The Insider Problem Revisited

Matt Bishop
Dept. of Computer Science
University of California at Davis
One Shields Ave.
Davis, CA 95616-8562
email: bishop@cs.ucdavis.edu

1. Introduction

The "insider threat" or "insider problem" is cited as the most serious security problem in many studies. It is also considered the most difficult problem to deal with, because an "insider" has information and capabilities not known to other, external attackers. But the studies rarely define what the "insider threat" is, or define it nebulously. The difficulty in handling the "insider threat" is reasonable under those circumstances; if one cannot define a problem precisely, how can one approach a solution, let alone know when the problem is solved?

Definitions of the "insider threat" have some common elements. For example, a workshop report [4] defined the problem as *malevolent (or possibly inadvertent) actions by an already trusted person with access to sensitive information and information systems*.[1] Elsewhere, that same report defined an insider as *someone with access, privilege, or knowledge of information systems and services*.[2] Another report [6] implicitly defined an insider as *anyone operating inside the security perimeter*.[3]

Each of these definitions seems reasonable. Consider the following three scenarios.

1. A manager at a military base learns she is about to be dismissed. She enciphers critical files on the system, and offers to provide the deciphering key if the contractor pays her $10,000 as severance pay, and agrees not to prosecute her. The contractor agrees. Here, the manager did not damage information on the system; she simply denied access for some time. Without access to the system, and knowledge of which files were critical, the attack would have failed. This is an example of an "insider attack" [8] satisfying the first definition.
2. A system administrator at a bank accesses a financial system that she is responsible for. She notices that $10,000,000 was transferred from account 1011 to account 6734. The account of a close friend is number 6834. She moves the money to her friend's account, alters the original log file entry to change the number to that of her friend's account, and deletes the log entries showing the money being moved from account 6734 to account 6834. This seems to qualify as "unauthorized access by an insider," which is one of the types of misuse reported in the *2003 FBI/CSI Computer Crime and Security Survey* [5]. This also satisfies the second definition.
3. A janitor at a large company takes papers with the social security numbers of employees from the trash and uses this information to commit identity fraud. This satisfies the third definition.

---

1. [4], p. xi.
2. [4], p. 10.
3. The report contrasts insider threats with "problems that originate from outside the perimeter, such as denial of service attacks, worm infections, and website defacements" ([6], p. 3).

The differences among these definitions illuminates the different interpretations of the "insider threat." In the first, the attacker has access to information and/or resources, and is trusted in some way, presumably not to abuse that information or resource. Note the abuse may arise from violations of the security policy, or for actions not covered by the security policy but covered by the trust; we shall elaborate on this later. The second definition broadens the notion of an insider to include anyone who knows something specific about the systems and services under consideration, whether they have access or not. The third definition covers anyone within the security perimeter, regardless of their level of privilege. Presumably, access is required; but access may be only to one entity within the security perimeter, and need not be the system under discussion. A janitor, for example, may have physical access to the console of a system when he cleans a room. Given these three definitions, the "insider threat" refers to different problems, requiring different solutions.

The goal of this paper is to provide a definition of an "insider," and hence the "insider threat," that is based upon the common elements of most definitions. It uses this new definition to suggest an approach to handling the "insider threat" using existing mechanisms and research to ameliorate the problem.

2. Background

Some observations will clarify the ideas underlying this suggestion. Although these are well-known, they bear repeating for the applications we will make.

2.1. Security Policy

A security policy defines "security" for a given site or set of sites.Most security policies provide for *trusted users* to whom the policy either does not apply or to whom some parts of the policy do not apply. For example, in a traditional Bell-LaPadula model with strong tranquility, labels of entities do not change. In practise, this is too restrictive, so a trusted user (the *site security officer*) is allowed to set and change labels. Indeed, in their demonstration that Multics satisfies the model [1], Bell and LaPadula explicitly defined trusted users as subjects against whom the *-property is not enforced. The users are trusted not to violate that property.[4]

Most computer systems provide such a user as a matter of convenience. The best known examples are those of the "superuser" or "root" on UNIX and UNIX-like systems, and "Administrator" on Windows systems. The rationale is that such users can intercede when something goes wrong, and repair the damage or prevent further damage. A security policy may constrain actions based upon procedural mechanisms, but typically there are few if any enforcement mechanisms controlling the use of these accounts.

