Title

A Risk Management Approach to the 'Insider Threat'

Permalink

Authors

Bishop, Matt
Engle, Sophie
Frincke, Deborah A
et al.

Publication Date

2010

DOI

10.1007/978-1-4419-7133-3

A Risk Management Approach to the “Insider Threat”

Matt Bishop, Sophie Engle, Deborah A. Frincke, Carrie Gates, Frank L. Greitzer, Sean Peisert, and Sean Whalen

Abstract Recent surveys indicate that the financial impact and operating losses due to insider intrusions are increasing. But these studies often disagree on what constitutes an “insider;” indeed, many define it only implicitly. In theory, appropriate selection of, and enforcement of, properly specified security policies should prevent legitimate users from abusing their access to computer systems, information, and other resources. However, even if policies could be expressed precisely, the natural mapping between the natural language expression of a security policy, and the expression of that policy in a form that can be implemented on a computer system or network, creates gaps in enforcement. This paper defines “insider” precisely, in terms of these gaps, and explores an access-based model for analyzing threats that include those usually termed “insider threats.” This model enables an organization

Matt Bishop
Matt Bishop, Dept. of Computer Science, University of California at Davis e-mail: bishop@cs.ucdavis.edu

Sophie Engle
Sophie Engle, Dept. of Computer Science, University of California at Davis e-mail: sjengle@ucdavis.edu

Deborah A. Frincke
Deborah A. Frincke, Pacific Northwest National Laboratory e-mail: deborah.frincke@pnl.gov

Carrie Gates
Carrie Gates, CA Labs., Inc. e-mail: carrie.gates@ca.com

Frank L. Greitzer
Frank L. Greitzer, Pacific Northwest National Laboratory e-mail: frank.greitzer@pnl.gov

Sean Peisert
Sean Peisert, Dept. of Computer Science, University of California at Davis e-mail: peisert@cs.ucdavis.edu

Sean Whalen
Sean Whalen, Dept. of Computer Science, University of California at Davis e-mail: shwhalen@ucdavis.edu

to order its resources based on the business value for that resource and of the information it contains. By identifying those users with access to high-value resources, we obtain an ordered list of users who can cause the greatest amount of damage. Concurrently with this, we examine psychological indicators in order to determine which users are at the greatest risk of acting inappropriately. We conclude by examining how to merge this model with one of forensic logging and auditing.

1 Introduction

Modern culture considers the "insider" a term of both honor and opprobrium, implying that the person so identified is privy to access or information that others are excluded from. Therefore, people should regard their words or actions as less questionable because they are undoubtedly based on more information, or better information, than others have available. But the insider, being privy to that access or information, can wreak far more damage than an outsider by exploiting their knowledge and abilities.

These widely held opinions define "insider" as some mixture of access and knowledge—and more often than not, the term is never explicitly defined. Even when it is, the definitions are often contradictory or specific to a particular environment. For example:

1. An insider is someone who is authorized to use computers and networks [Sch02];
2. An insider has access to the keying materials or full control of some nodes [NS05];
3. An insider has "access, privilege, or knowledge of information systems and services" ([BA04], p. 10)
4. An insider is anyone who operated inside the security perimeter [Pat03]; and
5. An insider is a database subject who has personal knowledge of information stored in one or more fields marked confidential [GGG02].

These definitions are different. For example, the second definition requires "full control" of nodes, whereas the fourth simply requires someone to be within the security perimeter. The first applies to anyone who has permission to use the computers and networks, whether or not they have full control of nodes or are within the security perimeter. The third definition does not require authorization; it merely requires knowledge. And the last definition is specific to database systems.

Two factors underlie these definitions. The first is *access*. All require the insider to have some degree of access to resources. The mode of access differs: "use," "read," "control," or "write," for example. So does the level of access, for example "privileged," "ordinary," or "full." The second is *knowledge*. Knowing something about the information systems and services or values in the database implies the ability to act on that knowledge. For our purposes, we consider knowledge a form of access because it implies one knows data about *that particular organization*. In other words, knowing that a user named "root" has password "gleep" is useless unless one knows that the Department of Redundancy Department has on its server that particular user with that particular password.

A third factor, implicitly mentioned by the above, is *trust*. Authorization implies some level of trust, as does access to keying materials or operating within a security perimeter. As with knowledge, we consider trust a form of access because in practice, organizations trust those with access. How this trust is granted differs: employment implies some level of trust; a security clearance implies a (perhaps more general) type of trust. For our purposes, trust as an intangible expression of assurance in the rectitude or predictability of an individual is not relevant. If that trust leads to the granting of access, or the imparting of knowledge, then it is relevant because of the effects of the trust, not because of the trust in and of itself. This again causes us to focus on access [BG08].

The implication of the distinction between "insider" and "outsider" is that they have different capabilities or affect the organizations in question differently, and that one can provide a precise characterization of both the differences between an "insider" and an "outsider." The problem is that such a characterization is rarely attempted, the papers relying on the reader's intuition; and when it is attempted, the characterization is either imprecise or specific to a particular set of facts. The above examples show this. Further, the difference in capabilities relies on knowledge or access, and does not provide a precise set of knowledge or access capabilities that lead to the distinction between "insider" and "outsider."

Our theme is that the distinction between "insider" and "outsider" is not binary; rather, there are "attackers" with varying degrees and types of access. One can call some set of these attackers "insiders," with the complement being the "outsiders," but countermeasures should focus on the access and not on whether the attackers are insiders. Thus, we see attacks as spanning a continuum of levels and types of access, and use that as the basis of our discussion. We emphasize that people comfortable thinking in terms of "insiders" and "outsiders" can superimpose that partition on our notion of "attackers with varying levels of access." That partition, however, will vary based on circumstances and environment.

The next section describes our model in detail, discussing how we use access to identify sets of users and resources of interest. We can then refine membership in our sets using various techniques; here, we focus on psychological indicator-based assessment techniques. Following this, we examine countermeasures.

2 Insider Threat Assessment

Methods of detecting insiders have two distinct parts. First is to determine whom to worry about; in other words, who has the capability to launch such an attack. Second is to determine who is likely to attack; this is psychological, because having a capability does not mean it will be used. We consider these separately.

As with all security, we begin with the security policy. This poses an immediate problem. *Which* security policy do we begin with? The obvious answer (the current policy) is insufficient because policies are imprecise. Even policies expressed using policy specification languages only capture part of the site's intended policy. In fact, a "policy" in practice is an abstract expression of requirements, desires, and other factors. As it is refined, different instantiations emerge.

Carlson's Unifying Policy Hierarchy [Car06, BEG$^+$08] provides a framework for capturing the different refinements (see Table 1). At the top of the hierarchy is the *Oracle Policy*. This is a mechanism that can supply a correct, precise answer for any policy question asked. It is a management mechanism, not a technical one. It can include non-technical factors such as intent, custom, law, and so forth. In practice, it is non-deterministic and not Turing-computable. Hence it should be viewed as the "ideal, intended policy" even when that policy, or parts of that policy, are unclear. An example statement in an Oracle Policy would be:

> Mary is allowed to read patient medical data for the purpose of compiling statistics for CDC reports on the geographic spread of medical conditions.

