An Architecture for Contextual Insider Threat Detection

Michael Kirkpatrick*, Elisa Bertino
Department of Computer Science
Purdue University
{mkirkpat,bertino}@cs.purdue.edu

Frederick Sheldon
Cyberspace Science Information Intelligence Research
Oak Ridge National Laboratory
sheldonft@ornl.gov

Abstract

Recent studies have shown there is a growing concern about the damage possible when trusted organization insiders behave maliciously. In particular, data exfiltration can lead to loss of revenue, damage to an organization's reputation, and disruption of service for critical infrastructure systems. In this work, we introduce the **C**ontextually **A**daptive **IN**sider threat architecture (CAIN), which incorporates contextual and risk-based access control with anomaly detection. While traditional Mandatory Access Control (MAC) can offer high assurance for information security, its rigid structure can hinder workers' productivity. The goal of CAIN is to balance these dual goals of data protection and flexible access. This paper outlines the design goals of CAIN, as well as the behavior of its components. We also describe our initial design for building a prototype of our system and our future work.

1 Introduction

As organizations became increasingly connected via corporate intranets as well as the Internet, researchers developed a number of tools for protecting data from external threats. Firewalls, virus scanners, intrusion detection systems, and virtual private networks (VPNs) all focus on creating a division between the members of the organization and external attackers. The assumption is that the security threat to data and systems lies with potential attackers outside of the organization. As these tools have become mature technologies and have been widely adopted, a new direction in security research is to examine the threat posed by insiders.

In this work, we use the defintion of "insider" to be any person that has currently or has previously had authorized access, as used in the 2004 report by CSO Magazine, CERT, and the United States Secret Service [12] (hereafter referred to as the CERT report). While other works do not consider people who no longer have access as insiders, the advantage of the CERT definition is that former employees may use their knowledge of the organization as part of an attack. Specifically

excluded from the definition of an insider is an external attacker who has gained illicit access by exploiting a vulnerability of some sort. Also, we are not considering collusion, as inside attackers are more likely to work alone or to help an external group than to collude with other insiders[1].

It is important to note that this definition of insider makes no assumption about the person's role or ability. That is, an insider can be a someone with management duties, an information technology specialist or administrator, a member of the sales department, etc. An insider can also be a part-time employee, a contractor, a vendor, or a corporate alliance partner, with the only assumption made that the person has some sort of access to the organization and/or its systems.

The insider threat problem is particularly challenging, because it is difficult to predict or protect against, it is often misperceived or underestimated, and it has the potential for high impact [11]. Despite the challenge, the CERT report demonstrated the need for to address the problem. According to the report, 29% of attacks on survey respondents' organizations were from insiders. Of these attacks, 34% involved "critical disruption" to the organization, its customers, or the larger critical infrastructure, which includes systems of government, telecommunications, finance, energy, etc. One of the most notable facts is that almost half of the respondents indicated that some or all of the insider-based attacks were discovered *accidentally*. This fact indicates that there is a critical need to develop tools for detecting and preventing insider attacks.

In this work, we introduce the **C**ontextually **A**daptive **IN**sider threat architecture (CAIN). The goal of CAIN is to provide a detection and rapid response mechanism for a shared file system. CAIN considers both data exfiltration, wherein an attacker is attempting to steal data for financial gain or as part of an ideological campaign, as well as sabotage, in which the goal is to corrupt data. Our design includes a number of components that perform statistical analysis on user's access patterns to detect if their behavior begins to deviate from their expected duties, indicating the user may be attempting an attack.

While many insider attacks could be prevented by implementing tight Mandatory Access Control (MAC) policies, doing so appears to be impractical. In particular, rigid access controls are often perceived as too cumbersome, and organizations err on the side of granting too much access [38]. CAIN addresses this challenging by permitting the use of contextual and risk-based access control policies. That is, an organization can set a restrictive base of permissions, while the access control component of CAIN performs an analysis and grants provisional access if the risk is below a set threshold.