2.2. Trust and Assumptions.

All security rests on trust. Security policies embody trust in a number of ways. First is that when trusted entities violate the rules of the policy, they do so for good cause. If this assumption is incorrect, a trusted entity may cause damage or loss against which the policy is to guard—in short, the policy and its enforcement mechanisms would be ineffective. Second is that the security policy partitions all states of the system into "allowed" and "disallowed" states. If there are states

4. See [1], Table 1, p. 78.

not described by the security policy, this assumption is violated because it is unclear whether the states are "allowed" or "disallowed."

Policies defined non-rigorously—for example, in a natural language like English, French, or Russian—also make assumptions about environment and other legal and cultural constraints that inhibit certain enforcement mechanisms, or require the use of others. For example, if a country requires that all cryptographic keys be registered with the police, any cryptographic mechanisms must be augmented with an enforcement mechanism to transmit the key to the police. Similarly, if a culture values privacy, enforcement mechanisms requiring users to reveal personal information will not work well.

3. Redefining the Insider Threat

Consider the notion of an "insider" first. Merriam-Webster's dictionary [9] defines an "insider" as a "member of a group, category, or organization: as … a person who is in a position of power or has access to confidential information; or … one (as an officer or director) who is in a position to have special knowledge of the affairs or to influence the decisions of a company." In the context of a threat, this definition means that the insider uses that access, knowledge, or information to do something nefarious. So the insider must be trusted not to violate the confidences entrusted to her.

In terms of computer security, the insider is one who has some property that distinguishes her from others. This property requires that the insider be able to take action that would violate the security policy *were it done by an untrusted user*. The insider is trusted to do so only when appropriate. As an example, consider the proof for *get-read* satisfying the Bell-LaPadula mode. The "trusted subject" is a subject whose current security level does not dominate that of the object.[5] Were the subject to act nefariously, the *-property would not hold.

The specific property, or properties, that distinguish an insider from other users, varies among different situations. Example properties are:

1. Having access to sensitive information and information systems (this corresponds to the first definition in the Introduction);
2. Knowing something about a specific information system or service, or having access or privileges to those (the second definition in the Introduction)
3. Operating inside a security perimeter (the third definition in the Introduction)

Combine this with the notion of a policy. A security policy makes specific rules about what is, and is not, allowed. An attack must violate some rule in the security policy. Trusted entities may violate some rule in the security policy, but such a violation by a trusted entity is not considered an attack. So enforcement mechanisms do not block a trusted entity from violating the security policy.

We therefore propose the following definition:

Definition 1. An *insider with respect to rules R* is a user who may take action that would violate some set of rules *R* in the security policy were the user not trusted. The insider is trusted to take the action only when appropriate, as determined by the insider's discretion.

---

5. The model requires that the maximum security level of the subject dominate the security level of the object, even when the subject is trusted.

When the set of rules is clear from context, or irrelevant, we will omit "with respect to rules *R*." The "properties" mentioned above are the ability to violate the rules in *R*.

Definition 2. The key notion underlying this definition is that the insider is trusted to know when to violate some aspect of the security policy. As examples:

1. under the rule of controlled physical access to the computer, a janitor with access to the room containing the computer is an insider because he has that access.
2. under the rule of access to the computer system allowed to authorized users only, an ordinary user is an insider because she can give access to anyone by sharing her password.

Definition 3. The *insider threat* is the threat that the insider may abuse her discretion by taking actions that would violate the security policy when such actions are not warranted.