This statement specified an action ("read"), a resource ("patient medical data"), and a condition ("for the purpose of compiling statistics ..."). The purpose embodies intended use; Mary cannot read the data, for example, to sell it to anyone. In terms of the domains in Table 1, the intent is a condition e that must be met for the subject s (Mary) to be able to perform action a (read) on object o (the patient medical data).

The Oracle Policy contains elements that are ambiguous when mapped onto a computer system; it also contains elements that are infeasible for a computer system to enforce. For example, what exactly is "patient medical data?" Presumably, data on the system is labeled in some way, for example by being in certain (sets of) directories or having a tag marking it as PATIENT or CONFIDENTIAL. But if the data is stored in a different file, and unlabeled, Mary may not realize it is confidential, protected data and thus she may reveal it. This is an imprecision. To take this a step farther, Mary has authority to access even data that is so labeled when she needs to for her job (compiling statistics). She does not have that permission when she needs

Table 1 The Unifying Policy Hierarchy. The entities are subjects $s \in S$, objects $o \in O$, and actions $a \in A$. The condition $e \in E$ describes additional constraints on the ability of s to execute a on o due to external factors such as intent. The "Run-Time Instantiation" is, strictly speaking, not a policy, but instead a description of what a user can do, whether those actions are authorized or unauthorized; that is, it encompassed unauthorized actions possible due to security flaws.

Level of Policy	*Domain*	*Description*
Oracle Policy	$S \times O \times A \times E$	What should ideally be authorized, including intentions.
Feasible Policy	A subset of $S \times O \times A$ containing system-definable entities	What can be authorized in practice, considering system constraints.
Configured Policy	A subset of $S \times O \times A$ containing system-defined entities	What is allowed by the system configuration.
Run-Time Instantiation	A subset of $S \times O \times A$ containing system-defined entities	What is possible on the system, factoring in any flaws or vulnerabilities.

to access it for a purpose unrelated to her job (selling the data). This is embodied in the condition *e*, intent, mentioned earlier. But the computer system and controls cannot read minds or divine intent with perfect accuracy any more than humans can; thus, security controls in this situation do not impede access.[1] So, those aspects of the policy that deal with intent, for example, are infeasible for a computer system to enforce.

Thus, the Oracle Policy is an idealized statement of what the exact security policy would be. It is unambiguous, and able to provide a decision about each quadruple. It partitions all possible states into two sets, "allowed" and "not allowed." The Oracle Policy may evolve over time; but in all cases, at all times, it can determine whether a given state is in the "allowed" partition of states, or the "not allowed" partition of states.

The *Feasible Policy* is a refinement of the Oracle Policy that can actually be implemented on a (possibly idealized) computer system. It differs from the Oracle Policy in that the Feasible Policy is grounded in technology and feasible procedures. For example, the above Oracle Policy statement would be represented as

> The account "mary" has read access to patient medical data.

Here, the notion of intent has been jettisoned, because (as of this writing) a computer system cannot determine intent. The exact mechanism that the computer uses to identify data as "patient medical data" is left unspecified because there are several ways to do so; all are implementation-dependent.

In most cases, the Feasible Policy does not capture the Oracle Policy precisely. Thus, there are "gaps" between the Feasible Policy and the Oracle Policy. For example, under the Feasible Policy, Mary is allowed to read patient data even when she plans to sell it. Under the Oracle Policy, Mary would not be allowed to read the data in that case.

In fact, the above statement glosses over a second gap: the difference between Mary and her account. Anyone who gains access to Mary's account can act as Mary. Therefore, the Feasible Policy does not restrict access to Mary. Rather, it restricts access to the associated account "mary."

The instantiation of the Feasible Policy on a particular system is called the *Configuration Policy*. Unlike the Feasible Policy, the Configuration Policy is aimed at a particular system, and its features constrain the instantiation of the policy. For example, the Configured Policy might represent the above Feasible Policy statement as:

> The account "mary" has read access to files in the directory "DBMS:patientdata" with suffix "pmd"

because on this system, all patient data is kept in files with names ending in "pmd" and in the directory "DBMS:patientdata." This expression can be instantiated using the file access controls on the system, the naming conventions for files

[1] Instead, they try to detect when this type of breach has occurred, or try to deter this type of breach through a variety of means.

and directories, and the directory structure. As with the Oracle and Feasible Policies, there are gaps between the Feasible Policy and the Configuration Policy. For example, if patient medical data were put in files other than those identified in the Configuration Policy, Mary might not have access to them. Thus, this denies her access to data that the Feasible Policy specifies she should have access to.

This also points out an interesting aspect of the gaps. The gap between the Oracle and Feasible Policies identified above is one of *granting* rights so that the Feasible Policy does not contain a restriction (denial of rights) that the Oracle Policy imposed. The instantiation provides subjects more rights than they should have. The gap between the Feasible and Configuration Policies identified above is one of *denying* rights, so that the Configuration Policy grants a right that the Feasible Policy does not contain. The instantiation provides subjects with fewer rights than they should have. For our purposes, we focus on the former type of gap. Gaps created by the deletion of rights enable denial of service attacks, and they can be treated in the same way as gaps created by the granting of rights.

Ideally, when the Configuration Policy conforms to the Feasible Policy, the Configuration Policy would describe the capabilities of every subject on a system. Were the system implemented correctly, the subjects would be so constrained. But in practice, software has security flaws, called *vulnerabilities*. These vulnerabilities add rights.[2] For example, suppose a buffer overflow in a privileged program allows Mary to obtain the privileges of a system administrator. The rights she has are no longer those that the Configured Policy gives her. Therefore, the *Run-Time Instantiation* describes the actions that subjects can take on objects.

As with the other pairs of policies, there exist gaps between the Configured Policy and the Run-Time Instantiation. In our example, when Mary exploits the vulnerability in the privileged program to acquire the system administrator privileges, she can now access files that the Configured Policy intended to deny her access to.

2.1 Example

Consider a company that has a policy of deleting accounts of employees who leave (either voluntarily or because they are fired). The system administrators are responsible for doing this. Thus, there are tools to delete accounts, and a mechanism for determining when an account is to be deleted–perhaps the Human Resources Division sends the system administrator a note. The three highest policies would be:

- Oracle Policy: When an employee leaves, the employee is to lose all access to the company systems.
- Feasible Policy: Upon notification from the Human Resources Division that an employee has left the company, the primary account associated with that employee is to be disabled.

[2] Of course, they may also delete rights.

- Configuration Policy: The password and remote access authorization mechanisms for the primary account associated with an employee are to be disabled, so the employee cannot access the account.

Here, the Oracle and Feasible policies overlap considerably, but two gaps are apparent. The first stems from the relationship between "an employee leaves" and "notification ... that an employee has left." If the employee does not notify the company she is leaving, but simply stops coming to work, the company may not realize that the employee has left. Thus, the account will not be deleted even though the employee has left. A second, less speculative version of this gap arises when the Human Resources Division fails to notify the system administrators in a timely fashion. In that case, the departed employee still has access to the company systems.

The second gap is the distinction between "lose all access to the company systems" and "the primary account ... is to be disabled." The assumption here is that without an account, the departed employee cannot access the system. This assumption relies on the employee having no access to any other account, or any other means for gaining access (such as connecting to a web server on a company system or using FTP to access files in an FTP directory). Should the employee be able to do so, the Feasible Policy will grant rights that the Oracle Policy denies, creating the gap.