In this paper, we present the underlying structure of CAIN and the statistical techniques for our approach. We describe our future work, including a CAIN prototype for a network file system (NFS). We close with a discussion of open challenges for future research.

2 Related Work

A number of works have shown the severity and reality of insider threats [11, 15, 12, 35, 10]. Recent work to defend against insider threats has focused on cost-sensitive document classification [27], semantic analysis of documents [37, 32], and the use of Bayesian networks [9, 20]. Similarly, a number of works have examined the utility of various statistical techniques for anomaly detection of system calls [21], network messages [29], and computer usage [28]. Ni *et al.* have applied statistical classifier techniques, such as support vector machines, to identify typical behavior for defining roles

[1]To be precise, for data theft, almost half of attacks involve collusion with either outsiders or other insiders. However, in other attacks, such as sabotage, insiders almost exclusively work alone [10].

in role-based access control (RBAC). Our work differs from these in multiple ways. First, most of these works are not designed specifically for addressing insider threats. Next, none of these works focus on the development of a tool for insider threat detection and response, which is our goal. Finally, we are also exploring the use of alternative mathematical approaches to uncertainty, such as Dempster-Shafer theory [26].

Our design for a kernel-based access control system is based on SELinux [31, 34]. Our decision request engine (DRE) is roughly comparable to the SELinux security server, although our aim is to incorporate contextual and user profiling in the access control mechanism. Existing multi-level security designs [8, 7] do not incorporate this information, so they are too restrictive for our work.

A number of works have focused on incorporating risk into flexible access control systems [38, 14, 16]. Others have defined extensions to RBAC that consider contextual and spatiotemporal factors [13, 6, 19]. Smaldone *et al.* have proposed the use of a working-set of files for remote access [30]. We have included a number of these factors into our design for CAIN. Additionally, IFEDAC [22] is an interesting approach that incorporates information flow into discretionary access control (DAC) to prevent trojan horse attacks. While this approach would be beneficial to incorporate into our access control model, it does not address the problem of insider threat detection.

Finally, there are two key features of our design that are related to previous work. First, our architecture aims to follow the XACML model [24] for access control. Specifically, we define a process decision point (PDP), a process information point (PIP), and a process enforcement point (PEP). Next, Kamra *et al.* [18, 17] have proposed techniques for anomaly detection in databases, as well as defining a policy language for enforcing multi-level responses to attacks. This approach has inspired our separation of the anomaly detection engine (ADE) and the anomaly response engine (ARE).

3 Problem Statement and Approach

While a number of studies have been done to demonstrate the need for protection against insider threats, there is little consensus over what precisely constitutes an insider. A number of characteristics have been identified in one or more previous works [4, 25, 5, 33, 10]. An insider may have currently granted authorization to use a system or network, or his permission may have been revoked. His status as an insider may give him knowledge and privileges not granted to the population at large. An insider may be falsely perceived as working loyally for an organization, or he may be a loyal but incompetent user who makes critical mistakes.

Additionally, an insider may have a variety of backgrounds and motivations [33, 1, 3, 10]. He may be malicious, acting for financial, business, or ideological gain. He may be an information technology specialist with detailed knowledge of critical systems. Disgruntled employees may become attackers, seeking retribution for a perceived wrong. Non-malicious insiders may bring harm to the organization through careless actions. In particular, a computer virus or other malware may take advantage of the privileges granted to an employee.

Finally, there are different types of damage that can result from an insider threat [2]. *Data extrusion* refers to any movement of data out of the organization's domain of control, whether intentional or accidental. However, *data exfiltration* is intentional data extrusion in violation of a security policy[2]. *Corruption* is the insertion of incorrect data, or the unauthorized modification of

[2]Note that data exfiltration can occur even if the policy is not explicitly stated. For instance, making a copy of a company's client list and selling it to a competitor is obviously exfiltration, even if the employee is never directly

existing data. *Unauthorized communication* involves establishing an illicit network connection or creating a covert channel, typically leading to data extrusion or corruption.