Because the insider is trusted, security enforcement mechanisms will not block the illicit action. Continuing with our examples:

1. The janitor has keys to the room where the computer is, so he can enter and access the physical device. The enforcement mechanisms that keep untrusted people out of the room would not keep the janitor out, as he is trusted to access (and clean) the room.
2. The user knows her password and account name, and can give those to anyone. The enforcement mechanism assumes that anyone knowing the account name and password is an authorized user, and so cannot detect violations of the rule.

For completeness, we also define the insider problem.

Definition 4. The *insider problem* is the problem of dealing with the insider threat.

As a result of these definitions, the "insider problem" is tied to a lack of enforcement mechanisms. Specifically, an insider is an entity about which the policy makes assumptions, specifically that the entities are trusted to act in specific ways. Because of the nature of these assumptions, the enforcement mechanisms will *not* detect violations of the trust. This characterizes the insider problem: a violation of trust.

4. Approach to the Problem

Two different aspects of the problem arise from our definitions. The first derives from the inability of enforcement mechanisms to validate actions by insiders. The second arises from the assumptions underlying security policies.

4.1. Enforcement Mechanisms

Enforcement mechanisms fail to detect attacks from insiders with respect to rules *R* because, by definition, insiders are trusted not to attack the system by violating the rules in *R*. So, any violation of those rules is for good cause ("as determined by the insider's discretion"). This means that preventing insider attacks is feasible only if a mechanism can determine the reason for the violation of the rules in *R*, and if the reason constitutes an abuse of discretion, block that violation. This can be done only if the discretion is narrow and clear bases for violating the rules exist. But if that were true, those cases could be put into the security policy, negating the need for trust. Hence, preventing insider attacks is not feasible.

But detecting them may be feasible. The method for detection is to have enforcement mechanisms record violations of the rules in *R*, and then analyze them looking for anomalous behavior. The anomalies here are defined with respect to the expected behavior of an insider with respect to

the rules in *R*. If the purview of the detection mechanisms is broad enough to capture actions that would constitute an abuse of discretion, then insider attacks can be detected after the fact.

The key is to define what constitutes an "abuse of discretion." The principle of fail-safe defaults [7] says that one should define those actions that are allowed, and disallow all others. Unfortunately, this approach will not work because the essence of discretion and trust is to enable people to handle unexpected situations. In many cases, abuses are defined after the fact, when the nature of the abuse, or the reason that particular actions are undesirable, becomes clear—the exact opposite of the approach suggested by the principle. In other cases, the abusive actions may occur beyond the scope of the system, for example with the copying down and selling of information.

For this reason, it appears that anomaly detection is the method most likely to detect insider attacks based upon the inability of the enforcement mechanisms to stop compromise by trusted users. At that point, an administration may provide corrective procedural mechanisms (such as removing the offender from a sensitive job, or firing the offender). Even better would be to revise the security policy to bar the actions that were deemed a violation of discretion, and (when possible) develop enforcement mechanisms to prevent repetitions. This reduces the assumptions underlying the policy.

One could also use the anomalies to create a "shadow policy" that applied only to trusted users. This policy would provide fine-grained reasons for allowing access to resources, or denying access to resources. As anomalies are detected, a machine learning program would assess whether the anomaly constituted an insider attack, and if so adjust the shadow policy accordingly. Before a trusted user could perform some action, the action would be compared to the shadow policy. If the action were allowed, it would not be recorded; otherwise, it would be recorded for duture analysis (and, possibly, blocked). For ordinary users, of course, the shadow policy would never be examined.

As an example, consider a subject on a system implementing the Bell-LaPadula model. A shadow policy states that subjects that read an object at a higher security level than the subject, and then write to an object at the same security level as the subject, are leaking information from the first object. A subject's current security level is L1, and its maximum security level is L3, where L3 dom L1. The subject accesses an object with security classification L2. As the subject is trusted, the read is allowed but recorded. If the subject writes immediately thereafter, this would violate the shadow policy. Either the write is logged and an alert raised, or the write is blocked. In the first case, the security officer could intervene to determine if the write did leak information, and take remedial action if so. In the second case, the security officer could examine the request, and decide whether to allow or deny the requested write.

