Combatting Insider Threats

Peter G. Neumann

Abstract Risks from insider threats are strongly context dependent, and arise in many ways at different layers of system abstraction for different types of systems. We discuss various basic characteristics of insider threats, and consider approaches to the development and use of computer-related environments that require systems and networking to be trustworthy in spite of insider misuse. We also consider future research that could improve both detectability, prevention, and response. This chapter seeks to cope with insider misuse in a broad range of application domains—for example, critical infrastructures, privacy-preserving database systems, financial systems, and interoperable health-care infrastructures. To illustrate this, we apply the principles considered here to the task of detecting and preventing insider misuse in systems that might be used to facilitate trustworthy elections. This discussion includes an examination of the relevance of the Saltzer-Schroeder-Kaashoek security principles and the Clark-Wilson integrity properties for end-to-end election integrity. Trustworthy system developments must consider insider misuse as merely one set of threats that must be addressed consistently together with many other threats such as penetrations, denials of service, system faults and failures, and other threats to survivability. In addition, insider misuse cannot be realistically addressed unless significant improvements are made in the trustworthiness of component systems and their networking as well as their predictably trustworthy compositions into enterprise solutions — architecturally, developmentally, and operationally.

1 A Contextual View of Insiders and Insider Threats

We consider a broad spectrum of problems relating to insider threats, along with techniques for preventing, detecting, diagnosing, and understanding specific ex-

Peter G. Neumann
Principled Systems Group, Computer Science Lab, SRI International, Menlo Park, CA 94025-3493 USA, e-mail: neumann@csl.sri.com.

C. W. Probst et al. (eds.), *Insider Threats in Cyber Security,*
DOI 10.1007/978-1-4419-7133-3_2,

ploits – within the context of overall system and application trustworthiness. For present purposes, an *insider* is simply a system user who is granted and can use certain privileges. Intuitively, insider threats involve such users misusing those privileges, potentially causing violations of confidentiality, data integrity, system integrity, system survivability, identity management, accountability, denials of service, and anything else relating to abuses of trust. For generality, we allow that the 'user' could also be some sort of human surrogate or other computer entity – process, agent, or system – presumably but not necessarily acting on behalf of specific human users.

The meaning of 'insider' is strongly dependent on the application context. Furthermore, the concept is relative to the privileges available (either given or somehow otherwise acquired), which may be hierarchically layered or otherwise granted. For example, in health-care applications, a doctor may have access to personal and medical data of many patients, whereas a patient may be able to access only certain portions of his or her own data—and nothing else. Administrators, insurance companies, pharmacies, and employers might have varied restricted access, as might nurses, third-party staff such as offshore transcription services, and surprisingly many others. In addition, banks, credit bureaus, data clearing houses, and many other people and institutions may have legitimate access to some medical information and related personal data. Furthermore, researchers and others might also have certain access rights, but possibly only with some sanitization or anonymization of personal data that might attempt to mask personal identities. Deborah Peel (patientprivacyrights.org) estimates that about 4 million people in the U.S. have some sort of access to medical record information—and thus they can all be considered as insiders in some respect.

Distinctions between insiders and outsiders can be slippery, particularly in the absence of system security that effectively reduces the likelihood of penetrations and external denials of service. Similarly, distinctions between malicious acts and accidental events are often misleading – in that events occurring accidentally could often be triggered intentionally, and adverse events may occur unbeknownst to the inadvertent triggerer. Clearly, a system often needs to be protected against both kinds of events. For example, most life- or safety-critical systems must address use and misuse by both insiders and outsiders in an architecturally integrated way, and enforceable policies must be established and consistently implemented. These subtleties can be quite significant in assessing how we should approach insider misuse within the more general context of system and network trustworthiness.

There are many different definitions of insiders, some of which are in conflict with one another. For example, some definitions exclude outsiders who have usurped privileges of insiders, whereas other definitions include those outsiders who have effectively gained privileges of insiders — perhaps due to inadequate system and process assurance. The latter individuals are herein referred to as *outside-inners*. Some definitions fail to define either 'insider' or 'outsider', simply implying that one is not the other. A definition attributable to a National Research Council study report is "a person who is allowed inside the security perimeter of a system and consequently has some privileges not granted outsiders." That definition suffers from defining one

type of user in black-and-white disjunction—as the opposite of the other—where neither is adequately defined and grey areas are ignored. It also suffers from the fact that today's computer operating system and networking infrastructures are vulnerable, and enable outsiders to carry out destructive integrity attacks almost as if they were insiders. It further suffers from the reality that there is typically no single security perimeter; indeed, at each layer of abstraction there may be different perimeters for different aspects of trustworthiness that satisfy different subsets of the set of given requirements, where compromises of one perimeter may directly or indirectly also compromise the integrity of other perimeters. Note that the different requirements for security, reliability, survivability, human safety, usability, and so on may be conflicting, and the respective perimeters of trustworthiness may actually be nonoverlapping – which suggests that the notion of an insider is actually multidimensional with respect to the different requirements and different types of applications. Furthermore, a supposed perimeter of trust may actually encompass the entire system, its total operational environment, all of its potential users, and perhaps the entire Internet – particularly in badly designed systems. The approach taken here generalizes other formulations into a multidimensional framework.

Misuse implies use that is contrary to expected operational behavior. However, that is too much of an oversimplification. In practice, the concept of *misuse* is meaningful only with respect to a policy that defines what usage is acceptable and what is not. Unfortunately, a basic gap exists between use that is intended to be acceptable and use that is actually possible (*e.g.*, [3]). Within that gap, subgaps exist – for example, between what is possible (because of design flaws and implementation bugs) and what is actually authorized, as well as limitations that result from inadequate granularity and expressiveness of access controls.

A useful distinction exists among three alternative misuse cases with respect to any particular layer of abstraction: *compromise from outside, compromise from within,* and *compromise from below.* Compromise from outside represents activities of outsiders relative to that layer of abstraction, but as noted above can result in outside-inners. Compromise from within represents actions of insiders with respect to that layer. Compromise from below represents actions of insiders with respect to some lower layer who can compromise the integrity of higher layers.

One historical example of how this trichotomy fits into the constructive architecture of an operating system is given by the Multics hierarchical ring structure [20], which provided eight layers of protection. That mechanism essentially ensured that each ring could not be compromised from higher layers, while the lowest-layer rings were under stringent development and operational configuration control. This mechanism also provided enhanced system survivability.

The determination of who is an insider and who is an outsider is also relative to what boundaries might be assumed to exist. That is, an insider at one layer may be an outsider with respect to a lower layer or with respect to a different perimeter. For example, someone who can manipulate bits in memory or secondary storage using hardware diagnostic tools might be called a hardware insider. Someone who can manipulate operating system parameters because she has authorized use of certain root privileges would be considered an insider with respect to the operating system.

Someone who can tamper with a browser because he is the maintainer of Web facilities would be considered an insider with respect to the webware. A similar analysis is given by Matt Bishop *et al.* [2]. For a recent compendium of articles on insider misuse, see [23]. See also the other chapters in this book, as well as Section 4 of the 2009 Roadmap for Cybersecurity Research [10], devoted to combatting insider threats. For some pithy examples of evident misuse, see the Appendix to this chapter (excerpted from [16]).

2 Risks of Insider Misuse

To understand the problems, we need to explore various kinds of insiders further, as well as types of misuse, threats, vulnerabilities, risks, and knowledge and experience that might be applied.

2.1 Types of Insiders

Differences among users may involve physical presence and logical presence. For example, there may be logical insiders who operationally are physically outside, and physical insiders who are logically outside. For present purposes, we consider both logical and physical insiders.

Clearly there are different degrees of logical insiders, relative to the nature of the systems and networks involved, the extent to which authentication and authorization are enforced, and the exact environment in which a user is operating at the moment. A user in one operational domain may be an insider at one moment and an outsider otherwise, with respect to each of the various so-called contexts noted above.

