Unhelpfulness as a Security Policy
or
It's About Time

Ruth Nelson
Information System Security
48 Hardy Ave.
Watertown, MA 02172

Abstract

This paper suggests the possibility of controlling the rate of release of information as well as whether the information can be released at all. If the user must have access to information, but does not require fast access to large amounts of data, the system can release the information to that user in a slow and unhelpful manner. The addition of the parameter of time acts as a deterrent to information collectors and intruders; less information is available, and the user must access the system repeatedly and for a longer time to get it.

Investigation of rate of release has led to further understanding of the principle of least privilege. The principle of least privilege has generally been espoused by the computer security as highly desirable. It has been applied to computer security, but only in limited ways. Consideration of time allows a refinement of the concept and offers the possibility of more flexible and fine-grained control.

Security as Risk Management

A view of security as risk management rather than as an absolute, measurable quality of a system has led to a number of useful concepts [1]. In this view, security is a tradeoff between controlling the risk to a system resource and permitting needed access to that resource. All accesses involve some risk; even "trusted" users or software may misbehave. The security designers goal is to understand the system functionality and operation so that the tradeoffs are made explicitly rather than abstractly. Application-specific security is an important part of the tradeoff, since the need to access particular resources depends on what they actually are, not just on the level of sensitivity they are assigned. Some elements of this security paradigm are as follows:

- **Security is local, not global.**

  Resources are physically located in some machine under the control of some management. System policies should reflect the specific intentions of the system management and the specific services offered by the local system.

- **Security is specific, not general.**

  General security policies can provide a starting point for security, but cannot reflect the application functionality of the system nor specify that control.

- **Security is concrete, not abstract.**

  The resources of the system include actual machines, software, and information. Managing these only as abstract subjects and objects is indirect, at best.

- **Security is relative, not absolute.**

  It is impossible to define security, let alone achieve it. The art is to minimize the risk of operation, and to record sufficient audit data to recognize and track possible intrusions and failures.

Rate of Release as a Security Parameter

Most current security policies specify whether certain information may be released to certain users or not. The decision is usually based on a comparison between the user's privileges and the sensitivity of the data, with no consideration of the amount of information to be released in a given time.

0-8186-7318-4/95 $4.00 © 1995 IEEE

It is interesting to consider policies where access to information is permitted, but where the information is released in a slow and reluctant fashion. The intent of policies like this is to discourage large scale collection of information, but to allow small amounts to be given out when needed. Since it takes a long time to collect information, the probability that unauthorized users will be detected is increased. All users will be discouraged from trying to get information out of the system because it is intentionally unhelpful. The unhelpfulness may help to protect the confidentiality of the system information and slow down but not stop its dissemination.

None of the formal policies in use today use rate of authorized release as a security parameter that can be designed into a system. However, there are several real, informal systems in which slow, reluctant release is a security feature.

NSA Unclassified Telephones

An example of an unhelpful, somewhat secure, system is the operator assisted telephone information system formerly accessible at the National Security Agency. The Agency telephone book (of the unclassified numbers) is considered classified, but small extracts from it are not. At one time, there was a publicly advertised information number for NSA. The information operators at NSA would give out the unclassified telephone extensions of Agency employees, but only if the caller knew the name of the employee, and they gave out only one number at a time. If a caller wanted more than one number, he or she had to call back - and probably wait long enough so that an operator would not recognize that numerous numbers were being collected by a single caller.

This system allowed outside callers to get the numbers of particular Agency employees, presumably for legitimate business. It did not allow "browsing," looking through the phone book for interesting names or departments. It also did not help employment agents, for example, to collect lists of Agency employees for recruitment or other nefarious purposes.

The Agency changed its policy since the development of the STU-III telephone. Operators no longer give out any phone numbers unless the caller is calling from a STU-III. The security value of this rule is not in the confidentiality of the exchange, but rather in the assurance of the identity of the caller. They have, in effect, added a source authentication policy to the slow release policy.

NSA Organization Charts

A similar, less automated system was and is in place for NSA organization charts. Organization charts at NSA are generally considered classified. However, small parts of those charts are unclassified for some parts of the Agency. Employees often give contractors partial charts of their organization, and mail must be marked with an organization designator for efficient delivery.

Marketing departments in large corporations collect this information and put together their own NSA organization charts, which are far more inclusive than any NSA employee would give out, and probably contain enough information to be classified by NSA standards. This is a practical example of aggregation happening. The policy on classification of the organizational information is clearly not complete, and NSA knows that the information is being released and aggregated.

There is actually a rate-of-release security factor operating to protect this data. NSA reorganizes rather frequently, so that the aggregated information is often out of date and unreliable. The relationship between the rate of dissemination of the organizational information and the rate of NSA reorganization is not fixed by any policy, however; this makes the protection afforded by the slow release hard to assess.