Given these characteristics and threats, we establish the following working definitions:

Insider Any person with current or previous authorized access to, or knowledge of, a system or network.

Insider Threat The vulnerability of an organization to harm, whether financial, operational, or lost reputation, as the result of the actions of an insider, whether deliberate or unintentional.

Insider Threat Problem The challenge of protecting an organization's data against data extrusion, corruption, or unauthorized communication that can result from an insider threat.

Although it would be desirable to prevent all damage caused by insider threats, the subtlety of the problem make doing so impossible. As an informal illustration, if a user prints a sensitive document, the file server cannot determine what the user intends to do with the document after it is printed. Thus, our aim is not to prevent insider attacks, but rather to detect the possibility of an attacker and respond in ways to mitigate the resulting damage.

The goal of the CAIN project is to develop a kernel-based contextual access control mechanism for a file server. In addition, daemon processes on the server perform statistical anomaly detection and provide a multi-level response mechanism. The underlying assumption is that an attack requires the user to deviate from his typical behavior and computer usage. By detecting anomalous behavior, an organization may take earlier action to monitor an insider who may be considering an attack. Although this tool may not prevent insider attacks, it may help organizations to respond and recover from attacks in a rapid manner.

4 Architecture

CAIN consists of a combination of active and passive components that are distributed through both kernel and user space. The actors include a kernel engine responsible for run-time access control decisions and multiple privileged user space engines[3] are responsible for profiling user behavior, as well as detecting and responding to anomalous behavior.

4.1 Active Components

The active components of CAIN are defined as follows.

- **Decision Request Engine (DRE)** – When a system call is invoked, the call requests an access decision from the DRE. The DRE consults a number of resources, including the policy, the user's context, the user's behavioral profile, and a cache mechanism (see ARC below). The DRE is serves as the active component of the PDP, and corresponds to the SELinux Security Server.

told doing so would be a policy violation.

[3]It is important that the reader not conflate the term "user space" with processes run by users. The user space engines run as privileged daemons, sending asynchronous updates to the kernel only when necessary; implementing these engines in kernel space would lead to unacceptable performance costs.

- **Policy Refinement Engine (PRE)** – The PRE computes and updates a profile of each user's typical behavior. The PRE reads the logged access requests and the corresponding decisions, aggregating statistical data across multiple time frames (daily, weekly and monthly). Although the PRE has no direct contact with the PDP, its output is used by the DRE; hence, we classify the PRE as part of the PIP.

- **Anomaly Detection Engine (ADE)** – The ADE examines the access log, the user's profile, and the user's context searching for anomalous access requests. The ADE uses statistical classifier techniques, such as support vector machines, to classify requests. As output, the ADE produces a summary of the anomalous requests, including the unique characteristics of the request, the magnitude of the variation,

- **Anomaly Response Engine (ARE)** – The ARE uses the report of the anomalous request as produced by the ADE and performs a risk analysis of the request. Based on the risk level computed, the ARE provides a multi-tiered response, including updating the user's context, sending an alert to an administrator, and flagging the request for further investigation.

4.2 Passive Components

- **System Call (Syscall)** – Whenever a user process requests access to read from or write to a file, the process invokes a syscall that serves as the interface between the process and the operating system. In our design, the syscall invokes a security hook that contacts the DRE. Once the DRE evaluates the request, the syscall returns either a file descriptor or a value indicating why the access was denied.

- **Access Request Cache (ARC)** – To expedite the access decision, the DRE maintains a cache of recent requests and the corresponding response. If the user's context and profile have not changed, and the current request matches a previous request, the DRE can use the response stored in the ARC instead of performing a new evaluation.

- **Policy** – The policy specifies the accesses that should be granted or denied. In general, we assume a basic RBAC policy, although other policies could be used, as well.