A similar gap exists between the Feasible Policy and the Configured Policy. The Feasible Policy requires that the employee be denied access to the primary account. The Configured Policy describes how this is to be done on the system in question—here, by disabling the password (so the user cannot authenticate correctly by supplying the password) and by deleting any indicators of "remote trust" that enable remote users to access the account without authenticating locally.[3] The gap lies in the assumption that those actions will disable the account. If the user has set up a program that runs every evening and mails important data to the employee's Gmail account, then the steps required by the Configuration Policy do not achieve the desired goal, that of disabling the account.

Finally, even were there no gaps between any of the above three policies, one must consider the actual policy enforced by the system, the Run-Time Instantiation. Suppose the system runs a web server with a buffer overflow vulnerability that starts a command interpreter when an attacker triggers the overflow. Thus, the Run-Time Instantiation gives the departed employee access to the company system.

This suggests a precise definition for the notion of an "insider."

Definition 1. Let P_L and P_H be representations of a policy at different levels of the Unifying Policy Hierarchy. If a subject has a different set of rights in P_L than it has in P_H, it is called an *insider*. An *insider attack* occurs when an insider employs any rights that exist in P_L and that do not exist in P_H.

In our example above, suppose Nancy leaves the Marvelous Company for a better job. The Human Resources Division of the Marvelous Company, in accordance with

[3] For example, on a UNIX or Linux system, the hosts.equiv, .rhosts, and .shosts files would be changed appropriately or deleted.

its policy, directs the system administration to disable Nancy's account. But the system administrator is ill that day, and does not do so. Then Nancy still has the same trusted access to the company systems that she had when she was an employee. This is a classic case of an "insider."

More generally, the introduction identified three key properties that most definitions of the term "insider" are based on:

- *Access*: The insider needs some degree of access to resources. In the above definition, the subject has rights to certain resources. Those rights give it some form of access. Thus the definition covers this property.
- *Knowledge*: The insider needs to know about the resources available to it. A subject (presumably) knows it has a particular right, and what it does; hence, it knows about the resource involved. Thus, the definition covers this property.
- *Trust*: The insider must be trusted to honor the restrictions imposed on it. In the definition, P_L provides the subject with rights that it could use to exceed restrictions that P_H poses on it. The subject is trusted not to use those rights. Hence the definition covers this property.

Compare these to our definition of "insider":

- *Access*: A subject cannot employ a right without access, because if there is no access, any rights are effectively inert and cannot be used. Thus the definition covers this property.
- *Knowledge*: When a subject employs a right, the subject must know that it has the right, and must know how to use it. Thus the definition covers this property.
- *Trust*: A subject trusted not to use rights that P_L gives it, but that P_H does not, uses those rights. Hence this definition describes the betrayal of trust underlying an insider attack.

2.2 Summary

The definition of "insider" presented above is based on the exploitation of gaps in the representation of a policy at different levels. It says nothing about which insiders to fear, because most people who meet the definition of "insider" pose little to no threats. So, we first determine who poses risks by establishing the sets of users about whom we must be concerned. Once this access-based assessment is complete, we turn to an assessment of the psychological profiles of people who might try to exploit these gaps, and thereby launch insider attacks. For convenience, we refer to these people as "malicious insiders."

3 Access-Based Assessment

Given the above definition of insider, we now examine how to classify entities (subjects, usually people but possibly including autonomous agents) and resources (objects) in order to determine the risk of an insider attack, and the exposure that would result from such an attack.

We build on our observation that the differences in access granted by different layers define the insider. As an example, recall that the Oracle Policy gave Mary access to personal medical data under some conditions, but the Feasible Policy tied that access to the representation of Mary on the system, namely Mary's account. Thus, we describe a model that characterizes subjects in terms of access to resources.

Our model, the *Attribute-Based Group Access Control Model (ABGAC)* [BEG$^+$08, BEG$^+$09], groups both subjects and resources into sets defined by attributes of interest. Role-Based Access Control [FK92] is a specialization of this model, in which the subjects' attributes of interest are the job functions of the subject. To illustrate the difference, consider system administrators, a well-defined role for RBAC at most institutions. The access may differ based on time of day, which—to use RBAC—must be folded into the job function. With ABGAC, we simply define the groups by two attributes: system administration (also a role) and being in the building from midnight to 8:00AM.

We define the following components of the model.

Definition 2. A *resource pair* consists of an object and an access right.

For example, the pair (personal medical record, read) is a resource pair describing how some set of subjects can access the personal medical record. Similarly, (building, enter after midnight) describes the ability to enter a building after midnight. The objects and rights may be any objects and rights of interest; they need not be virtual resources, or exist on computers.

Often, a set of different resource pairs will be similar enough to be treated the same way. An example would be a set of printers to which jobs are assigned based on load. If a user can print a document on one printer, she can print the document on any of them. The next definition captures this.

Definition 3. A *resource domain* is a set of resource pairs.

The utility of this definition is actually greater than the above paragraph suggests. Specifically, an important characteristic of resource domains is that they are oriented towards the object and not the subject. For example, the ability to print on a printer may enable a covert channel: send the file to be printed, and see if it prints immediately. If so, that corresponds to a "1" bit. If it is queued for printing later, that corresponds to a "0" bit. In this case, an appropriate resource domain would consist of two resource pairs, one for printing a document on the printer (thereby manipulating the covert channel) and one for determining whether a specific document was printing or waiting to print (reading the covert channel).

In order to launch an attack, a malicious insider will often need access to multiple resource domains. For example, the attacker might need to read information from the resources containing personal medical information, and then save it in a local file for later reference (or transmission). Assuming the pairs (database entry of personal medical information, read) and (local file, write) are in two different resource domains, the following definition ties them together in this situation:

Definition 4. An *rd-group* is the union of resource domains.

The rd-groups combine with subjects' protection domains to define groups of users. Intuitively, associated with each rd-group is that set of users who can access all the resources in the rd-group in the manner indicated by the resource pair in that rd-group. Thus,

Definition 5. A *user group* is the set of all subjects whose protection domains contain the associated rd-group.

As an example of how these definitions fit together, consider an organization's information technology group. Among the resources it manages are desktops printers. The organization's operations rely on customer addresses, customer credit card information, and customer purchasing history (collectively called "customer data"). The company also tracks its CEO's email for legal reasons. There are two senior system administrators named Alice and Bob, and two junior system administrators, named Charlie and Eve. All system administrators have access to the customer data, but only the senior sysadmins have access to the financial information and CEO's email.

Our challenge is to find the insiders associated with the CEO's email.

From the above information, we define resource pairs based on access to the resources. The resources are the CEO's email, customer data, desktops, and printers. These can be accessed for reading, writing, or physically (as, for example, when Eve takes a printout off the printer). Thus, the resource pairs are:

(customer data, read)	(desktop, read)		(CEO email, read)
(customer data, write)	(desktop, write)	(printer, write)	(CEO email, write)
	(desktop, physical)	(printer, physical)	

Next, we define the resource domains of interest. rd_1 captures the ability to read the CEO's email, necessary for an attacker to acquire it. rd_2 and rd_3 represent the ability to save the email to a resource that can then be removed from the organization's premises.

rd_1 = { (CEO email, read) }
rd_2 = { (desktop, write), (desktop, physical) }
rd_3 = { (printer, write), (printer, physical) }

For an attacker to acquire the CEO's email, the attacker must somehow read it and then get a physical copy of it. Thus, the rd-groups of interest are:

$rdg_1 = rd_1 \cup rd_2$ = { (CEO email, read), (desktop, write), (desktop, physical) }

$rdg_2 = rd_1 \cup rd_3$ = { (CEO email, read), (printer, write), (printer, physical) }

We next look at the user groups induced by rdg_1 and rdg_2. As only Alice and Bob have read access to the CEO’s email, and all system administrators have both write and physical access to the printers and desktops, then rdg_1 and rdg_2 are clearly subsets of the protection domains of Alice and Bob. Thus, the relevant user group is composed of Alice and Bob.