There are other examples where the real security policy of an organization is "unhelpfulness." Information must be released and used. The organization just wishes to slow the release down to near the minimum necessary for its operations. In the tactical military arena, the aspect of timely, last-minute release of information is well-understood, since at some point, troop movements or other military operations become highly visible and clearly not secret.

Principle of Least Privilege

Control of release rate offers a flexible, realistic way to increase security in information systems without preventing them from providing needed

services. The principle of least privilege states that security is enhanced by limiting privilege to the smallest amount necessary for proper function. The term "privilege" includes ability to control the system, as well as access to information. This principle has generally been espoused by the computer security community as very desirable. However, in computer systems, its meaning is generally limited to restriction of operating system privileges plus "Mandatory Access Control" (MAC)[2]. Privilege can be managed and controlled at a much finer level of granularity if axes other than system control and sensitivity label are considered.

This fine-grained, flexible control often requires that consideration of the specific system functions to be controlled and the semantic content of its data. Therefore, this type of least privilege is likely to be domain-specific. It does not have to be totally ad hoc, however, and systems can be designed and configured to provide the necessary support. The VISE [3] work illustrates this for functional limitation of access; similar support could be provided for unhelpfulness.

Simple Access Control

In systems that implement subject-object security policies such as MAC and DAC, the binary decision is applied to "objects" under the control of the system. The security enforcement mechanism compares the sensitivity label of the object with the privileges of the subject and makes a yes or no decision on the entire object, independent of the amount of information or data it contains. In the case of MAC, there are additional rules that specify appropriate data labels of information written by the subject, but these rules do not affect the nature of the access rules.

In some cases, the user's privileges may change. For example, a particular access may be permitted during normal working hours but not at night, or access may require permission of a supervisor. These restrictions still result in access being granted or denied based on the subject-object relationship.

Aggregation and Inference

Systems that provide protection against aggregation offer a refinement of this decision process, releasing a limited amount of information and attempting to prevent a user from collecting too much. In order to do this, the system must have some measure of the amount of information a subject has accessed and some limit on the amount it can have.

A further refinement is a security policy that considers the control of inference. Inference control requires consideration of the relationship of units of information, as well as the sensitivity of the information. Some systems provide a measure of application-independent inference control, but its effectiveness is limited, because inference is inherently semantic and the semantics of secrecy are complex and hard to structure [4].

Functional Limitation of Access

In many systems, the desired functionality is understood and is provided by a set of applications programs working on somewhat structured data. These functions can be managed and controlled, as pointed out by Clark and Wilson [5] and generalized by GTE [3]. Adding this dimension of control allows the security designer to specify more clearly the specific information that may be released by a system, and how that information is managed.

Unhelpfulness

To control the release rate of information, it is necessary to understand and manage a number of factors:

- Information sensitivity
- Information content
- Information quantity
- Information release history or some measure thereof.

Thus, an explicit unhelpfulness policy usefully incorporates existing access control structures and mechanisms, and adds to them the concept of time.

Tools for Unhelpfulness

The security profession has not addressed the issues of intentional, but slow or reluctant, information release. In particular, we do not have good mathematical tools to describe this type of policy or to measure the success of its enforcement by an automated system. There are mathematical

techniques used in other fields which may apply, and it would be interesting to investigate their use. Some possible analogies are radiation leakage, and exposure to pesticides in agriculture.

For investigation

This discussion was not intended to provide any conclusions about the usefulness of unhelpfulness as a security policy. Its intent is to demonstrate a parameter that seems helpful and relevant in

- managing secure systems
- refining the system approach to the principle of least privilege
- protecting our systems
- detecting system misuse.

We may not wish to incorporate rate management techniques in our systems. However, it is useful to recognize unhelpfulness security policies when they do exist and to consider adding them when they can improve security. If we do this, and if we incorporate support for managing and measuring the rate of information release from our systems, we may have more secure and more flexible systems. We can have intentional unhelpfulness for the sake of security, rather than random unhelpfulness based on ignorance.

References:

1. R. Nelson, D. Becker, J. Brunell and J. Heimann, "Mutual Suspicion for Network Security," Proceedings of the 13th National Computer Security Conference, Baltimore, MD, September 1990.

2. Department of Defense Trusted Computer Security Evaluation Criteria, DoD 5200.28-STD, National Computer Security Center, December 1985.

3. C. Limoges, R. Nelson, J. Heimann, D. Becker, "Versatile Integrity and Security Environment (VISE) for Computer Systems," New Security Paradigms Workshop II, Little Compton, RI, August 1994.

4. R. Nelson, "What is a Secret - and - What does that have to do with Computer Security?," New Security Paradigms Workshop II, Little Compton, RI, August 1994.

5. D. Clark and D. Wilson, "A Comparison of Commercial and Military Computer Security Policies," Proceedings of the 1987 IEEE Symposium on Security and Privacy, Oakland, CA, April 1987.