- **Profile** – The profile consists of persistent information about the user and his access habits. Further information regarding the details of the profile is provided below.

- **Context** – The context consists of transient information about the user's current session. That is, while the profile contains long-term information about the user, the context reflects the current, short-term status. The factors included in the context are listed below.

- **Log** – The log file contains information regarding the access request parameters, decision, timestamp, and information regarding the process that issued the request. This file is examined by the PRE to establish long-term trends of the user's requests, as well as by the ADE in extracting anomalous behavior.

Each user of the system has a profile that reflects that person's typical computer usage. Building on the work of Shavlik & Shavlik [28], we have identified a number of characteristics (represented by the mean and median for numeric values) for constructing each user's profile, as shown in Figure 1. Figure 2 shows similar characteristics for the user's current context.

User Profile Characteristics	
Regular working hours (represented by time for first and last daily access)	Number of access permission errors
	Bytes received per second
Common applications and associated file types	Bytes transmitted per second
Working set of files and directories	Print job average and total size
File access duration per file type	Most common activated roles
File write operations per second	All authorized roles
Number of files open simultaneously	

Figure 1: Characteristics stored in each user's profile

User Context Dimensions	
Current time	Number of open files
Active roles	Time since last file write
Provisional and emergency authorizations	Existing anomaly responses
Current working set of files and directories	Location

Figure 2: Characteristics stored in each user's immediate context

4.3 Information Flow

The information flow between components of our architecture is summarized in Figure 3. Note that this figure does not imply any specific synchronization between the components. Rather, the timing of the system can be described as follows.

- **Access Request** – The user issues a syscall requesting access to a resource. This access can be reading or writing to a file, executing a binary image, mounting a file system, opening a network socket, etc. The DRE examines the request, the user's context, and the user's profile, and compares this information with the ARC. If the ARC contains a corresponding entry, and the profile and context are unchanged, then the decision stored in the ARC is returned to the syscall.

  Otherwise, the DRE compares the request, context, and profile with the policy. If the policy is designed to allow a flexible, risk-based access control, the DRE may perform a rudimentary analysis and grant an access that is not explicitly allowed. For instance, if a user requests access to a file that is in a similar user's working set, the DRE may permit the access if the risk of doing so is below a set threshold. Regardless of the policy, the DRE then stores the request and response in the ARC, and returns response to the syscall. The DRE also logs the request and response for protected resources[4].

- **Profile Refinement** – The PRE runs as a batch process parsing the log file and updating the profile as appropriate. The scheduling of the PRE can be set for times when the system has a low workload, such as overnight.

[4]To reduce the burden of the logging and analysis components of our design, we do not log information related to reading unprotected (i.e., world-readable) files. However, if the user attempts to write to one of these files or to make a file world-readable, those events are logged, as they may involve leaking sensitive information.