For example, if a system supports multilevel security (or multilevel integrity [1]), or even some form of multilevel availability or multilevel survivability [12]), then the existence of compartments suggests that a user can be an insider in one compartment but an outsider in another compartment, or an insider at Top Secret but an outsider with respect to all compartments. In that a user may operate at different levels and compartments at different times, the concept of insider is both temporal and spatial. In some sense, all users of a single-level Top-Secret system could be called insiders with respect to confidentiality, although they would appear to be outsiders relative to those others who were cleared into a particular Top Secret compartment. Similarly, a user could be an insider with respect to multilevel security and an outsider with repect to multilevel integrity. Thus, everything is relative to the frame of reference – what the user is trusted to be able to do, what privileges are required, what data or programs are being referenced, and whether the user authentication is strong enough to ensure that user identities are not spoofed.

With respect to conventional operating systems, database management systems, and applications functioning as single-level systems (even if lumping multilevel in-

formation into a single level, typically called *system high*), there are typically ordinary insiders who have passed the login authentication requirements and have been granted certain limited access rights. In addition, there are special users who are authorized to act as a superuser or otherwise be allocated extra-powerful privileges. In contrast, Trusted Xenix [7]) was a system in which the superuser privileges were extensively partitioned, where no one user holds all of the privileges, and where the granted privileges are insufficient to gain possession of all other privileges. (The iterative closure of static privileges augmented by privilege-changing privileges must also be considered whenever we consider what privileges are actually attainable by a given user or group of collaborating users.) In that rather ideal case, we might have no complete insiders, but many different types of relative insiders. Unfortunately, in the absence of meaningfully secure systems and fine-grained access controls that are properly defined, properly implemented, and properly administered, that ideal is still a fantasy.

Thus, we are confronted with a wide variety of insiders that is inherently multidimensional. Here, we tend to consider insiders somewhat loosely, avoiding fine nuances among different kinds of insiders. We assume that relative to a particular computational framework, insiders are users who have been authenticated to operate within that framework. However, where appropriate, we qualify that to include reference to the authorized privileges that may be specifically associated with a particular instance of an authenticated user (such as a system administrator).

2.2 Types of Insider Misuse

Along with the variety of insiders is associated a variety of types of insider misuse. One immediate categorization involves user intent, as in intentional versus accidental misuse (noted above). Even among intentional misuse, there is a wide range of possible actions – from outright malice to relatively benign annoyance, with many degrees in between. However, whether the cause is accidental or intentional is sometimes not clear.

A second categorization involves the evidential nature of the misuse, that is, whether the misuse is intended to be detected or hidden. System and network denials of service may be overt, in that they are readily obvious once they are enabled. However, stealthy Trojan horses that act as sniffers or that quietly leak information are typically intended to be covert, and may be intended to remain undetected as long as possible.

Although the focus here is primarily on intentionally malicious misuse, it is generally unwise to ignore accidental misuse. For example, the apparent success of what might be considered accidental but tolerated misuse could easily inspire subsequent malicious misuse. Furthermore, it is generally unwise to ignore stealthy forms of misuse. To the extent that detecting accidental misuse can be dealt with by the same mechanisms that are used for intentional misuse, accidental misuse need not be treated separately. Similarly, to the extent that stealthy misuse can be dealt with by

the same mechanisms that are used for more obvious misuse, stealthy misuse need not be treated separately – apart from possibly additional means of detecting it. Because seemingly accidental misuse may in fact be intentional misuse in disguise, stealthy misuse can be extremely dangerous; as a consequence, it is potentially risky to ignore any particular mode of insider misuse. Nevertheless, responses may differ depending on whether the cause is deemed to be accidental or malicious.

3 Threats, Vulnerabilities, and Risks

There are clearly differences in the nature of the various threats, especially in the possibility of outside-inners. Although an insider might conceivably have greater knowledge of the environment, and may thereby present greater threats, the differences between insider threats and outsider threats are often not stereotypically characterizable. If a system has meaningful authentication, many of the outsider threats can be made much less risk-prone, whereas most of the insider threats clearly remain. Also, firewalls that are well-designed, well-implemented, and well-configured can help somewhat, but today are also largely vulnerable to many attacks (such as active pass-through attacks using http, JavaScript, Active-X, PostScript, other forms of executable content, cross-site scripting, SQL injection, and bogus URLs in phishing attacks). The availability of meaningful additional authentication for insiders could be useful in inhibiting masquerading. With extensive monitoring, robust authentication may also help discourage misuse – especially if the identity of the perpetrator could be established and traced reliably. This may be especially relevant to insider misuse, if the true identity of the apparent user can be unequivocally determined (subject to exploitations of operating-system vulnerabilities – including manipulations of audit trails).

It is of course useful to consider insider threats in their own right. In today's systems, insider vulnerabilities and outsider vulnerabilities are both out of control. Serious efforts are needed to improve security and reliability of system and networks, and indeed to improve the overall survivability in the face of a wide range of adversities. With good external security in critical systems, insider risks may be much more serious than outsider risks. However, meaningfully precise access control policies and meaningfully secure fine-grained access controls may reduce the damage from the insider threats.

Table 1 itemizes some of the threats that appear to differ from outsiders to insiders. It ignores threats that are common to both outsider and insider perpetrators, such as carrying out personal attacks on individuals or corporations through an anonymous e-mail remailer, sending spams, creating monster viruses from a toolkit, creating risky mobile code, tampering with existing mobile code, intentionally crashing a system or component (although there are potentially mechanistic differences among insiders and outsiders), and so on. Nevertheless, to simplify the table, outside-inners are logically considered as outsiders unless they are knowledgeable enough to appear indistinguishable in something like a Turing-test sense from the insiders as

Table 1 Threats to Security

Attribute	Outsiders	Insiders
Authentication	Penetrations, attacks on PKI/authentication infrastructures, war dialing	Misuse of intended authority by over-authorized users, usurpation of superuser access and root keys
Authorization	Unprivileged exploitation of inadequate controls	Privileged manipulation of access controls
Confidentiality	Unencrypted password capture or compromise of encrypted passwords	National security leaks and other disclosures; access to crypto keys(!)
Integrity	Creating Trojan horses in untrusted components, Word macro viruses, untrustworthy Web code, in-the-middle attacks	Inserting Trojan horses or trapdoors in trusted (and untrusted) components; altering configurations, schedules, and priorities
Denials of Service	External net attacks, flooding, physical harm to exposed equipment	Disabling of protected components, exhaustion of protected resources
Accountability	Masquerading, DoS attacks on accounting infrastructures	Hacking beneath the audit trails, altering audit logs, compromising misuse detection
Other misuses	Planting pirated software on the Web	Running a covert business, insider trading, resource theft

whom they are masquerading – as might be the case with disgruntled recent ex-employees. More realistically, the indistinguishability may be more like the ability of an outsider to masquerade as an insider if just a little social engineering is all that is required.

3.1 Relevant Knowledge and Experience

Some differences are likely to exist in the knowledge available, the knowledge required, and the knowledge actually used in perpetrating various types of misuse. Understanding these differences may be useful in analyses associated with detected misuses.

For example, insiders might seem to have greater knowledge of what to look for in terms of sensitive information and particularly vulnerable programs in which to plant Trojan horses—including especially system administrators. In system-high systems, legitimate insiders are already likely to be gratuitously granted information to which they do not necessarily need access. In compartmented multilevel-secure systems, users would have clearances associated with authorizations, although that works both ways: a user not entitled to access a particular compartment is effectively an outsider with respect to that compartment, and indeed may not even know of the

Table 2 Knowledge Gained and Used

Outsiders	Ordinary Insiders	Privileged Insiders
Direct info and inferences from web info (such as penetration scripts), help files, social engineering; chats/ BBoards helpful	Experience gained from normal use and experiments; familiarity with sensitive files, project knowledge; collusion easy	Deep knowledge from experience; ability to change and abuse privileges; ability to create invisible accounts; collusion even easier

existence of the compartment if the system is properly implemented and operational procedures are properly enforced. However, users cleared into that compartment have an enormous advantage for potential misuse over users who are not – assuming isolation is suitably enforced and operationally deployed.