At this point, the Unifying Policy Hierarchy must be considered. Throughout this example, “Alice” and “Bob” are people, but as noted earlier computer systems represent people with accounts. Therefore, anyone with access to Alice’s or Bob’s account also poses a threat. This illustrates a critical aspect of this model.

When one speaks of a “resource,” one must identify all types of access to that resource. For physical resources such as printers, this is usually straightforward.[4] For virtual resources, such as the CEO’s email, access may be obtained not only in the usual ways (such as by reading the files using the appropriate system calls), but also in less usual ways (such as reading the disk device directly and reassembling the file from the information on the disk, or reading temporary buffers in memory). Ideally, all these methods will be identified and added to the set of resource pairs.

A similar consideration holds for Alice and Bob; without loss of generality, use Alice as an example. If she has a home computer and uses that to access the company systems, it is likely that she occasionally walks away from the computer (to use the bathroom, say) without locking it or logging herself out. This gives access to her account to anyone in the house. When she takes it to a repair shop (because the company will not repair personally owned systems), the repair people may obtain her company password or be able to acquire it through nefarious means. All these entities must be considered when user groups are developed.

This leads to the question of risk. Clearly, there are too many possibilities to enable the analysis of all of them. So, some type of risk analysis mechanism must winnow out those possibilities for which the danger of attack is small.

Our approach is to determine the cost of a successful insider attack. The analyst can then determine whether the benefit of not defending against such an attack outweighs the cost of defending, taking into account the likelihood of the attack.

For purposes of discussion, let U be the set of user groups and D the set of rd-groups. Define the cost function $C\colon U \times D \to \mathbb{R}^n$, where $\mathbb{R}^n$ is the set of vectors describing the costs of compromise. This suggests two approaches.

The first approach is to minimize the impact of a successful attack. As C induces a partial order over the elements of its range, one can minimize the vector components of the value of C for any element of its domain. It may not be possible to minimize all of them simultaneously. In that case, management must decide which values are most critical to minimize.

The second approach is to minimize the number of subjects who pose a threat. This approach is appropriate when the costs of compromise are high enough that

[4] But not always. Consider that a single printer may be virtualized on a computer, in which case all aspects of access must be considered. Similarly, if the printer is mirrored so whatever it prints is also saved to a disk or printed elsewhere, access to those other resources is equivalent to access to the printer in question.

they must be defended against, yet only post hoc procedures will work. Returning to our example of personal medical information, Mary simply must have access to it to do her statistical analysis. Denying her that access means the company will not obtain critical information. Hence the cost of compromise cannot be minimized, because the personal medical information either leaks or it does not leak. Effectively, cost of compromise is a delta function, taking on the values 0 and the cost of compromise. But if the number of users who can access the personal medical information is minimized, then the number of people who could leak the information is also minimized. This may reduce the probability of the information being leaked, rather than the cost of leaking it.

The Unifying Policy Hierarchy model poses a challenge, because determining the number of subjects with access to the data requires an application of that model. An example for Alice was given above. Hence there will be unknown subjects with access to the data. The approach we take is to treat the known users as proxies for unknown users. Suppose Alice's son uses the home personal computer to visit a web site and accidentally download a malicious applet, which installs a keyboard sniffer. After his bedtime, Alice uses the home personal computer to connect to the company's computer, and the author of the malicious applet gets the company computer's network address, Alice's account name, and Alice's password. He then accesses Alice's account. Even though he is an unknown subject, the analyst can approximate the effect of his compromise by examining the effect of Alice's compromising the data.

This takes us to an examination of people: how should the analysis determine whether an individual is likely to be a malicious insider, or to give access accidentally to a malicious insider?

4 Psychological Indicator-Based Assessment

Research characterizing psychological profiles of malicious insiders focuses largely on case studies and interviews of individuals convicted of espionage or sabotage [Gel05, KG05, Par98, Dir90]. Band et al. [BCF^{+}06] and Moore et al. [MCT08] summarize findings that reveal behaviors, motivations, and personality disorders associated with insider crimes such as antisocial or narcissistic personality. Anecdotal research is post hoc, mostly derived from interviews with convicted criminals, and speculative in its predictive value. Also, assessing such personality disorders and motivations in an organization is difficult at best, and management or human resources staff may not be able to do so accurately and consistently because a typical organization does not administer psychological or personality inventory tests. Another challenge is that no studies assess and compare the prevalence of these "insider threat" predispositions with occurrence rates in the overall employee population—an important comparison needed to validate the hypothesized relationship.

Nevertheless, the body of research using case studies warrants continued efforts to address psychosocial factors. One approach is to develop predictive models that correlate the psychological profiles or behaviors that have been observed in case studies to insider crime–for example, personal predispositions that relate "... to maladaptive reactions to stress, financial and personal needs leading to personal conflicts and rule violations, chronic disgruntlement, strong reactions to organizational sanctions, concealment of rule violations, and a propensity for escalation during work-related conflicts" ([BCF$^+$06], p. 15 and Appendix G). While the factors described in the extant research reflect psychological profiles inferred from case studies and interviews by staff psychologists, an alternate approach would attempt to synthesize a set of indicators from this research and derive a set of corresponding observables that would serve as proxies for the indicators. These observables could then be extracted from a manager's evaluations of staff behavior and performance. A complementary approach is to develop instructional methods to raise managers' awareness of, and enhance their ability to detect, the warning signs of potential insider attacks. Examples of this approach are the workshops and interactive training that US-CERT [MCT08] offers, and a research and development program at the Office of the Secretary of Defense that is developing game-based methodologies to be used in this approach [DOD08].

Greitzer, Frincke and Zabriskie [GFZ09] discuss the availability and appropriateness of different types of behavioral/psychosocial data for detecting malicious, or potentially malicious, insiders. While they conclude that certain types of data monitoring would be inappropriate for reasons of privacy and ethics, they find that several sources of employee data are worthy of consideration, as summarized below.

- *Personal Information*. Generally, use of personal information within federal institutions is not likely to be appropriate or legal, no matter how useful it might be in mitigating insider threats. The employee's legal right to, and expectation of, privacy for medical records and life events such as birth, adoption, or divorce trumps the organization's desire to predict insider attacks. An employee's marital and financial problems likely could not be used either. However, such life events are known to increase stress in many individuals; signs of trouble may arise not only from such personal events as divorce or death in family, but also from work-related stress due to performance issues [BCF$^+$06].
- *Manager's assessment of employee morale*. An attentive manager should be mindful of an employee's personal situation and whether that employee's behavior reflects stress or other issues. Such attentiveness creates a supportive working environment that leads to higher employee satisfaction and less likelihood of disengagement, stress, and resulting insider attacks. Therefore, regardless of the personal life events that may underlie behavior, an attentive manager can provide judgments useful in a monitoring and analysis program. Further, an auditable trail of such information lets employees examine and correct any biased opinions as well as protecting the organization from liability.
- *Social and Organizational Information*. Unlike personal information, most work-related employee data may be used legally to observe, report, and correct inappropriate or suspicious behavior. Many employees receive annual performance evaluations that may address issues about productivity, attitude, and interpersonal skills. Recurring "rule conflicts" or "personality problems" may be ob-

served before actual insider threat events. These observations might be elements of strategies to reduce the threat of insider attacks. Many authorities suggest that managers should keep detailed records and note trends about events that result in employee disciplinary action [BCF$^+$06]. Feedback obtained from "360 degree evaluations" by associates, direct reports, and managers should be useful in assessing psychosocial factors (particularly if a manager is reluctant to provide negative feedback). Also, employee records may contain complaints by or against the employee and information related to employment applications such as education and work history.