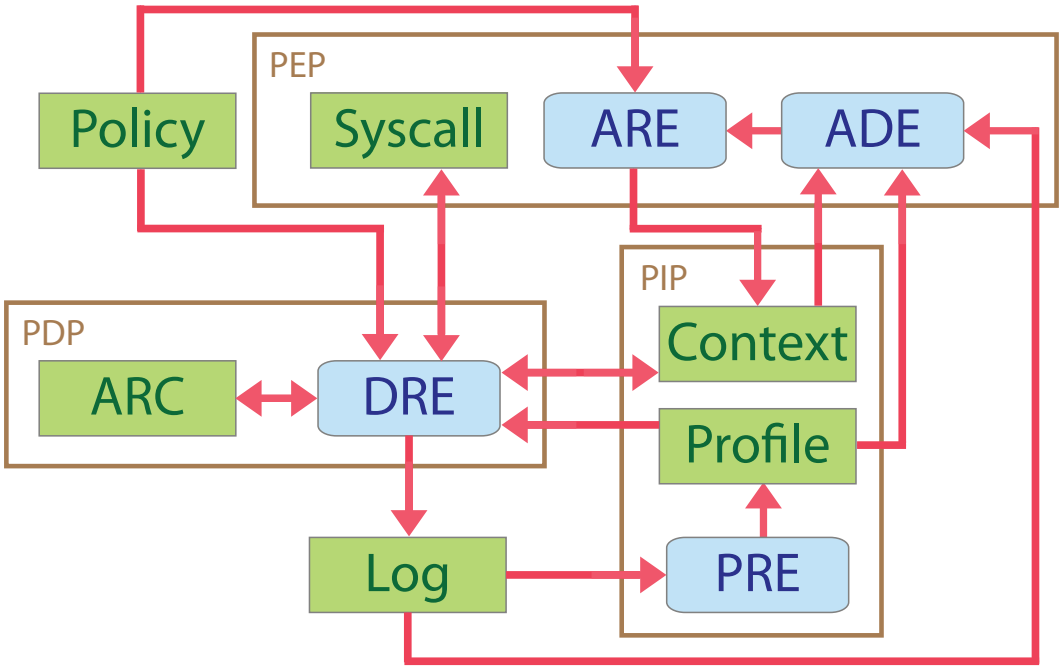

Figure 3: Information flow in the CAIN architecture

- **Anomaly Detection and Response** – The ADE runs as a background process, regularly pulling data from the log, user's context, and user's profile. Depending on the sensitivity of the system and the need for rapid response to such threats, the ADE could be scheduled to run frequently, such as every hour. The ARE is invoked by the ADE as necessary when a potential threat has been identified.

4.4 Statistical Techniques

Our design incorporates a number of components that perform statistical and probabilistic analysis. First, the PRE constructs a profile of the user's typical behavior. As stated previously, our definition of the profile is derived from the work of Shavlik & Shavlik [28], indicating that the profile consists of simple averages and sums. Thus, the PRE's analysis involves basic operations that can be quickly executed.

If a risk-based access control policy is permitted, the DRE must perform a computation to evaluate the risk of granting an access. However, the calculation must occur quickly, as the DRE is consulted for every access request. As such, the risk is a weighted sum of the profile and context dimensions. If the sum is below a threshold defined by the organization, the request is granted provisionally and is logged as appropriate. The provisional authorization is then added to the user's context for consideration in future requests.

The ADE performs the most complex statistical analysis. We are exploring two approaches for anomaly detection. Ni *et al.* [23], have demonstrated that support vector machines (SVM) have a high accuracy rate identifying roles and entitlements. On the other hand, Laskey *et al.* [20] have defined an approach for detecting threatening behavior with Bayesian networks. We are

implementing both approaches for evaluation in future work.

Once the ADE has completed its analysis, an appropriate response must be raised. One particular challenge of automating responses is to reduce the number of false positives. Our approach is to apply Dempster-Shafer theory as a foundation for risk analysis. That is, for a request that is flagged as anomalous by the ADE, the ARE uses the dimensions of the profile and context, as well as the magnitude of the request's distance from normal behavior, and uses the Dempster-Shafer method of combining evidence to construct a probability that the request is part of an attack. Based on thresholds defined by the organization, the ARE can respond in a number of ways, such as alerting an administrator, placing a warning in the user's context, or revoking provisional accesses.

5 Current & Future Work

While the focus of our work has previously been to design the architecture and to identify the requisite statistical techniques described above, our current work is to implement a CAIN prototype for an NFS-based file server. Our approach is to use the Linux Security Modules (LSM) [36] framework for implementing the DRE, as well as the passive components of the design. The PRE, ADE, and ARE are designed as daemon processes that interface only with the passive components of the architecture. Once these modules are in place, we intend to perform an empirical evaluation of the performance of the statistical techniques and refine our design as appropriate.