Table 2 makes a distinction among outsiders, ordinary insiders, and specially privileged insiders such as highly trusted system administrators, recognizing that we are lumping together users with common logical characteristics.

3.2 Exploitations of Vulnerabilities

There is a likelihood that an experienced insider can operate close to normal expected behavior (especially if engaged in a long-term effort at what in terms of a anomaly detection system would resemble statistical-profile retraining), which would be more difficult to detect. This increases the need for a variety of analysis techniques and correlation (see below).

Today, we have pervasive deficiencies in authentication, authorization, accountability, operating system security, network security, and intelligently deployed access controls. Given the absurdly poor existing state of the practical art of defensive security, the differences among exploitations by outsider and insiders may be less relevant than they would be in the presence of stronger security.

Insider exploitations might conceptually be thought of as somewhat simpler to manage in the presence of stronger system security. Enormous benefits could result from intrinsically better operating system security, network security, pervasive encryption, user authentication, and well-managed authorization. One of those benefits would be that detection and response could be much more precisely targetted, rather than having to address all security vulnerabilities. However, insider threats would still represent a significant problem, because many of those threats would not have been eliminated.

Table 3 Potential Severity of Risks Incurred

Outsiders	Ordinary Insiders	Extra-Privileged Insiders
Very serious in badly designed and poorly implemented systems, perhaps less serious with good user authentication and good auditing	Potentially very serious unless strong separation of roles, MLS, and fine-grained access controls; beware of system-high systems	Extremely serious, even with strong separation of roles and separation of privilges, MLS levels and compartments; misuse of multipurpose root privileges is inherently risky

3.3 Potential Risks Resulting from Exploitations

The potential risks may vary significantly from outsiders to outside-inners to ordinary insiders to highly privileged system administrators. However, it is in itself risky to give too much credence to these differences, because of several factors:

- When the security of systems, servers, firewalls, and networks is weak, both outside-inners and insiders can cause serious harm.
- Some outsiders such as terrorists may have highly visible major havoc in mind. Alternatively, outside-inners might try to mask the existence of clandestine Trojan horses, trapdoors, and other system aberrations. In general, exactly the same situation applies to insiders, although the stealthy route would generally be more likely in the presence of strong authentication that hinders insider masquerading and provides a fairly clear chain of evidence. Each type could create highly undetectable effects or massive disasters entailing major risks.
- Measures of risk are highly speculative, and strongly dependent on the application environment. (One man's feat is another man's poison.)

4 Countermeasures

Where possible, prevention is vastly preferable to detection and attempted remediation (although cases of insider misuse generally exist in which prevention is inherently difficult). For example, the Multics system architecture (see [5] and http://www.multicians.org/) stressed the importance of prevention by isolating privileged execution domains from less-privileged executions, isolating one user from another while still permitting controlled sharing (via access-control lists, access-checked dynamic linking, and dynamic revocation, as well as user-independent virtual memory), and using some sensible software-engineering concepts. Use of some of the Saltzer-Schroeder [22] security principles is directly relevant to minimizing insider misuse. The most obviously applicable principles here are separation of privileges, allocation of least privilege, and open design. In addition, ease of use (generalizing Saltzer and Schroeder's psychological acceptability) could provide incen-

tives for insiders to avoid the excuse of security being too complicated, which otherwise often results in the creation of unnecessary vulnerabilities. These and other principles are discussed further in the context of election systems in Section 7.

If there is no meaningful security policy, then the task of detecting and identifying deviations from that policy is not meaningful. If there is no fine-grained context-sensitive prevention in systems and networks, then even if there were a meaningful security policy, it would be difficult to implement it. With respect to insiders, enterprises operating within a system-high approach suggest that insider misuse is ill-defined – in the sense that everything may be permitted to all authenticated users. Thus, to have any hope of detecting insider misuse, we first need to know what constitutes misuse. Ideally, as noted above, it would then be much better to prevent it rather than to have to detect it after the fact.

The absence of rigorous authentication and constructive access controls tends to put the cart before the horse. For example, what does *unauthorized use* mean when almost everything is authorized? Recall the Internet Worm of 1988, which was an outside-inner attack. Robert Tappan Morris was prosecuted for *exceeding authority*; yet, no authorization was required to use the `sendmail debug` option, the `finger` daemon buffer overflow, the `.rhosts` mechanism, and copying an encrypted but then unprotected password file. This may have been *misuse,* but was not *unauthorized misuse*. The same issues arise with recent malware.

Finer-resolution access controls are of particular interest in minimizing insider misuse, such as fine-grained access-control lists and fine-grained roles, separation of duties, compartmentalized protection for integrity, and attribute-based encryption. Some of those controls date back to the Multics file system [6] in 1965, and have been the subject of refinement and alternative approaches ever since. Past work on strongly typed, hardware-tagged, capability-based systems (*e.g.*, [17]) could also considerably reduce opportunities for insider misuse; in such systems, access is impossible for anything for which an appropriate capability is not available. Of course, various forms of multilevel security and multilevel integrity could also help to narrow down the possibilities for insider misuse, albeit with the associated administrative baggage. However, all these approaches create further usability issues and administrative complexity.

Although relevant not specifically to insider misuse, but more generally to the development of trustworthy systems, several other thrusts are also of interest here – for example, a report on how to develop principled assuredly trustworthy composable architectures [13] and subsequent reflections on trustworthiness [14]. Also somewhat relevant is a paper by Paul Karger [8] that applies access controls to programs. That approach might be interesting in controlling the extent to which insider-introduced malware (particularly Trojan horses) could be blocked, assuming that the insider is not privileged to alter the access controls. Approaches to sandboxing have similar goals, limiting what would-be malware might be able to do.

In summary, better policies are needed establishing what threats are relevant, and what constitutes misuse. Better user authentication could not only prevent intruders from gaining insider access, but could also provide positive identification of insiders that might diminish their ability to masquerade as other insiders and to

otherwise hide their identities. Authorization is typically not fine-grained enough, which limits the effectiveness of access controls and misuse detection. Oversight and accountability are essential. Monitoring tools need to address detection of insider misuse.

4.1 Specification of Sound Policies for Data Gathering and Monitoring

Commericial products for misuse detection tend to assume a collection of known vulnerabilities whose outsider exploitations are associated with known policy violations. Existing products tend to be aimed primarily at penetrators and intrusion detection, and are not easily applied to detecting insider misuse. Policies for insider misuse tend to be strongly application-domain specific, and should dictate what is to be monitored, at what layers of abstraction. Thus, it is essential to have a well-defined policy that explicitly defines insider misuse, or else a policy that explicitly defines proper behavior and implicitly defines insider misuse by exclusion.

A much better understanding of the application domain is needed for monitoring users for potential insider misuse. Also, more detailed data may need to be collected. Furthermore, when someone is suspected of devious behavior, it may be desirable to go into a fine-grain analysis mode, although that has its own serious potential privacy problems.

Today, commercial systems for misuse detection generally rely on system audit trails, network packet collection, and occasionally physical sensors for their inputs. Other sources of input data are necessary for detecting insider misuse, including detailed database and application logs. In either case, the analysis systems need to obtain some knowledge of the perpetrator if they are to trace the detected misuses back to their initiators. In closed environments, there can be much better user authentication than in open environments, although masquerading is still possible in many operating systems and application environments. Whenever that is the case, the actual choices of data to be gathered for insider-misuse detection tend to differ from that of intrusion detection. However, the existence of logical insiders who are physically outside and logical outsiders who are physically inside may make such distinctions undesirable – suggesting that making the assumptions (such as there are no outsiders, or there is no insider misuse) is unwise. The necessary use of encryption for stored information in highly sensitive systems may also complicate the gathering of information on potential insider misuse, and necessitate capture of unencrypted content – which raises serious some serious security and privacy concerns.