A predictive approach enables an attentive manager to speak with stressed employees and address underlying problems in order to avert an insider attack. But a predictive approach risks potential damage that may arise from a false accusation. Adopting a predictive approach requires distinguishing between detecting *indicators* that precede a crime and detecting criminal *evidence*. In a predictive model, detection involves identifying precursors, not identifying the actual exploit. Indicators may be misleading due to the uncertainties of behavioral measures. Therefore, it is critically important to keep the human in the loop; a predictive system should be a tool for "tapping analysts on the shoulder" to suggest possible "persons of interest" on whom to focus limited resources. The underlying concept of the system is to preserve the analyst's ability to make key decisions, and responsibility for those decisions, while helping to reduce the information load, the risk, and the cost of false alarms.

Greitzer and his colleagues [GFZ09, GPK$^+$08] and Greitzer and Frincke (Chapter TBD, in this volume) describe a predictive modeling approach that combines traditional cyber security audit data with psychosocial data, to support a move from an insider threat detection stance to one that enables prediction of potential insider presence. Based on case studies that have been reported in the literature, this approach holds that the objective of predicting or anticipating potential insider threat risks is best served by using organizational/behavioral data in addition to cyber data to support the analysis. Without incorporating psychosocial data, the authors argue that prediction is extremely difficult because analysis of cyber data alone is likely to reveal malicious activity only after it has occurred.

A major focus of the predictive analysis approach is the specification and incorporation of psychosocial factors, and description or delineation of observable proxies for such factors. Greitzer and Frincke (see Table 1 in Chapter TBD, this volume) list a set of twelve such proxies, referred to as psychosocial indicators and describe a study designed to test the model against expert judgments of severity of different combinations of indicators. A particular aspect of the model development and expert knowledge acquisition that is particularly relevant to the current chapter is the nature of risk, as interpreted by the subject matter experts consulted in developing the model. Figure 1 illustrates the indicators that we identify in the model that contribute to the potential risk of an individual to become an insider threat.

An important assumption is that any such indicator is worthy of consideration *only* if the employee exhibits extremely serious or grave manifestations of the indicator. Moreover, based on interviews conducted with a limited set of human re-

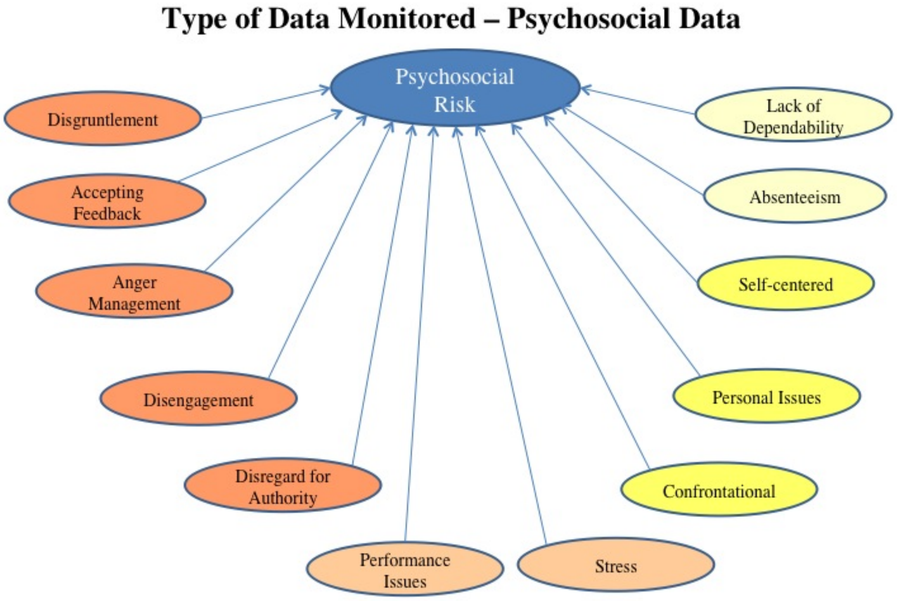

Fig. 1 Indicators that an individual is a potentially malicious insider. The darkness of each ellipse reflects the seriousness of the indicator.

sources experts, Greitzer and colleagues concluded that these twelve factors have varying levels of strength or utility in determining the risk of insider attack. The five factors shown in the darkest shade (disgruntled, accepting feedback, anger management, disengagement, and disregard for authority) are generally considered to be more serious than the other factors. As a general rule, it would take a preponderance of the other (lesser) indicators to lead one to be concerned about an individual employee, compared to perhaps only one or two of the most serious indicators.

In addition, the research conducted to date has found a fair degree of variability in the judgments of human resources experts with regard to the association of these indicators with potential risk of insider attack. Initial studies performed by Greitzer and Kangas [Gre] asked human resources experts to assign threat or risk values using a scale numbered from 0 (least) to 10 (greatest) of various combinations of the twelve indicators, ranging from only one indicator observed to between 4 and 6 indicators observed. Only a subset of possible combinations of indicators was practical in these studies because of the large number of possible combinations. It was clear that raters were not simply counting the number of indicators in arriving at their risk assessment (because there was no significant relationship between the risk values and the number of indicators identified as present). However, while there was some degree of consistency among human resource raters (with R^2 in the range of 0.4–0.5 in general), there was not uniform agreement. Some raters might have been interpreting some indicators differently than other raters. The nature of the rating task is very demanding and possibly a source of "information overload" in its own right, particularly given the number of cases that were included in the study. A different approach to this knowledge engineering task, in which cases are presented more as narrative descriptions of the identified indicators rather than the indicator labels merely being checked off, is being studied.

Another challenge in modeling the combination of psychosocial indicators and workstation and other cyber indicators is the large difference in data volume and "tempo" between psychosocial and cyber indicators. Clearly the volume of data for workstation and cyber indicators is massive, and the volume of psychosocial data is relatively sparse. This in itself is not a great challenge, but temporal differences do present difficulties for predictive models that reason about risks based on integration of cyber and psychosocial data. Of most concern is the fact that these different types of data are *asynchronous*. Cyber data is monitored in real time (although not necessarily analyzed and available immediately) and are generally available in chronological order. In contrast, psychosocial data is collected infrequently and may not be in any sort of chronological order. For example, an employee's suspicious computer activity may be monitored at a given point in time, and at some later time it might be learned that the employee has been disgruntled. At the time the suspicious activity was analyzed, that activity alone might have been insufficient to yield a risk value significant enough to justify any security action. When the late information about disgruntlement is learned, the analysis system must be able to integrate it with the threat analysis. Asynchrony in such data and analysis imposes constraints on how data is maintained, how and how long analyses are conducted, and on the way components of the predictive model share information and analysis results. A sophisticated reasoning mechanism and information architecture is required to enable such asynchronous assessments to occur.