Additionally, we plan to expand the architecture as a distributed system across multiple servers and workstations. We would then look to deploy CAIN within a real organization for further evaluation of the statistical techniques. The goal of this deployment would be to accumulate data relating to the realistic behavior of users within an organization, and to have a detailed log if an attack occurs.

6 Conclusion

Although there are many intrusion detection tools available, these tools were designed under the assumption that attacks originate from an external source. That is, they are neither applicable nor sufficient for detecting and neutralizing insider threats. In this work, we have introduced the design of CAIN as a tool for insider threat detection and response. We have described the components of CAIN, as well as the information flow within the architecture. Our design permits the use of contextual and risk-based access control to offer increased flexibility over rigid techniques, such as MAC. However, the use of user profiling and statistical anomaly detection give organizations the ability to identify potential insider threats and respond appropriately. We have described our approach for developing a prototype of CAIN for an NFS server. The design we have presented in this paper is a first step toward creating a tool for organizations to address the threat posed by malicious insiders.

References

[1] Security awareness bulletin. Tech. Rep. 2-98, Defense Security Institute, September 1998.

[2] Information assurance technical framework (iatf), version 3.1. Tech. rep., National Security Agency, September 2002.

[3] Information security: Further efforts needed to fully implement statutory requirements in dod. Tech. Rep. GAO-03-1037T, Government Accountability Office (GAO), July 2003.

[4] President's national strategy to secure cyberspace, February 2003.

[5] Cyber security research and development. Tech. Rep. Broad Area Announcement 07-09, Department of Homeland Security (DHS), May 2007.

[6] Aich, S., Sural, S., and Majumdar, A. K. Starbac: Spatiotemporal role based access control. In *OTM Conferences* (2007).

[7] BAE Systems. XTS-400 Trusted Computer System for Multi-Level Security. http://www.baesystems.com/ProductsServices/bae_prod_csit_xts400.html.

[8] Bell, D. E., and Padula, L. J. L. Secure computer system: Unified exposition and multics interpretation. Tech. Rep. ESD-TR-75-306, MITRE, Mar. 1976.

[9] Calandrino, J. A., McKinney, S. J., and Sheldon, F. T. Detection of undesirable insider behavior. In *Cyber Security and Information Intelligence Research Workshop (CSIIRW)* (2007).

[10] Cappelli, D., Moore, A., Trzeciak, R., and Shimeall, T. J. Common sense guide to prevention and detection of insider threats. Tech. Rep. 3rd Edition – Version 3.1, CERT, January 2009.

[11] Chinchani, R., Iyer, A., Ngo, H. Q., and Upadhyaya, S. Towards a theory of insider threat assessment. In *International Conference on Dependable Systems and Networks (DSN '05)* (2005).

[12] CSO Magazine and CERT and United States Secret Service. 2004 e-crime watch survey: Summary of findings. http://www.cert.org/archive/pdf/2004eCrimeWatchSummary.pdf, 2004.

[13] Damiani, M. L., Bertino, E., Catania, B., and Perlasca, P. Geo-rbac: A spatially aware rbac. *ACM Transactions on Information Systems and Security* (2006).

[14] Fokoue, A., Srivatsa, M., Rohatgi, P., Wrobel, P., and Yesberg, J. A decision support system for secure information sharing. In *Proceedings of the 15th Symposium on Access Control Models and Technologies (SACMAT)* (2009).

[15] INFOSEC Research Council (IRC). Hard problem list, 2005.

[16] Johnson, M. E., Goetz, E., and Pfleeger, S. L. Security through information risk management. *IEEE Security and Privacy (forthcoming)*.

[17] Kamra, A., Bertino, E., and Nehme, R. V. Responding to anomalous database requests. In *Secure Data Management* (2008), pp. 50–66.

[18] Kamra, A., Terzi, E., and Bertino, E. Detecting anomalous access patterns in relational databases. *The VLDB Journal 17*, 5 (August 2008), 1063–1077.