4.2 Detection, Analysis, and Identification of Misuse

In the absence of good prevention, it is of course desirable to detect known defined types of misuse (*e.g.*, through rule-based detection) as well as otherwise unknown types of anomalous misuse (*e.g.*, seemingly significant deviations from expected normal behavior). The latter type of detection could be particularly important in identifying early-warning signs of misuse. o Because there are potential differences in the data that may need to be collected, there may be some differences in the approach to detection of misuse among the different types of misuse, depending on the relative roles of insiders and insider misuse. If insiders can exist only within local confines (for example, as in the case of a multilevel security compartment in a system with no remote users and no Internet connectivity), it may be unnecessary to collect packets and other network data – which themselves constitute potential security and privacy risks. If privileged logical insiders are also able to access their systems remotely (for example, using encrypted programs such as `ssh` from outside) and are in some sense then indistinguishable from outsiders at least geographically or from their external Internet presence, then networking data may also be relevant. Clearly, the presence of strong authentication has an impact on carrying out insider misuse detection.

Similarly, there may be differences in data retention requirements among misuse-detection system. If the intent is to gather sufficient information to prosecute insider misusers, then the situation is quite different from detection whose aim is merely to detect the presence of misusers so that other extrinsic methods (such as wiretaps, cameras, and physical surveillance) can be invoked. (These differences may also apply to outsiders – although the relative priorities are likely to be different.) In general, long-term retention of raw audit logs and of digested (analyzed) data is recommended.

The marketplace for intrusion detection is aimed primarily at detecting known attacks by outsiders – for example, with signature-based expert systems seeking to detect exploitations of known vulnerabilities. The idea of rapidly deploying an analysis system is meaningful for a given firewall, or for a given operating system, or for a given application for which a set of rules have already been written. However, insider attacks tend to be much more domain specific; insider analysis requires more detailed analysis of the threats and risks, some skilled implementation of rules, judicious setting of statistical parameters, and some further work on analysis of the results. These rules must also take into account discrepancies between the actual access controls and what kind of access is considered appropriate. Once again, the extent to which explicit fine-grained access controls can be defined and enforced has a direct influence on what kinds of insider misuse need to be detected. Thus, new approaches are needed to better address the insider threats. This is not a straightforward off-the-shelf installation process.

In a multilevel compartmented system/network environment, in which there are presumably no outsiders and in which the insider threat predominates, monitoring and analysis take on multilevel security implications, with many opportunities for covert channels and inferences. Monitoring can be done compartmentally, but aggre-

gation, higher-level and cross-compartment correlation on an enterprise-wide basis present serious potential multilevel security problems.

More emphasis is needed on not-well-known forms of insider misuse, on interpretation of detected anomalies, and hierarchical and distributed correlation. Much more emphasis is needed on tools to aid in the deployment and configuration of analysis tools for domain-specific applications. Serious effort might also be devoted to multilevel-secure analysis (and response) in contexts in which MLS systems might be important. Procedural and psychological approaches are likely to predominate, much greater awareness of the threats and risks of insider misuse is likely to drive new approaches.

An enormous risk exists relating to false accusations of supposed culprits despite the inability to carry out any definitive traceback to host systems and individual logins. This problem is exacerbated by the long-time retention and undeletable mirroring of erroneous data throughout the Internet, and the difficulties in correcting widely disseminated erroneous information (as was the case in the 'Swift Boating' of John Kerry in the 2004 U.S. Presidential election campaign).

4.3 Desired Responses to Detected Anomalies and Misuses

In some cases of outsider attacks (particularly denials of service), it is more important to stave off the attacks than to let them continue. In other cases, it may be appropriate to let the attacks continue but to somehow confine their effects (as in the case of sandboxing and honeypots). A similar range of responses exists for insiders. In some cases of insider misuse (particularly where the perpetrator has been identified and prosecution is anticipated), it may be particularly important to detect the misuse, to allow it to continue (perhaps under special system constraints and extended data gathering such as key-stroke capture), and monitor it carefully – without giving away the fact that detailed surveillance is being done.

Thus, there are clearly differences in the desired responses that may be considered once misuses have been detected. However, the full range of possible responses may also be applicable to both insiders and outsiders – although possibly in different degrees in the two cases. In any case in which continued misuse is allowed, serious risks exist that undetected contamination and other integrity problems may occur and remain subsequently. This must be factored into any dynamic strategies for real-time response to detected misuse.

5 Decomposition of Insider Misuse Problems

This section looks at insider misuse in the context of the bigger picture of security. It considers development and operation, and the effects of those issues on misuse

detection. It also specifically addresses the importance of user profiling and the desirability of extending it to include psychological factors.

5.1 Stages of Development and Use

Each of the stages in system development and use has its own problems and potential vulnerabilities that must be considered with respect to insider threats. Various system development methodologies address some of these problems. (For example, see [13, 14].)

- Requirements: Insider threats are often ignored, even in highly sensitive standalone systems that tend to run at system high. Security, reliability, survivability requirements are often short-sighted, incomplete, or unsatisfied in system developments.
- System architecture and design: Many commercial systems are short-sighted, hindered by their needs for backward compatibility with earlier nonsecure systems and networking, and handicapped by a serious lack of commitment to robustness. These systems are primarily aimed at low-hanging fruit, or else must have their rule bases updated frequently to keep up with the malware du jour. In contrast, the research community has progressed significantly in recent years. For example, [18] discussed some of the research directions as well as the desirable characteristics of future systems for anomaly and misuse detection that could be applicable to insider misuse as well as intruders.
  System design must encompass authentication and authorization that is relevant to the insider threats. Authentication can seriously impede outsiders, but not if the systems rely on fixed passwords (especially if those passwords are transmitted unencrypted, or are replayable, or used in single-signon applications across boundaries of trustworthiness). Authorization is typically not fine-grained enough, and limits the effectiveness of access controls and misuse detection. Boundary controllers may be useful, but are typically vulnerable to denial-of-service attacks.
- Implementation: Many serious security-related implementation flaws persist. As just one example, buffer overflows continue to appear despite years of knowledge of their origins and the ensuing risks. Serious attention to software engineering discipline is sorely lacking.
- Operation: Even with ideal system development, accountability is fundamental to monitoring and analyzing insider misuse. It needs to be tightly coupled with strong authentication and access controls.
- System administration: System administrators are hard-pressed to cope with security flaws, security patches, and administering misuse detection. System design must include more effective mechanisms to aid admins, such as self-configuring detection and analysis tools that can be easily tuned to the threats of greatest significance according to the perceived risks.

- High-level enterprise management: Existing analysis techniques do relatively little for enterprise-wide monitoring and correlation across multiple network and system platforms.
- System support: Although vendors may not always have the customer's best interests at heart, many customers seem to be overly naive. In an insider misuse workshop on 12 April 1999, Ed Amoroso mentioned AT&T's experience with Net Ranger (later acquired by Cisco). When he and his colleagues finally tried to install it, months after receiving the CD, they discovering that certain files were missing from the installation CD. When they complained to Cisco, the Cisco folks indicated they had never before heard about this problem; apparently no one had ever successfully installed it!
- Monitoring: Data from appropriate audit trails (operating systems, DBMSs, applications) and network data (packets, network management information) needs to be hierarchically abstracted, heterogeneous, diversified, and collected only where needed for potential privacy reasons. Existing data sources tend to have little abstraction and contain huge quantities of relatively useless information.

Detection of insider misuse (and especially hitherto unrecognized threats) deserves much greater attention, using a wide variety of approaches. Historically, two early SRI efforts specifically aimed at detecting insider misuse are worth noting. The first, begun in 1983 for the CIA, sought statistically significant deviations from the expected normal behavior of IBM mainframe users represented in user profiles. Later, in the 1990s, a variant of the NIDES system was considered for the classified FBI Field Office Information Management System (FOIMS), using a rule-based expert system applied to database logs. Neither system was actually deployed, perhaps because each institution decided that insider threats were realistically minimal! (Note the first bulleted item in the Appendix.)