5 Application of Risk to System Countermeasures

In this chapter, we have discussed two constructions for defining insiders: our modified version of the Unifying Policy Hierarchy, as well as the ABGAC framework. These constructions offer a more precise means of defining insiders with respect to *insiderness* than has been previously captured. The framework and hierarchy also provide a means for defining first-order protection in the form of access control.

However, defining access control rules for insiders—even more precise ones—still leads to a conflict: how do we mitigate damage (either accidental or intentional) while allowing these users to do their jobs? The problem is that a situation may arise that the developers did not, and could not, foresee. For example, consider the "doomsday scenarios" posited during the cold war in novels such as *Fail-Safe* [BW63] and movies such as *Dr. Strangelove* [Kub64], in which a control center is trying to recall bombers sent to attack the Soviet Union by accident. The aircraft either ignore what the commander has been taught is a spurious transmission from the enemy (even though, in this case, it is not spurious and from the command center), or they do not receive the recall code because a missile has destroyed the radio receiver.

These difficulties arise because security systems are relatively rigid—they serve as firewalls that allow permitted actions and block forbidden actions. This is not the way firewalls work in human society. For example, security in an industrial building is provided by security guards, access control cards, security cameras, and logging of door accesses. One security policy might be that only authorized people can enter a development lab. But in practice, doors can be propped open and access cards can be "borrowed." So, the building should have a security guard who can notice someone acting unusually—like carrying an expensive piece of equipment—and ask them for identification. A laptop being disconnected from the Internet can be detected by examining wireless access point logs and then looking at the history recorded by nearby security cameras. By more tightly interlocking the components of system security (authentication, authorization, and auditing), and by accepting that the system may be in a somewhat uncertain state, the organization runs more smoothly and efficiently without taking on undue risk.

Ideally, a system should automatically block all forbidden actions and permit all allowed actions using some variety of access control matrix [Lam74]. But Jones and Lipton showed that this is infeasible [JL75]. For example, suppose that a defense involving physical access is broached: a laptop owner walks out of her office for a few minutes, and leaves her office unlocked. If she did not lock her screen, then the failure of the physical access protection defeats the protection by authentication—someone can walk in and start using the laptop. One can try to protect against this by using an anomalous behavior detector, but such systems are notoriously imprecise, with many false positives. This would be intolerable, as it would either impede legitimate use or require too many people to analyze the reports in a timely fashion.

As with intrusion detection, there exists a conflict between security and usability. Specification-based intrusion detection [KRL97] suffers from the same problems as access control due to its "binary" nature (either allow the action, or block it). Anomaly based intrusion detection [Den87] uses statistical variations whose thresholds can be altered to flag or ignore more suspected attacks, but the consequences are that either real attacks are missed, security administrators are quickly overwhelmed with false positives, or legitimate users are mistakenly denied access. Masquerade attacks [LJ88] and insiders complicate this problem considerably.

An alternative is to find a means of protecting systems with a policy that defines both forbidden and allowed actions—as well as the actual countermeasures (e.g., blocking and allowing)—as a spectrum of possible decisions rather than as a binary decision, similarly to how we define "insiderness" above. For example, rather than denying access to read a file, allow the access but log it. Rather than denying all access to a file system, restrict access only to reading, and then to only to reading a specific partition. We call this notion *optimistic access control* because the system is *optimistic*: it allows actions up until a particular security threshold is met. Beyond this threshold the effects of the actions are unrecoverable, so only then do the controls become *pessimistic* and block the actions. Optimism does not preclude other forms of protection. For example, an anomaly detection system might inform the triggers to various countermeasures.

Merging the ABGAC and the Unifying Policy Hierarchy constructions with optimism provides a framework for managing insiders using a spectrum of non-binary countermeasures. Determining how to apply the countermeasures is still a challenge. A natural countermeasure is forensic logging of events. Even in the physical world, financial transactions have long been allowed but recorded, and entry to physical facilities is often handled in a similar manner.

We have previously discussed the importance of computer forensics and the need for better solutions [PBKM07a], in particular ones that use a systematic approach [PBKM07b] rather than ad hoc solutions. *Laocoön* [Pei07],[5] is a model of forensic logging and analysis that uses attack graphs based on intruder goals to impose a structure on log data, thereby describing the sequence of steps that take place and the data to show that these steps took place. The model can denote the set of events to be logged, prune unrelated data, aid in linking events into steps of an attack, and help bound the conditions that lead to an unusual or unexpected step in an attack. When implemented, the system can record forensic data at arbitrary levels of granularity, in standardized formats that are easy to parse. It can then correlate information, and prune away information not related to violations of the goals in question. If logging can be limited to the data represented in a model, then the analysis can be limited to only the data needed to understand the attacks of interest.

This approach leverages optimism to trigger forensic logging and thus reduce a human forensic analyst's labor while simultaneously making the system more usable. The result of merging Laocoön with optimism is the ability to apply different access control and countermeasures for different measures of risk and trust as defined in the ABGAC model. Thus, we have an improved means of conducting post mortem analyses of events. This does not mean that we can pre-classify all such events as good or bad—such is the providence of intrusion detection, not forensics—but we can use the logged data to determine what happened, and where in a chain of insider events, something may have happened which merits further investigation.

Recall that the Unified Policy Hierarchy defines ideal, feasible, practical, and run-time abstractions of a security policy. The gaps between these layers encapsulate limits on what each abstraction can specify. The ideal/feasible gap encapsulates technological limitations, the feasible/configured gap encapsulates efficiency and configuration errors, and the configured/run-time gap encapsulates implementation errors. The ideal/feasible gap is roughly equivalent to what Schneider referred to as *EM enforceability* [Sch00] and also what is used by designers of high assurance systems to develop auditing subsystems [Bis03, §24.3] based on precisely defined security policy models such as Bell-LaPadula [BL75] and Chinese Wall [BN89]. Additionally, the feasible/configured gap often relies in part on computational complexity; that is, what is efficient. These equivalences are important because EM en-

[5] Laocoön was the Trojan (an ancient detective of sorts) who recommended not letting the Trojan horse into Troy.

forceability defines the limits of what security policies can be enforced, and therefore defines the limits of which optimistic countermeasures (including logging) can be deployed. Also, while computational complexity may not define what is impossible (unlike EM enforceability), it can still guide a system administrator to define what is reasonable.

Merging the ABGAC and Unifying Policy Hierarchy constructions with optimism provides a framework for managing usable and flexible security policies on a diverse group of systems, with a diverse group of users (including insiders), using a spectrum of non-binary countermeasures. This approach leverages optimistic access control to allow policies on discrete computer systems in a way that they can be merged together, and allow computation and data transmission to continue. Optimism itself can employ a variety of non-binary countermeasures, shifting between threshold values as indicated by the risk level from the ABGAC model.

A small example will demonstrate how all this fits together.

