have grown large despite the active resistance of security managers. During this growth, we have retained the attitude that a network is a collection of separately accredited components that have to share data.

Pairwise it is not too difficult to decide what controls are needed, but to make global statements (i.e. across the entire network) about the protection of data-the only kind of statements that the security manager is really interested in-it is necessary to compute the transitive closure of every allowed flow, and the possible paths may be hundreds or thousands of segments long. This cannot be done without computational assistance (i.e. one's intuition no longer works), and the assistance must be obtained from the very systems that the security manager is loath to trust. It is not difficult to see why the security of networks is thought to be a hard problem. It is also not difficult to see why security managers typically believe that the computing people are out of control.

We should not be terribly surprised at these results. The separate accreditation of network components is the network analog of creating a new security area at the protected site (another security island). Adding several thousand new physical security areas at the site would make the site unmanageable-and it doesn't work any better with computers. It is attractive to accredit separately because the other alternative-re-evaluating the entire network every time a new node is added-is an obvious failure. Unfortunately, it seems necessary for a single person to understand the operation of the entire network in order to understand its security properties For many existing networks, it is already impossible for a single person to understand the entire network operation. In any case, it is clear that any network can grow to this state.

As a result, any security policy or management scheme based on "global understanding" of the network is bankrupt. Such a policy, while adequate for a small network will ultimately be insufficient. The result of long term development may be failure of the network to provide adequate service to its customers because its security managers or accreditors are conservative.

Development can also be continued beyond the point where it is adequately secure because the developers are persuasive. Most likely, both results will occur.

The natural human response to this situation is to modularize. We want to break the network into independent pieces that can be understood separately and whose interactions can be analyzed pairwise. Within the Department of Energy community, we have used a concept called partitioning to provide the needed modularity. A **partition**[†] is a division of the components of a network for access control purposes such that no component is in more than one partition. The systems within a partition must have the same protection requirements and the users satisfy a common clearance requirement. Relatively free exchange of information is allowed within a partition. For the purposes of security analysis, the systems within a partition are all equivalent. One can think of them as a single system, even though they may not offer this functionality to their users.

Partitioning, applied to collections of-operating systems, was an adequate paradigm for the 70s and the early 80s when interactions were between separate systems and were fairly simple. It put off the inevitable for another decade. It is much less adequate today when intersystem interactions are more numerous, occur at a lower level of detail in the network, and are less obvious then the user. Network File Systems (NFS) and diskless workstations, where the use of the network to obtain a requested piece of information is completely hidden from the user, are good examples of this new style of interaction.

The new types of system interaction demand a new paradigm for modularizing security while, at the same time, making modularization more important. Since the interactions are occurring (that is, are initiated) at a lower level in the system, the modularization must be finer grained than partitions. Instead of discussing, for example, how computer systems manufactured by Digital Equipment Corporation (DEC) and International Business Machines Corporation (IBM) communicate with each other, we must move down to the level of interactions between processes.

In the following, I offer two new paradigms for modularizing network protection. In the first, we will categorize systems based on the amount of internal mechanism needed to provide adequate security in the operating environment seen by the system. Systems requiring the same degree of trust will be characterized by an index number called the "Trust Index." The idea is to focus attention and to place security mechanisms, which are expensive, where they are most needed.

[†]Boldface words are being defined. The definition follows immediately.

The other paradigm, based on ideas of the secure telephone system, rejects the need for centralized control of the network. Security responsibility is distributed to its logical extreme to see what the effects might be. Neither one of these paradigms solves all of the problems and renders security easy. It is not easy; many hard problems still have to be solved. However, it may be possible to survive the imminent collapse of our ability to protect data in large networks.

## 3 Protecting VIPs

The problem of protecting important people (a physical security problem, by the way) offers some interesting insights. To begin, the problem is dramatically different from the problem of protecting fixed plants. There is no fixed, or even very well determined, security perimeter. The asset is continually on the move through an environment that is largely friendly but is presumed to contain some very hostile elements. It may or may not be possible to identify the hostile elements when they are seen, and their intentions are unknown.

This problem, which seems much more difficult than protecting a fixed and slowly changing computer network, is solved every day. We should be able to draw some lessons from how it is done that can be applied to the network problem. One technique that is relatively easy to port is the idea of many layers of protection. Protection of a Head of State includes several layers, probably at least four. The strongest protection is immediately around the asset, and the layers get progressively weaker as the distance from the asset increases. The outer layers, however, are not only weaker, they are less trusted than the inner layers. The outer-most security layer for a Head of State probably consists of increased surveillance by the local police. While it is performing a valuable service, it is completely untrusted by the VIP's immediate bodyguard.

Sensitive data requires protection. More sensitive data requires more protection. However, sensitivity alone does not mean that protection mechanisms must be built into the system. Often, better protection of data can be obtained by physical means. The requirement for internal protection mechanism arises from the need to operate over a range of sensitivities, either in the data or in the authorization of the users. It is because of the range of sensitivities that we must trust the system to make critical decisions: should this person obtain that data; should this process perform that function. "More sensitive data requires more protection," may simply mean a stronger lock on the door. However, a greater range of sensitivity requires stronger internal protection mechanisms.