Misuses and anomalies that have been detected require some abstraction in their reporting and diversity of sites to which reporting occurs. (For example, EMERALD [19, 18] allowed for a wide variety of destinations, including passing the results to higher-layer instances of EMERALD and directly to system administrators.) Correlation is needed across wider scopes across multiple target systems and networks, and across multiple analysis platforms. Much more sophisticated and understandable interpretation of analysis results is essential at varying layers of abstraction.

5.2 Extended Profiling Including Psychological and Other Factors

It is clear from the above discussion that detecting insider misuse must rely heavily on user profiling of expected normal behavior (although some use can be made of application-specific rules). Efforts to date have concentrated on relatively straightforward statistical measures, thresholds, weightings, and statistical aging of the profiles, independent of particular users. Considering that much is already known about insiders, it would seem highly desirable to include additional information in

the profiles. Physical attributes might include access to buildings, rooms, and computers based on real-time access records, using badges, logins to local machines, biometric authentications, interactive pager probes, cameras, and other sensor data. Planning data might include expected activities such as travel schedules, special arrangements regarding working hours, and facility restrictions. The combination of physical whereabouts and expected whereabouts could also be used to detect stolen badges or stolen authentication information of people who are highly trusted.

In previous systems aimed at insider misuse as well as intruders, statistical profiling (*e.g.*, in NIDES and EMERALD) provided the capability of monitoring individualized computer activities, such as which editors the user prefers, which programming languages, which mail environment, which variants of commands, and so on. This approach seems to be less relevant today, but still has potential where intrusion is not a primary concern.

Personal on-line behavior can also be profiled statistically by extending the analysis information that is recorded, such as with whom an individual tends to exchange e-mail, which Web sites are visited regularly, and even what level of sophistication the user appears to exhibit. There are also biological factors that might be monitored, such as how often a user gets up for breaks (activities that could also be monitored by physical access controls).

In environments in which monitoring key strokes is not considered intrusive, some effort has been made to monitor key-stroke dynamics. This approach tends to be much less reliable in general, particularly with confronted with network and satellite delays. Also, if you are typing with one hand because you are drinking a cup of hot coffee with the other hand, your typing dynamics are of course specious.

In addition to providing a real-time database relating to physical whereabouts, and extending statistical profiling to accommodate subtle computer usage variants, it would also be appropriate to represent certain external information regarding personal behavior, such as intellectual and psychological attributes.

As an example of an intellectual attribute, consider writing styles. There are already a few tools for analyzing natural-language writing styles. Profiles of individual-specific "msipelings", the frequency of obscenities and the choice of explicit expletives, the relative use of obscure words, and measures of obfuscational proclivities and Joycean meanderings might also be quite useful. (Recall Tom Lehrer's warning: Don't write naughty words on walls if you can't spell.)

Psychological factors do not seem to have been explored much in the past, especially in the context of insider misuse. Psychologists routinely observe certain standard behavioral characteristics and analyze deviations therefrom. Some of those characteristics that are particularly relevant to potential insider misuse might be modeled in extended user profiles. As one specific example, we might be able to develop measures of relative paranoia, based on how often a particular user invoked certain commands to observe who else might be observing what that user was doing in real time, or the use of aliases in posting to newsgroups. A measure of aggressive behavior could be interesting, but would probably require some human reporting of perceived relative hostility levels in e-mail messages received from a given individual. Measures of anger and stress levels in general computer usage could also

be conceived. However, considerably more effort is needed to characterize which psychological attributes might be effectively utilized. However, this is not likely to have much success, because there are not well established characteristics, and human variabilities are likely to confound them anyway.

If this approach is considered possibly fruitful, we should approach some psychologists who are familiar with computer users and ask them to speculate on psychological factors that might be both computer detectable and behaviorally discriminative with respect to insider misuse. On the other hand, users tend to be not particularly inherently risk-aware. However, see a *CACM* Inside Risks column by Dr. Leonard Zegans [25], which observes that, with respect to computer technology, users tend to take risks unconsciously and in many cases unwillingly.

The concepts noted above have significant privacy implications that must be addressed.

6 Requirements for Insider-Threat-Resistant High-Integrity Elections

The general problem of dramatically increasing the integrity of computerized election processes is in some sense a paradigmatic hard problem that encompasses a wide diversity of requirements addressing voter privacy, system integrity, data integrity, data confidentiality, system survivability, accessibility, and other issues. This problem nicely illustrates that the notion of an 'insider' is highly context dependent. sequentially and hierarchically varying, and distributed.

Insider threats exist essentially in every phase of election processes, including before, during, and after voting actually occurs. Within the development and use of voting technology, insider threats abound among system designers and implementers, system purveyors, system administrators, workers storing or installing voting machines, poll workers, poll judges, election officials, and anyone else with physical or logical access to voting systems – including voters who may be able to introduce Trojan horses or compromises (*e.g.*, through altered or privileged access cards). Note that anyone with physical access may be an insider in certain respects, but ideally would seem to be an outsider with respect to altering software, ballot definitions, data, and so on. Unfortunately, existing systems are sufficiently flawed that almost anyone could be a potential insider. Insider threats also exist among election officials and government employees extrinsic to and essentially independent of any technology,

Requirements for trustworthiness appear in various guises across the entire spectrum of system development and operation. They are relevant to electronic systems, but also to some extent to paper-based and mechanical lever systems. They apply irrespective of whether computer technology is used to prepare ballots, to tabulate ballots, or just to compile results – or indeed not at all (which is increasingly rare in the U.S., but quite common in other countries). They must also include oversight of people, on whom the systems and procedures are ultimately heavily dependent.

Risks of insider misuse arise whenever such a requirement is not satisfied (*e.g.*, [16], click on Election Problems). Suggested but by no means complete integrity requirements are summarized in the following stages. Ideally, end-to-end assurances of integrity, security and privacy must span all of these stages.

- Voter registration. Prospective voters must be impartially vetted for eligibility. Voter database entries for all eligible registrants must be timely and correct, tamper evident, and carefully audited for misuse; database errors must be detected as soon as possible, and be much easier for voters to correct quickly when wrong or not up to date. Accidental and intentional disenfranchisement of legitimate voters must be avoided, with voters assumed innocent until proven guilty.
- Voter authentication and authorization. Voter identification and validation must be impartial. Dependence on erroneous databases to validate users must be avoided, and errors easily corrected. Requisite alternative databases or paper backups must be available during elections in case primary sources are unavailable. Disputed challenges must result in voters casting provisional ballots, which must be fairly and promptly counted if valid. (Experience with the the Employment Eligibility Verification System (EEVS) [15] and its successor US-VISIT suggest that database errors present some enormous problems.)
- Voter information. Voters must be officially notified in a timely manner that is clearly differentiated from bogus information (exemplified by last-day phone calls informing selected voters that their polling places had been changed, or that they should vote on Wednesday to reduce polling place congestion, as occurred in November 2008).
- Polling place availability and accessibility. Adequate voting machines, paper ballots, provisional ballots, and so on must be available for all polling places. Past experiences with long delays in certain precincts need to be avoided. Alternative methods must be available (*e.g.*, paper ballots) when machines are inoperable. Early voting and absentee voting both should be suitably vetted and tamper evident; they present many problems of their own, and require careful oversight. (They must also be based on voter convenience, rather than constrained by draconian rules.) Disabled and disadvantaged voters must be allowed to vote conveniently and without undue time pressures from poll workers.
- Ballot layout and allocation. Ballot formats must be easily understood, unambiguous, and nonbiased, preferably thoroughly vetted beforehand. Voters must be given correct ballots (*e.g.*, in the proper language, and for the correct precinct, especially in multiprecinct polling places).
- Vote casting. Erroneous and poorly designed ballot faces and machines with confusing human/system interfaces must be avoided. Machines with identified fundamental flaws (such as the lack of audit trails and observed vote flipping) must not be used. The voting process must be voter friendly. Votes must be correctly recorded, and that process must be demonstrably verifiable.
- Vote counting and canvassing. The tabulation of votes must be *demonstrably* correct, with suitably high probability, reproducible through independent cross-checking, and demonstrably not subject to accidental errors, manipulation, and