5.1 Example

In voting and elections, insiders, usability, and security all raise issues of security [PBY09]. Consider how a voter casts a ballot in the United States. When distinguishing marks are made on paper ballots (e.g., the voter signs the ballot), many jurisdictions do not count the ballot because stray marks may communicate the identity of the voter to an auditor. Laws in the United States prevent auditors from reverse-engineering the identity of a voter, because this enables both coercion of the voter and selling votes. On electronic voting machines, the same laws apply, but the problem, and consequently enforcement mechanisms, are more difficult to define. The goal of preventing the association of voters with ballots (as with paper ballots) conflicts with the need to record audit logs in detail sufficient to enable a forensic analysis to determine if something went wrong, and if so what caused the problem. But those logs may include information such as touches on a touch screen, the number of times that a voter has selected or deselected a particular candidate, or the number of times that a voter has visited a particular screen—ideal covert channels between the voter and the auditor. There are technological solutions to limit the capacity of this channel (such as adding noise or enforcing regularity to log data) [Den80, DAH^{+}87], but there are also solutions that consider procedural steps and the nature of insiders.

Consider how to apply Laocoön to provisional ballots, which highlights the problem of insiders. A provisional ballot is cast when the poll workers cannot determine whether the voter is entitled to vote. When paper is used, the voter marks their vote on a ballot normally. She then places the ballot in an envelope and seals it. This is given to a poll worker, who places the envelope in a larger envelope, writes the name of the voter and the reason for the provisional vote on the outer envelope, and drops it into the ballot box. When the votes are counted, provisional ballots are separated from the other ballots. One election official determines whether the ballot should be counted. If so, the election official removes the inner envelope from the outer envelope and passes it to another election official. That official removes the ballot from the envelope, and puts it with the other ballots from that precinct. Modeling this situation requires an Oracle Policy to dictate the allowable actions by the election officials. But, if implemented electronically, the method required for handling *provisional electronic ballots* relies on additional Feasible and Configured Policy constraints to devise a technological method for dividing the data. Thus there is a technological gap between the Oracle and Feasible policies, representing the challenge with enforcing procedural policies.

Now, working similarly with electronic ballots and audit logs, one problem is that auditors have access to all data. Thus, one possible solution is to divide the data in a way that separates the groups of people and the data such that no technical forensic analyst can see information that describes votes that were cast, and no non-auditor can view the log data. As above, this requires an Oracle Policy to dictate the allowable actions by both the auditors and the vote counters, and gaps exist with the Feasible and Configured Policies. Though this problem cannot be eliminated, it can be reduced by enforcing a policy that takes the threat into account. The Configured Policy is defined to start at the entry to the system, end at the data (the audit logs and ballots), and place bounds on the intermediate steps. The policy then monitors those paths to address the Oracle/Feasible gap. For example, the poll workers can be monitored with video cameras, and/or a "two person rule" requiring that no single person be left with the ballots at any time be enforced. All of these countermeasures—logging, monitoring, and the two-person rule—are also optimistic: they allow activity to proceed but with an effect to limit or monitor damage due to reduced trust.

As an alternative to modifying the system to detecting insider attacks as a means of enforcing a policy to address the gap between the Oracle and Feasible policies, the system can also be modified to detect specific vulnerabilities with regard to insiders [BPH$^+$09]. For example, anything in the system that identifies the ballot uniquely, and associates it with the voter, can be eliminated. For forensic purposes, any unique item or number or code being given to the voter represents a potential vulnerability. Any time such a unique identifier is given to the voter (or by the voter), the fact that it is given should be recorded (*not* what it is, though—otherwise the forensic audit itself compromises secrecy and anonymity). Similarly, patterns in ballots can make the ballot uniquely identifiable, so ballots as a whole should be preserved. The dissemination of such unique identification can be used to trace its use later in the fault graph by looking through logs for evidence of communication of the unique identification.

Forensic evidence captured and used in the courtroom is another key where insiders and security are both important factors. For example, consider again our model of forensic logging. The model describes the data to be logged and the means to log it. But the model has its roots in technological events. What happens when the events are not attacker goals, but legal ones, such as preserving a chain of custody of evidence? Just as with voting, the technological and legal models must be merged so that any gaps between the two can be captured in the model and addressed through monitoring [PBM08]. Additionally, the notions of protecting the integrity of the data and monitoring the paths to disrupt the integrity of must be addressed, as do the entry points into the "system" (e.g., doors).

5.2 Summary

Taking technological steps to ameliorate the insider threat is challenging. Optimistic access control and rigorous applications of forensic logging provide several key benefits: a means for applying gradations of security enforcement, rather than simply binary enforcement; a means of applying security dynamically, rather than statically; and a means of providing more accurate results by gathering more information while delaying strong enforcement. The benefits are increased usability for legitimate users and more effective security against both inside and outside threats, without compromising either goal.

6 Conclusion

This chapter presents an alternate view of the insider problem. Rather than focusing on trust, the insider is defined by the ability to perform actions allowed by one policy level but disallowed at a higher policy level, or vice versa. This defines degrees of "insiderness," because different entities can be insiders in different ways. Alice may be an insider because she can read her manager's email. Bob may be an insider because he has physical access to the computer on which the email resides. In some sense, Bob's ability to access all the emails (not just those of Alice's manager) makes him more of an insider than Alice is.

This is the key point of this chapter. In some sense, the question of whether one is an insider is irrelevant. The critical characteristic is the degree of access that one has. That determines the threat, and moves us away from focusing on a (usually fairly arbitrary) definition of "insider." Instead, we examine who can access resources, and how, and what their psychological indicators are. The defensive techniques are the same for the insider and outsider who have the same degree of access.

Philosophically, this is satisfying because it simplifies the problem by eliminating a notion that is ill-defined, or defined in various ways, in the literature. Occam's razor is always satisfying to wield; whether the wielding of it in this case has simplified the analysis of the problem remains to be seen.

References

[BA04] Richard P. Brackney and Robert Helms Anderson. Understanding the Insider Threat: Proceedings of a March 2004 Workshop. Technical report, RAND Corporation, Santa Monica, CA, March 2004.

[BCF$^+$06] Stephen R. Band, Dawn M. Cappelli, Lynn F. Fischer, Andrew P. Moore, Eric D. Shaw, and Randall F. Trzeciak. Comparing Insider IT Sabotage and Espionage: A Model-Based Analysis. Technical Report CMU/SEI-2006-TR-026, Carnegie Mellon University Software Engineering Institute, December 2006.

[BEG$^+$08] Matt Bishop, Sophie Engle, Carrie Gates, Sean Peisert, and Sean Whalen. We Have Met the Enemy and He is Us. In *Proceedings of the 2008 New Security Paradigms Workshop (NSPW)*, Lake Tahoe, CA, September 22–25, 2008.

[BEG$^+$09] Matt Bishop, Sophie Engle, Carrie Gates, Sean Peisert, and Sean Whalen. Case Studies of an Insider Framework. In *Proceedings of the 42nd Hawaii International Conference on System Sciences (HICSS), Cyber Security and Information Intelligence Research Minitrack*, Waikoloa, HI, Jan. 5–8, 2009.

[BG08] Matt Bishop and Carrie Gates. Defining the insider threat. In *Proceedings of the 4th Annual Workshop on Cyber Security and Information Intelligence Research (CSIIRW)*, pages 1–3, New York, NY, USA, 2008. ACM.

[Bis03] Matt Bishop. *Computer Security: Art and Science*. Addison-Wesley Professional, Boston, MA, 2003.