Different components of a network see different local environments. If the network as a whole processes a wide range of sensitivities, then some components will be faced with a range of sensitivities and will require enough mechanism and enough trust (assurance of correctness) to handle the range. However, as with the bodyguard, there is no reason to suppose that all components of a network need to be trusted to the same extent. Hence, for the Trust Index a label is used to distinguish components that must be highly trusted from those that can be trusted less.

The Trust Index satisfies the need to modularize. Using it, one can divide the network into regions that are equivalent in the sense that all connected components with the same trust index "see" the same protection environment and process the same range of sensitivities. It is then possible to consider each region as a unit and assess the requirements for controlling flow between the units.

As with partitioning, the trust index decomposition imposes an equivalence class structure on the network where the number of equivalence classes is much smaller than the number of components. Even better, the number of classes does not usually grow when new nodes are added to the network. A formal data flow policy can be described that provides rules for moving data between different trust index regions. One can limit the exposure of data by forbidding direct communication between components that differ greatly in trustworthiness. Instead, data flows gradually from highly trusted components through components that require less trust because they are protecting less and making less complicated decisions. So, for example, the policy would not allow direct connection of an "open"[‡] workstation to a system processing sensitive data, but it would allow indirect, appropriately protected data flows. A somewhat analogous situation can be seen in VIP protection. Generally, people are allowed to move fairly freely into and out of the immediate area occupied by the asset. However, high speed movement that appears to be directly toward the asset would be stopped early, as far away from the asset as possible-a soft and permeable boundary that stiffens rapidly as a function of the rate of penetration.

[‡]An **open** system means that the system can be made available to users without consideration of their clearance. It does not necessarily mean that use is completely unrestricted.

## 4 The Secure Telephone System

There are several ways to simplify the security structure of networks. The previous section suggested a decomposition of network components that focuses design attention on the components that need internal mechanism to fulfill their responsibilities. It may be very useful in connection-rich environments where it is difficult to establish the security perimeter or where external connections are needed even though sensitive data is to be processed locally. The trust index decomposition rejects the notion that complete isolation is necessary to be able to protect sensitive data. The following paradigm rejects the notion that hierarchical control is necessary.

In the secure telephone system security responsibility is distributed out to the users. A network mechanism is provided that obviates any security mechanisms built into the network itself. It is not even necessary to know, at the time a call is initiated, that secure communication is possible. The connection is established, the need for and possibility of security is negotiated, and the secure connection is established using a trusted third party. After all this has happened the communicating parties still have the option of deciding not to communicate-the final access control occurs using the secure connection.

Security in the telephone system is established using a mixture of a particular encryption technology and human judgement. A computing network could be established today using this technology, but it would not be a very capable network, and this is not what I am suggesting. What I have in mind is much more radical: it is to distribute responsibility for protecting data to the data itself.

Suppose that an object[§] were really able to guarantee that the only way to get to its data is through its methods.[¶] Let's consider the potential effects on the security requirements for a network. To be gelatinous, if not concrete, consider a local area network comprised of workstations (not necessarily single user, but probably one at a time except for NFS mounts), and a print server. The paradigm is that a process object obtains information from a data object by sending it a message. Access control is performed by the data object according to the object's policy. The object's policy may be different from every other object's policy and may be quite complex. It may, for example, include consideration of user identity, clearance, and role. It may also include consideration of the local processing environments of both the requesting object and itself and many other things.

If protection responsibility and capability are given to a data object, then many other requirements could disappear. There would no longer be any security requirement on the transmission medium itself. There would be no security requirements on a file storage system other than being able to return objects that had been previously stored. This would take care of two currently pressing issues: how to protect data when all the users use removable media, and how to securely implement distributed file systems. Even the access control requirements for workstations would disappear-users would establish access rights directly with the data object, not the underlying system.

Operation of a print server would be somewhat different than it currently is because of the necessity to establish dynamically that the data to be printed is printable there. Security restrictions might prevent creating particular documents on particular printers, and these restrictions can vary in time. What is permitted now may not be allowed in ten minutes if a particular person leaves the room, for example. Thus, when requested to print, a data object would not respond with a stream of text that could be sent to a printer. Instead, it would respond by creating a printable object that would be able to decide, through its methods, whether to print on a particular printer. This object would be sent to the print server, would negotiate with the print server the conditions of printing, print the requested number of copies, and self destruct.

Obviously, there are many difficult problems that would have to be solved to make this vision a reality. The attractiveness of the paradigm comes from its ability to localize and simplify network security concerns. This happens partly because the need for hierarchical administration and global understanding of the network have been eliminated. An object containing sensitive data can freely be moved around the network. It is no longer necessary to consider whether the object can be accessed in a particular location before sending it there. Like the secure telephone system, where the central administration does not have a precise idea of how big the network is or exactly where all the nodes are located, the central network administration need not know the extent of the computer network or exactly where sensitive data is processed. The central administration establishes rules for local protection and local connection to the larger network, implements most of them in the object support mech-

[§]That is, not object as in subject and object, but *object à la* object-oriented programming.

[¶]Encryption may be a way to do this, but it may not be the only way or the best way.

anism, and leaves the rest to local administration.

## 5 Conclusions

In this paper we make the argument that centralization, isolation, and hierarchical control will ultimately defeat our ability to make large and complex networks that we can trust. Two new ways of looking at network security rejecting traditional notions of network security management and facilitating movement towards more complex, more capable trusted, networks. Although not argued here, both paradigms are formalizable and, hence, it will be possible to systematically study and demonstrate the correctness of trusted network systems.