Table 4 Insider Threats to Elections

Functions	**Intrinsic Insider Threats to Voting Systems**
Authentication	Misuse of various administrator and other login privileges; superuser usurpation; undocumented backdoor passwords
Access controls	Exploitation of granted privileges and flaws in voting software, underlying operating systems, and hardware
Confidentiality	Access to individual votes and crypto keys and passwords, before, during and after elections
Integrity	Tampering with voting machines and back-end tabulating systems: inserting bogus votes; altering existing votes; inserting trapdoors, Trojan horses, and -in-the-middle attacks
Service denials	Disabling protected components, resource exhaustion; power disruptions; rendering systems unbootable or otherwise unusable
Accountability	Altering audit trails and misuse detection subsystems (if any!)
Roles	**Extrinsic Insider Threats to Election Procedures**
Registration	Inserting bogus registrations, disenfranchising legitimate voters
Vetting voters by poll workers	Authorizing bogus voters, disenfranchising legitimate voters (*e.g.*, requesting *three* pieces of identification, or none, or providing a ballot for the wrong precinct)
Assistance	Misleading voters, *e.g.*, with erroneous instructions
Canvassing	Inserting bogus ballots, altering existing ballots or totals
Monitoring	Elections in which one party appoints both poll checkers
Remediation	Tampering with recounts, forcing do-overs

unmonitored tampering. The authentication of the totals must be essentially incorruptible.

- Monitoring. Oversight and visibility are essential. Computer audit trails and paper logs must be sufficiently comprehensive to allow for detection of errors, manipulation, and tampering, and valid for forensic use if needed. Insider denials of integrity and fraud need to be easily detected.
- Remediation. Following detected irregularities, meaningful definitive recounts must be possible via independent means (with mandated do-overs whenever sufficient evidence of uncertainty warrants). Random partial recounts must be mandatory, with greater extent for closer elections. Remediation itself must be free of errors, manipulation, and tampering.
- Openness throughout. All the above stages in the election process must be subject to thorough scrutiny and oversight. End-to-end assurances that these requirements are satisfied should be sought, encompassing registration, voter authentication, voting, overall accountability, and the resulting stages. Any gaps in the end-to-end assurance are likely to present opportunities for compromise.

An obvious conclusion from this brief itemization is that the insider threats are ubiquitous and pervasive. Every step in the election process is a potential weak link that can be easily exploited or accidentally exercised by insiders – especially in the absence of thorough monitoring and audit trails. When computers and electronic

communications are involved, the risks may be greater than in purely manual election processes. However, insider threats are significant in any case.

Table 4 summarizes some of the main threats from insiders. In this context, outside-inners are considered as insiders.

7 Relevance of the Countermeasures to Elections

Reflecting on the paradigmatic nature of the election problems, it is useful to consider how the countermeasures suggested above might apply to insider threats to elections. In that context, two sets of principles are particularly relevant: (1) Saltzer and Schroeder [22] as reformulated by Saltzer and Kaashoek (see Chapter 11 of [21] as well as some earlier extensions and reinterpretations [13]), and (2) Clark and Wilson [4]. These principles are paraphrased below for present purposes. (Clearly, some of these principles are relevant to outsiders as well as insiders.)

Saltzer-Schroeder-Kaashoek (SSK) Security Principles

- Economy of mechanism. Having to trust very complex systems is typically risky, especially in the presence of insiders. Complex mechanisms are inherently more likely to have vulnerabilities. Complex voting software that depends on a large underlying operating system with known vulnerabilities and many potential insiders engenders compromise from below through the operating systems and compromise from within through alterations of the voting software and its data.
- Fail-safe defaults. Access controls should generally permit access explicitly with a default of no access, rather than a default of total access unless explicitly denied. This simple-sounding principle can considerably reduce the likelihood of insider misuse, especially when combined with the other principles. Overly permissive access controls are inherently risk-prone. And yet, many of today's voting machines have few internal controls.
- Complete mediation. Whatever security controls are in place, they should not be bypassable or subvertible. This principle applies especially to monitoring and the completeness and integrity of audit trails.
- Open design. Reliance on the secrecy of a design and proprietary source code is generally a bad idea. Yet, security by obscurity pervades the marketplace for electronic election systems. The commercial voting system vendors in the U.S. persist with proprietary ownership of their software, its evaluations (which are commissioned and paid for by the vendors), and the internal data formats and election data. The notion that security would be diminished if anyone else had access to that proprietary knowledge is clearly a myth. However, the 'many eyeballs' approach must be countered by strict adherence to system integrity – which requires stringent version configuration control.
- Minimization of secrets. Closely related to the principles of open design and economy of mechanism is the notion of minimizing what must be trusted – and

indeed what must be trustworthy. For example, the confidentiality of cryptography necessarily depends on the protection of the crypto keys, but should not have to depend on the secrecy of the algorithm. In general, system architectures that strongly minimize what must be trustworthy are preferable to systems that do not. As an example, Ka-Ping Yee's Ph.D. thesis [24] on the Pvote electronic voting systems shows how the amount of software that must be trusted for election integrity can be dramatically reduced – to 460 lines of Python (not counting the Python execution environment, underlying operating system software, and hardware).

- Separation of privileges. Privileges should be defined separately for different roles, such as system developers, system operators, contractors, election officials, and auditors who attempt to resolve discrepancies, as well as authorized voters and provisional voters. Those privileges must be explicitly associated with needs to prevent unauthorized access to software, data, and system configurations. (See the principle of least privilege.)
- Least privilege. Given separation of privileges, only necessary privileges should be allocated according to appropriate roles. For example, voters should have no system privileges other than the ability to cast their votes. Vendors should not be permitted to alter certified code prior to or during an election (even if there is a complete audit record of what they have done and why). Vendor contractors and election officials who set up ballot faces should not be permitted to make any alterations to system configurations, software, or election data. Before and after elections, officials may have to perform some carefully controlled and audited reconfigurations – such as initializing an election, or closing it down. Vendors, contractors, and election officials should not be able to cast or alter any votes when serving in those roles. Test programs and test results should have absolutely no effect on the live election results. Voters should be unable to modify votes of other voters – or system configurations. And so on.
- Least common mechanism. Building a voting system on top of a widely used operating system that is also used for other purposes (such as system development and debugging) is not particularly wise. (However, using a lesser-known operating system still relies on security by obscurity—see above.) That approach can greatly increase the opportunities for insider misuse by anyone with access to the operating system. Similarly, system development and debugging should not be permitted on live systems. Furthermore, simplistic suggestions that ATMs (which identify and authenticate customers individually, with detailed audit trails and time-stamped photographic images) could also be used for voting (where anonymity and total privacy are desired) would most likely result in compromises of both banking integrity, vote integrity, and voter privacy!
- Ease of use. Where systems are not easily used, voters, poll workers, and others will make mistakes. Where systems are not easily maintained and are difficult for election officials to configure correctly with respect to security controls (or the lack thereof), certain responsibilities are likely to be outsourced to system vendors, such as ballot-face preparation, pruning supposed convicted felons from voter lists based merely on partial name matches, and oversight of the integrity

and ultimately the authority of elections. Clearly, these and other practices can increase opportunities for insider and even outside-inner misuse. Considerable difficulties are also created by systems that are supposedly useful for disadvantaged voters.

- Pervasive auditing. Stringent auditing within trustworthy systems could enable meaningful recounts in cases of disputes (a facility that is almost nonexistent at present) and further provide forensic-quality evidence in cases of tampering or accidental changes. Such auditing should achieve high integrity, completeness, nonalterability, and nonsubvertibility, and ideally must be derived independently from the systems being monitored. Redundancy is particularly important here. (This is a generalization and strengthening of what is termed 'compromise recording' by Saltzer and Schroeder.)