4.2. Assumptions in Policy

If a security policy partitioned all states of the system into allowed states and disallowed states, then the only trust involved would be that explicitly stated in the policy. Unfortunately, policies rarely partition the states into two sets. Actual policies have a third set, containing states which may or may not be allowed because the policy does not describe them, or is ambiguous in its description of them. These states may confer privileges on entities unintentionally.

Consider a bank policy that states that all checks over $100,000 must be signed by both the comptroller and the president of the bank. Implicit in this rule is that the comptroller and the president are roles filled by two distinct people. If not, one person could issue a check for $200,000.

This would violate the principle of separation of duty, which the policy intends to enforce. But as stated, there is no requirement that the two roles be filled by two distinct persons.

Here, the enforcement mechanisms will enforce the security policy. Thus, even though the president and the comptroller are not trusted entities in the sense of being able to override the rule of separation of duty, in the particular case of both roles being filled by the same individual, that rule does not apply. Hence for practical purposes, the individual is trusted.

The difference between this situation and the previous one is here the trust is implicitly, not explicitly, given. Detecting these situations is analogous to the situation of explicit trust, once that trust has been identified and made explicit. Techniques used to uncover assumptions [2,3] may be applied to do so.

5. Conclusion

This paper presents a new paradigm for the "insider threat." Specifically, an insider is a trusted entity that is given the power to violate one or more rules in a given security policy. Enforcement mechanisms are not applied against those trusted users. The insider threat occurs when a trusted entity abuses that power. The key issue in the insider threat is to determine what uses constitute abuse of that power.

Further, handling the insider threat requires alternate enforcement mechanisms, because those predicated upon the security policy fill fail by definition of the "insider." Either the policy must be augmented to eliminate (some of) the trust, or enforcement mechanisms external to the policy must be defined and implemented.

This view of the insider problem leads to an interesting observation. The insider problem is not "a" problem. Rather, it is a continuum of problems, ranging from the case of a rogue user with little to no privileges to the case of an official with a large number of privileges. What distinguishes the insider problem from others is that policy-based enforcement mechanisms will not work, because the explicit granting of trust creates an exception that those mechanisms honor.

Acknowledgement: Thanks to Karl Levitt and Todd Heberlein for insightful discussions. This work was supported by award CCR-0311723 from the National Science Foundation to the University of California, Davis.

6. References

[1] D. Bell and L. La Padula, *Secure Computer System: Unified Exposition and Multics Interpretation*, ESD-TR-75-306, ESD/AFSC, Hanscom AFB, Bedford, MA 01731 (1975).

[2] M. Bishop, "Best Practices and Worst Assumptions," *Proceedings of the 2005 Colloquium on Information Systems Security Education* (June 2005).

[3] M. Bishop and H. Armstrong, "Uncovering Assumptions in Information Security," *Proceedings of the Fourth World Conference on Information Security Education* (May 2005).

[4] R. Brackney and R. Anderson, *Understanding the Insider Threat: Proceedings of a March 2004 Workshop*, RAND Corp., Santa Monica, CA (Mar. 2004).

[5] *CSI/FBI Computer Crime and Security Survey*, Computer Security Institute (2003).

[6] J. Patzakis, *New Incident Response Best Practices: Patch and Proceed is No Longer Acceptable Incident Response Procedure*, Guidance Software, Pasadena, CA (Sep. 2003).

[7] J. Saltzer and M. Schroeder, “The Protection of Information in Computer Systems,” *Proceedings of the IEEE* **63** (9) pp. 1278–1308 (Sep. 1975).

[8] E. Shaw, K. Ruby, and J. Post, “The Insider Threat to Information Systems,” *Security Awareness Bulletin* No. 2-98, Department of Defense Security Institute (Sep. 1998); available at http://rf-web.tamu.edu/security/secguide/Treason/Infosys.htm.

[9] Merriam-Webster Collegiate Dictionary, available at http://search.eb.com/dictionary.