[BL75] David Elliott Bell and Leonard J. LaPadula. Secure Computer System: Unified Exposition and Multics Interpretation. Technical Report EST-TR-75-306, Electronic Systems Division, Air Force Systems Command, Hanscom AFB, Bedford, MA, 1975.

[BN89] David F.C. Brewer and Michael J. Nash. The Chinese Wall Security Policy. In *Proceedings of the 1989 IEEE Symposium on Security and Privacy*, pages 206–214, Oakland, CA, May 1989.

[BPH$^+$09] Matt Bishop, Sean Peisert, Candice Hoke, Mark Graff, and David Jefferson. E-Voting and Forensics: Prying Open the Black Box. In *Proceedings of the 2009 Electronic Voting Technology Workshop/Workshop on Trustworthy Computing (EVT/WOTE '09)*, Montreal, Canada, August 10–11, 2009.

[BW63] Eugene Burdick and Harvey Wheeler. *Fail-Safe*. Dell Puiblishing, Jan. 1963.

[Car06] Adam Carlson. The Unifying Policy Hierarchy Model. Master's thesis, University of California, Davis, June 2006.

[DAH$^+$87] Dorothy E. Denning, Selim G. Akl, Mark Heckman, Teresa F. Lunt, Matthew Morgenstern, Peter G. Neumann, and Roger R. Schell. Views for multilevel database security. *IEEE Transactions on Software Engineering*, SE-13(2):129–140, February 1987.

[Den80] Dorothy E. Denning. Secure Statistical Databases with Random Sample Queries. *ACM Transactions on Database Systems*, 5(3):291–315, September 1980.

[Den87] Dorothy E. Denning. An Intrusion-Detection Model. *IEEE Transactions on Software Engineering*, SE-13(2):222–232, February 1987.

[Dir90] Director of Central Intelligence/Intelligence Community Staff Memorandum ICS 0858-90. Project SLAMMER Interim Report (U). Project Slammer is a CIA-sponsored study of Americans convicted of espionage against the United States. A declassified interim report is available at: `http://antipolygraph.org/documents/slammer-12-04-1990.shtml` and `http://antipolygraph.org/documents/slammer-12-04-1990.pdf`, April 12, 1990.

[DOD08] Accelerated learning through serious game technology, 2008. SBIR OSD08-CR8: Human Systems.

[FK92] D. F. Ferraiolo and D. R. Kuhn. Role Based Access Control. In *Proceedings of the Fifteenth National Computer Security Conference*, pages 554–563, October 1992.

[Gel05] Mike Gelles. Exploring the mind of the spy. In *Employees' guide to security responsibilities: Treason 101*. Texas A&M University Research Foundation, 2005.

[GFZ09] Frank L. Greitzer, Deborah A. Frincke, and M. M. Zabriskie. Information Assurance and Security Ethics in Complex Systems: Interdisciplinary Perspectives *(in review)*. In Melissa J. Dark, editor, *Social/Ethical Issues in Predictive Insider Threat Monitoring*. IGI Global, Hershey, Pennsylvania, 2009.

[GGG02] Robert Garfinkel, Ram Gopal, and Paulo Goes. Privacy Protection of Binary Confidential Data Against Deterministic, Stochastic, and Insider Threat. *Management Science*, 48(6):749–764, Jun 2002.

[GPK$^+$08] Frank L. Greitzer, P. Paulson, L. Kangas, T. Edgar, M. M. Zabriskie, L. Franklin, and Deborah A. Frincke. Predictive modelling for insider threat mitigation. *Pacific Northwest National Laboratory, Richland, WA, Tech. Rep. PNNL Technical Report PNNL-60737*, 2008.

[Gre] Greitzer and Kangas. (personal communication).

[JL75] Anita K. Jones and Richard J. Lipton. The Enforcement of Security Policies for Computation. In *Proceedings of the Fifth Symposium on Operating System Principles (SOSP)*, pages 197–206, November 1975.

[KG05] Joseph L. Krofcheck and Mike G. Gelles. Behavioral consultation in personnel security: Training and reference manual for personnel security professionals, 2005.

[KRL97] C. Ko, M. Ruschitzka, and K. Levitt. Execution Monitoring of Security-Critical Programs in Distributed Systems: a Specification-Based Approach. In *SP '97: Proceedings of the 1997 IEEE Symposium on Security and Privacy*, page 175, Washington, DC, USA, 1997. IEEE Computer Society.

[Kub64] Stanley Kubrick. Dr. Strangelove or: How I learned to stop worrying and love the bomb. Distributed by Columbia Pictures, Jan. 1964.

[Lam74] Butler W. Lampson. Protection. *ACM Operating Systems Review*, 8(1):18–24, January 1974.

[LJ88] Teresa F. Lunt and R. Jagannathan. A Prototype Real-Time Intrusion-Detection Expert System (IDES). In *Proceedings of the 1988 IEEE Symposium on Security and Privacy*, pages 59–66, Oakland, CA, April 18–21, 1988.

[MCT08] Andrew P. Moore, Dawn M. Cappelli, and Randall F. Trzeciak. The "Big Picture" of Insider IT Sabotage Across US Critical Infrastructures, 2008.

[NS05] Peng Ning and Kun Sun. How to Misuse AODV: A Case Study of Insider Attacks Against Mobile Ad-Hoc Routing Protocols. *Ad Hoc Networks*, 3(6):795–819, Nov. 2005.

[Par98] D.B. Parker. *Fighting computer crime: A new framework for protecting information*. John Wiley & Sons, Inc. New York, NY, USA, 1998.

[Pat03] J. Patzakis. New Incident Response Best Practices: Patch and Proceed Is No Longer Acceptable Incident Response. Technical report, Guidance Software, Pasadena, CA, Sept. 2003.

[PBKM07a] Sean Peisert, Matt Bishop, Sidney Karin, and Keith Marzullo. Analysis of Computer Intrusions Using Sequences of Function Calls. *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 4(2):137–150, April–June 2007.

[PBKM07b] Sean Peisert, Matt Bishop, Sidney Karin, and Keith Marzullo. Toward Models for Forensic Analysis. In *Proceedings of the Second International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE)*, pages 3–15, Seattle, WA, April 2007.

[PBM08] Sean Peisert, Matt Bishop, and Keith Marzullo. Computer Forensics *In Forensis*. In *Proceedings of the Third International IEEE Workshop on Systematic Approaches to Digital Forensic Engineering (IEEE-SADFE)*, pages 102–122, Oakland, CA, May 22, 2008.

[PBY09] Sean Peisert, Matt Bishop, and Alec Yasinsac. Vote Selling, Voter Anonymity, and Forensic Logging of Electronic Voting Machines. In *Proceedings of the 42nd Hawaii International Conference on System Sciences (HICSS), Digital Forensics – Pedagogy and Foundational Research Activity Minitrack*, Waikoloa, HI, Jan. 5–8, 2009.

[Pei07] Sean Philip Peisert. *A Model of Forensic Analysis Using Goal-Oriented Logging*. PhD thesis, Department of Computer Science and Engineering, University of California, San Diego, March 2007.

[Sch00] Fred B. Schneider. Enforceable Security Policies. *ACM Transactions on Information and System Security (TISSEC)*, 3(1):30–50, February 2000.

[Sch02] E. Eugene Schultz. A Framework for Understanding and Predicting Insider Attacks. *Computers and Security*, 21(6):526–531, Oct. 2002.