Clark-Wilson Integrity Model

For completeness, the Clark-Wilson properties are enumerated here in a simplified form, even though there is some overlap with the Saltzer-Schroeder-Kaashoek principles, and some of them are less relevant specifically to insider misuse.

- Enforcement 1: Users may operate on data only through controlled operations, never directly (SSK: Encapsulation, complete mediation). Back-door access to programs and data, reset commands, and alterable audit trails are risky – before, during, and after elections!
- Enforcement 2. Users may perform operations only if explicitly authorized (Authorization).
- Enforcement 3. User identities must be authenticated (Authentication).
- Enforcement 4. Authorizations can be changed only by a security officer (Nondiscretionary controls). However, those privileged users clearly remain an insider threat.
- Certification 1. Data must be verified as a consistent representation of the real world (Data validation). This would also tend to flag situations in which the number of votes exceeds the number of voters.
- Certification 2. Programs must implement operations as *well-formed* transactions (Consistent transformations). Transactions need to be certified before going into operation. Insider manipulations of voting data could be detected via inconsistencies in cross-checking or violations of integrity seals.
- Certification 3. Systems must enforce separation of duties. (SSK: Roles, separation of privilege, and least privilege are fundamental to coping with insider threats in voting systems.)
- Certification 4. All operations must be audited. (SSK: Complete auditing)
- Certification 5. Operations must validate inputs, or else reject them (Input validation and atomicity). An untrusted operation must still be atomic – that is, either completely accepted or completely aborted.

Most of the Saltzer-Schroeder-Kaashoek and Clark-Wilson principles can contribute significantly to election integrity in the face of accidental and intentional insider misuse, along with other principles (*e.g.*, [13]). For example, the principle

of separation of policy and mechanism suggests that policy that may need to be altered should not be embedded in mechanism. A rather astounding example of the violation of that principle is found in Sequoia's election systems that are reprogrammed *after the system is certified* in order to support each different ballot face. For Diebold (now Premier), in every one of 17 counties using their equipment in the California election of November 2003, the versions of those systems that were in operation were not the certified versions. Such violations of what is a common principle in software engineering represent one of the most dangerous examples of opportunities for insider misuse in existing voting machines.

Overall, election officials today have only superficial control over the entire life cycle, including operations. The complexity of some of the all-electronic systems is such that the major vendors tend to provide their own personnel to help with setting up ballot faces and addressing technical problems that occur before, during, and even after elections. In the existing commercial systems, developers and vendors have considerable latitude in making surreptitious system changes that could alter the results of elections—including cases in which election software was not the certified software (as noted above). The absence of meaningful audit trails and the alterability of audit trails that do exist further complicate the process, because of the lack of demonstrable provenance and an almost complete lack of records on the alteration of code and election data. Ironically, the Nevada Gaming Commission and other states hold gambling systems to standards and evaluation processes that are astoundingly more stringent than those for voting systems.

As is the case elsewhere, various approaches to coping with insider threats may also be applicable to coping with outsider threats, and in some cases to natural disasters and other so-called acts of God. Once again, we stress the importance of system architectures and development processes that are capable of systematically addressing all of the critical requirements – including usability.

8 Research and Development Needs

Overall, there is still much research and development work to be done, some useful for trustworthiness in general, some specific to insider threats, and some specific to elections.

- The commonalities among insider and outsider misuses need to be understood, with respect to threats, methods, exploitations, detection techniques, and responses, taking advantage of those commonalities where possible, and resorting to different but compatible approaches where commonality is not immediately evident – all within the context of developing trustworthy systems that can address insider misuse as well as other critical requirements. Significant effort must be devoted to defining characteristic types of insider misuse. (For example, see [9, 11].)
- Finer-grained access policies and access controls are needed to help define what constitutes proper usage, thus facilitating the role of insider-misuse detection.

Greater effort needs to be devoted to detecting unknown modes of misuse, rather than focusing on just detecting known attacks. Hierarchical and distributed correlation is also required, to identify common patterns and user intentions. Extrinsic individual characteristics such as psychological behavior that might be included in profiling user activities need to be identified.

- Better system architectures to minimize what must be trustworthy (*e.g.*, Pvote [24]) and better software engineering practices are needed for developing high-integrity trustworthy computer systems (including subsystems for monitoring and analysis) to make them more composable [13] and interoperable (to permit mix-and-match multivendor systems), robust, evolvable, and extensible in their application domains—including attributes such as reliability, fault-tolerance thresholds, survivability, performance, and suitability for combatting insider misuse. For example, the NSF ACCURATE center (http://accurate-voting.org) is exploring various alternatives, including the development of VoteBox and experimentation with its usability (http://votebox.cs.rice.edu), Pvote, and other efforts.
- Real-time analytic systems must themselves be tamper resistant, to hinder integrity and denial-of-service attacks, alterations of evidence (either by malfeasors to cover their tracks, or by law enforcement in attempting to fake evidence).

9 Conclusions

In general, insider misuse cannot sensibly be treated as an isolated problem. For example, combatting it critically depends on the existence of meaningfully trustworthy systems and applications, nonspoofable identity management and user authentication, fine-grained access controls, controlled monitoring that is not excessively privacy invasive, carefully specified and enforced operational practices, transparency in the sense of being readily understood, privacy-respecting oversight, metrics for determining effectiveness of countermeasures, and ultimately the honesty, integrity, and diligence of trusted users. Although most of these approaches are relevant more pervasively, they are particularly important in elections – where total-system approaches and end-to-end assurances are essential in monitoring and protecting against insider threats.

In the spirits of Saltzer-Schroeder-Kaashoek and Clark-Wilson, this chapter may seem to be old wine in new bottles. However, the vintage is superb and still timely. Besides, still wine runs deep – the spirits are willing, but the flash (memory) is weak.

Acknowledgment. This chapter is based on a position paper, Combatting Insider Misuse, with Relevance to Integrity and Accountability in Elections and Other Applications, for the Dagstuhl Workshop on Insider Threats, 20-25 July 2008, It was prepared with funding from National Science Foundation Grant Number 0524111, under the SRI project for ACCURATE: A Center for Correct, Usable, Reliable, Auditable and Transparent Elections.

The author is grateful to Matt Bishop, Drew Dean, Jeremy Epstein, and Sean Peisert for some incisive interactions. Matt also presented my position paper at the Dagstuhl 2008 workshop.