[19] Kulkarni, D., and Tripathi, A. Context-aware role-based access control in pervasive computing systems. In *Proceedings of the 14th Symposium on Access Control Models and Technologies (SACMAT)* (2008).

[20] Laskey, K., Alghamdi, G., Wang, X., Barbará, D., Shackelford, T., Wright, E., and Fitzgerald, J. Detecting threatening behavior using bayesian networks. In *Proceedings of the Conference on Behavioral Representation in Modeling and Simulation* (2004).

[21] Liu, A., Martin, C., Hetherington, T., and Matzner, S. A comparison of system call feature representations for insider threat detection. In *Proceedings of the 2005 IEEE Workshop on Information Assurance* (June 2005).

[22] Mao, Z., Li, N., Chen, H., and Jiang, X. Trojan horse resistant discretionary access control. In *Proceedings of the 15th Symposium on Access Control Models and Technologies (SACMAT)* (2009).

[23] Ni, Q., Lobo, J., Calo, S., Rohatgi, P., and Bertino, E. Automating role-based provisioning by learning from examples. In *Proceedings of the 15th Symposium on Access Control Models and Technologies (SACMAT)* (2009).

[24] Organization for the Advancement of Structured Information Standards (OASIS). eXtensible Access Control Markup Language (XACML). http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml/.

[25] Science, C., and Telecommunications Board, N. R. C. Cyber-security and the insider threat to classified information, 2000.

[26] Sentz, K., and Ferson, S. Combination of evidence in dempster-shafer theory. Tech. Rep. SAND 2002-0835, SANDIA, April 2002.

[27] Seo, Y.-W., and Sycara, K. Addressing insider threat through cost-sensitive document classification. In *Terrorism Informatics*, vol. 18. Springer US, 2008, ch. 21, pp. 451–472.

[28] Shavlik, J., and Shavlik, M. Selection, combination, and evaluation of effective software sensors for detecting abnormal computer usage. In *KDD '04* (August 2004).

[29] Skillicorn, D. B. Individual and collective analysis of anomalies in message traffic. In *Terrorism Informatics*, vol. 18. Springer US, 2008, ch. 20, pp. 425–450.

[30] Smaldone, S., Ganapath, V., and Iftode, L. Working set-based access control for network file systems. In *Proceedings of the 15th Symposium on Access Control Models and Technologies (SACMAT)* (2009).

[31] Smalley, S., Vance, C., and Salamon, W. Implementing selinux as a linux security module. Tech. rep., NSA.

[32] Symonenko, S., Liddy, E. D., Yilmazel, O., Zoppo, R. D., Brown, E., and Downey, M. Semantic analysis for monitoring insider threats. In *The Second NSF/NIJ Symposium on Intelligence and Security Informatics (ISI2004)* (2004).

[33] The MITRE Corporation. Understanding the insider threat. In *Advanced Research and Development Activity (ARDA) Workshop* (March 2004).

[34] United States National Security Agency (NSA). Security-Enhanced Linux (SELinux). http://www.nsa.gov/research/selinux/index.shtml.

[35] United States Secret Service and CERT Coordination Center. Insider threat study: Illicit cyber activity in the banking and finance sector. http://www.secretservice.gov/ntac/its_report_050516.pdf, May 2005.

[36] Wright, C., Cowan, C., Smalley, S., Morris, J., and Kroah-Hartman, G. Linux security modules: General security support for the linux kernel. *Foundations of Intrusion Tolerant Systems* (2003).

[37] Yilmazel, O., Symonenko, S., Balasubramanian, N., and Liddy, E. D. Leveraging one-class svm and semantic analysis to detect anomalous content. In *Terrorism Informatics*, vol. 18. Springer US, 2008, ch. 19, pp. 407–424.

[38] Zhao, X., and Johnson, M. E. Access flexibility with escalation and audit. In *WISE 2008: Twentieth Workshop on Information Systems and Economics* (2008).