References

1. K.J. Biba. Integrity considerations for secure computer systems. Technical Report MTR 3153, The Mitre Corporation, Bedford, Massachusetts, June 1975. Also available from USAF Electronic Systems Division, Bedford, Massachusetts, as ESD-TR-76-372, April 1977.
2. M. Bishop. Position: 'Insider' is relative. In *Proceedings of the 2005 New Security Paradigms Workshop*, pages 77–78, Lake Arrowhead, California, October 2005.
3. M. Bishop, S. Engle, C. Gates, S. Peisert, and S. Whalen. We have met the enemy and he is us. In *Proceedings of the 2008 New Security Paradigms Workshop*, Olympic Valley, California, 2008.
4. D.D. Clark and D.R. Wilson. A comparison of commercial and military computer security policies. In *Proceedings of the 1987 Symposium on Security and Privacy*, pages 184–194, Oakland, California, April 1987. IEEE Computer Society.
5. F.J. Corbató. On building systems that will fail (1990 Turing Award Lecture, with a following interview by Karen Frenkel). *Communications of the ACM*, 34(9):72–90, September 1991.
6. R.C. Daley and P.G. Neumann. A general-purpose file system for secondary storage. In *AFIPS Conference Proceedings, Fall Joint Computer Conference*, pages 213–229. Spartan Books, November 1965.
7. V.D. Gligor *et al*. Design and implementation of Secure Xenix[TM]. In *Proceedings of the 1986 Symposium on Security and Privacy*, Oakland, California, April 1986. IEEE Computer Society. also in *IEEE Transactions on Software Engineering,* vol. SE-13, 2, February 1987, 208–221.
8. P.A. Karger. Limiting the damage potential of discretionary Trojan horses. In *Proceedings of the 1987 Symposium on Security and Privacy*, pages 32–37, Oakland, California, April 1987. IEEE Computer Society.
9. C.E. Landwehr, A.R. Bull, J.P. McDermott, and W.S. Choi. A taxonomy of computer program security flaws, with examples. Technical report, Center for Secure Information Technology, Information Technology Division, Naval Research Laboratory, Washington, D.C., November 1993.
10. D. Maughan *et al.* A roadmap for cybersecurity research. Technical report, Department of Homeland Security, November 2009.
11. P.G. Neumann. *Computer-Related Risks*. ACM Press, New York, and Addison-Wesley, Reading, Massachusetts, 1995.
12. P.G. Neumann. Practical architectures for survivable systems and networks. Technical report, Final Report, Phase Two, Project 1688, SRI International, Menlo Park, California, June 2000. http://www.csl.sri.com/neumann/survivability.html.
13. P.G. Neumann. Principled assuredly trustworthy composable architectures. Technical report, Computer Science Laboratory, SRI International, Menlo Park, California, December 2004. http://www.csl.sri.com/neumann/chats4.html, .pdf, and .ps.
14. P.G. Neumann. Reflections on system trustworthiness. In Marvin Zelkowitz, editor, *Advances in Computers, volume 70*, pages 269–310. Elsevier Inc., 2007.
15. P.G. Neumann. Security and privacy in the employment eligibility verification system (eevs) and related systems. In *Congressional Record*, Washington, DC, Jun 7 2007. U.S. House of Representatives.
16. P.G. Neumann. Illustrative risks to the public in the use of computer systems and related technology, index to RISKS cases. Technical report, Computer Science Laboratory, SRI International, Menlo Park, California, 2009. Updated now and then:

http://www.csl.sri.com/neumann/illustrative.html; also in .ps and .pdf form for printing in a denser format.
17. P.G. Neumann, R.S. Boyer, R.J. Feiertag, K.N. Levitt, and L. Robinson. A Provably Secure Operating System: The system, its applications, and proofs. Technical report, Computer Science Laboratory, SRI International, Menlo Park, California, May 1980. 2nd edition, Report CSL-116.
18. P.G. Neumann and P.A. Porras. Experience with EMERALD to date. In *Proceedings of the First USENIX Workshop on Intrusion Detection and Network Monitoring*, pages 73–80, Santa Clara, California, April 1999. USENIX. Best paper.
19. P.A. Porras and P.G. Neumann. EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances. In *Proceedings of the Nineteenth National Computer Security Conference*, pages 353–365, Baltimore, Maryland, 22-25 October 1997. NIST/NCSC.
20. J.H. Saltzer. Protection and the control of information sharing in Multics. *Communications of the ACM*, 17(7):388–402, July 1974.
21. J.H. Saltzer and F. Kaashoek. *Principles of Computer System Design*. Morgan Kauffman, 2009. Chapters 1-6 only. Chapters 7-11 are online: http://ocw.mit.edu/Saltzer-Kaashoek .
22. J.H. Saltzer and M.D. Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308, September 1975.
23. S. Stolfo, S. Bellovin, S. Hershkop, S. Sinclair, and S. Smith. *Insider Attack and Cyber Security: Beyond the Hacker*. Springer, 2008.
24. K.-P. Yee. *Building Reliable Voting Machine Software*. PhD thesis, University of California, Berkeley, 2007. Technical Report 2007-167; see also Technical Note 2007-136 for the security review; http://pvote.org.
25. L.S. Zegans. The psychology of risks. *Communications of the ACM*, 51(1):152, January 2008. *Inside Risks* column.

Appendix: Some Illustrative Examples of Insider Misuse

The ACM Risks Forum (http://www.risks.org) and an annotated index [16] include many cases of insider misuse. Selected insider cases noted here illustrate a wide range of applications and effects.

• Aldrich Ames (who was in charge of monitoring abuses) was guilty of spying activities inside the CIA. Robert Hanssen had been spying largely undetected for almost 22 years. After the apprehension of Ames, the FBI placed Hanssen in charge of detecting additional spying activity – namely, his own!

• An Autotote ex-programmer with insider knowledge hacked the winning Breeders' Cup Pick-Six horse-race off-track betting system, after a previous trial went undetected. The off-track system transmitted results to a central facility only after the completion of the fourth race. Dummy bets were placed for the first four races with wild cards for every possibility for the subsequent two races. Before transmission, the dummy bets were altered to specify the known winners of the first four races. As a result, one Pick-Six and many Pick-Five choices were winners. Drexel University fraternity buddies were implicated. Programmer Chris Harn was sentenced for only a year and a day in jail, because he helped the authorities incriminate his buddies, who received two- and three-year sentences.

• Harrah's Tahoe was victimized by a $1.7-million payoff scam, with insertion of a Trojan-horse chip and an electronically triggered payoff suspected. An attempted lottery fraud in Pennsylvania resulted when an insider managed to print a winning ticket with a backdated timestamp, but was detected.

• The Washington DC Real Property Tax Administration Adjustments Unit was victimized by a group of insiders who filed for fake property tax refunds for $30 to $50 million – and remained undetected for over 10 years, despite the size of the conspiracy, which included Bank of America employees who knowingly cashed the fraudulent checks. This case may still be in the courts. (Browse on 'Harriette Walters'.)

• Various law-enforcement misuses of databases and thefts of criminal records have been reported, involving police, an FBI employee, and a former Drug Enforcement Agency employee. An IRS agent was accused of giving a defendant tax data on judges and jurors. Social Security Administration employees sold 11,000 Social Security Numbers to activate cards stolen in the mail. Forty people including nine postal workers were arrested in an extensive credit-card fraud in Washington DC. In Virginia Motor Vehicle frauds, illicit driver licenses were sold for as much as $3,500; some years ago, the going rate was only $25. California DMV clerks were fired for similar offenses. Database misuse by 11 prison guards in Brooklyn leaked names of informants to prisoners, warning about impending searches.

• Numerous cases of insider misuse and fraud in banking, credit cards, identity theft, stock trading, security guards, call centers, a hospital nurse changing prescriptions and treatments, ... The recent Madoff Ponzi scheme appears to be just the emerging tip of an insider-fraud iceberg. Volkswagen Corp lost $260 million to a computer-based foreign-exchange fraud; four insiders and one outsider were convicted. A military pay fraud netted $169,000 using a bogus account. A massive New York City tax fraud wiped out $13M in taxes; many insiders were implicated. Joseph

Jett created $350-million phantom profits undetected by Kidder Peabody oversight, and received a bonus of $9M. Many bank customers (48,000 at Wachovia, 600,000 at Bank of America, and others at Commerce Bank and PNC Bank of Pittsburgh) were notified that their financial records were potentially compromised by an insider operation. Insider thefts of sensitive information include 6000 AIDS records stolen from a Miami hospital. Misuse of sensitive personal information has led to many security violations and compromises. In addition, various cases of stolen or lost laptops have involved compromise of unauthorized or inappropriately stored insider information.

• Election irregularities. In 1984, David Burnham reported on election fraud in *The New York Times.* In Colorado, absentee ballot fraud was reported going back to 1984. In 1999, 22 people were indicted in Louisiana in a bribery/kickback scheme. In 2009, five insiders in Clay County, Kentucky, were indicted for previous election frauds, including systematically altering ES&S iVotronic electronic ballots after voters had been intentionally misled about the user interface. Other cases of fraud are also suspected.

http://www.springer.com/978-1-4419-7132-6

Insider Threats in Cyber Security
Probst, C.W.; Hunker, J.; Bishop, M.; Gollmann, D. (Eds.)
2010, XII, 244 p., Hardcover
ISBN: 978-1-4419-7132-6