Insider attack and real-time data mining of user behavior

G. F. Anderson
D. A. Selby
M. Ramsey

Early detection of employees' improper access to sensitive or valuable data is critical to limiting negative financial impacts to an organization, including regulatory penalties for misuse of customer data that results from these insider attacks. Implementing a system for detecting insider attacks is a technical challenge that also involves business-process changes and decision making that prioritizes the value of enterprise data. This paper focuses primarily on the techniques for detecting insider attacks, but also discusses the processes required to implement a solution. In particular, we describe a behavior-anomaly-based system for detecting insider attacks. The system uses peer-group profiling, composite feature modeling, and real-time statistical data mining. The analytical models are refined and used to update the real-time monitoring process. This continues in a cyclical manner as the system self-tunes. Finally, we describe an implementation of this detection approach in the form of the IBM Identity Risk and Investigation Solution (IRIS).

Introduction

The problem of insider attack, also called insider misuse, involves a type of computer security threat that has been studied for many years. Early work by Anderson [1] describes the nature of insider attack and classifies perpetrators of insider attacks into three groups: masqueraders, legitimate users, and clandestine users. Masqueraders are users who obtain the login credentials of legitimate users and use these credentials to improperly access enterprise applications and data. People classified as legitimate users are those who have authorized access to enterprise computing resources but who may misuse their access privileges to download excessive amounts of information or view information not needed for performing their job duties. Finally, clandestine users gain administrative access privileges beyond or even unrelated to what they need for their job duties. Clandestine users typically have knowledge of the enterprise security systems and bypass those systems to access information. The solution discussed in this paper focuses on detection of insider attacks by masqueraders and legitimate users. Phyo and Furnell [2] provided an interesting alternative method of classifying and detecting insider misuse by considering the computing infrastructure level (e.g., network, system, and application levels) at which misuse can be identified. Other extensive prior research has been conducted in the areas of systems, algorithms, and techniques for detecting insider misuse. A seminal paper by Denning [3] develops a model for a real-time intrusion detection system, discusses various detection approaches including the one that our solution uses, and provides a fundamental framework for developing intrusion detection systems. Kumar [4] provides an excellent study of the nature of intrusions and detection techniques. A substantial literature, including a paper by Lazarevic et al. [5], also exists that discusses the evaluation of techniques used for intrusion detection, including the statistics-based anomaly detection discussed in this paper.

Enterprises spend much of their computer security budget on preventing attacks from outside hackers who are either attempting unauthorized access or introducing malignant code such as worms and viruses into the enterprise. Insider attacks are more difficult to identify and block than outsider attacks because they occur inside the enterprise firewall by users who appear to be trusted

0018-8646/07/$5.00 © 2007 IBM

after passing through standard authentication and authorization processes. Developing a defense against insider attack must strike a balance between simplifying access to help user productivity and implementing a reasonable level of security. Legal and ethical privacy issues exist when an insider attack solution is implemented, although those issues are outside the scope of this paper. One way to minimize the impact on users is to perform a post-analysis of the log files created by security monitor applications, middleware programs, application servers, and other enterprise applications. This technique detects the insider attacks after the data has been accessed rather than as they are happening.

A study [6] conducted by PricewaterhouseCoopers and *CIO Magazine* showed that 33 percent of information security attacks originated from internal employees, while 28 percent came from ex-employees and company partners. A survey [7] conducted by the *Chief Security Officer* (*CSO*) magazine found that 22 percent of the responding organizations had experienced critical system disruption to their organization as a result of an insider attack, with seven percent responding that the incidents had resulted in loss of customers.

In this paper, we discuss the type of insider attack in which employees of an enterprise engage in a pattern of resource-access behavior that exceeds what is necessary for their employment duties. That is, while employees may have general authorization for accessing specific applications and data in the course of performing their jobs, they access excessive amounts of data or data that is unrelated to their assigned tasks. The *CSO* magazine study [7] found that authorized users with valid accounts carried out 78 percent of insider attacks, and in 43 percent of the cases, the persons used their own userids and passwords while accessing the data. Although we are focusing on employee behavior, the techniques we describe apply to business partners of an enterprise who have access rights to sensitive applications and data. Examples of insider misuse reported recently include the following:

1. An account manager changed the address of an account he managed, had a new credit card and PIN sent to his own address, and then used the card to withdraw money from the credit card account [8].
2. A company selling personal information to other companies and government agencies had large amounts of data accessed in an unauthorized manner by users who had set up fake companies in order to appear to be legitimate customers [9].
3. A former help desk employee at a communications company pleaded guilty to a scheme to steal and sell 30,000 consumer credit reports of customers of that company [10].

Our solution approach

We describe a system for early detection of insider attacks. This "closed-loop" system analyzes historical data in order to determine peer-group access patterns, initializes our real-time data-mining component with the historical baselines for the data, and then monitors user-behavior statistics in real time. The information collected during real time then updates the historical analytical tool, allowing refinement of the behavior models that in turn update the real-time monitoring component, and so forth in a cyclical self-tuning manner.

Our system operates by developing peer groups—that is, groups of users with similar sets of characteristics. Such groups are derived from the investigative staff's experience and knowledge of the population and from data-mining techniques, which we explore shortly.

An example of a peer group is a group of customer service representatives working at an enterprise help desk. We anticipate that people in similar job functions will have similar access patterns to enterprise applications and similar levels of demand on the systems for information. Other personal and/or employment-related information, such as job experience, geographical location, or personnel ratings, might be used when determining how to classify a specific user according to a peer group. Additionally, such information can be supplemented by knowledge of the business processes with which each type of user is involved. System operators may profile clusters of users as peer groups, and such operators are the only people who actually know how peer groups are constituted and used in the analysis. Therefore, an individual, in order to avoid the type of behavior that appears anomalous to our system, would have to a) know which peer group he or she is in and b) ensure that all of the others in that group appear to behave in the same way as the individual in question. Clearly, this makes it difficult for potential system abusers to defeat the system unless they collude with the operator of the solution.

The IRIS (IBM Identity Risk and Investigation Solution) insider threat solution combines a real-time data mining appliance with the behavioral modeling and analysis capabilities of the our entity profile management system (EPMS), which applies advanced analytics to score transactions for various kinds of risk. Our approach begins with an analysis of historical access information from log files generated by portals, application servers, operating systems, and databases. Higher-level user-access information can be gathered from event-consolidating solutions such as the Common Audit Reporting System (CARS), which is a part of the Tivoli* Access Manager (TAM) product family, or from transaction recording and correlation tools such as those available from Intellinx [10]. The type of information we gather for analysis depends on the specific security issues

that the enterprise wishes to address. Examples of access-related attributes that we may monitor are the following:

- The number of accesses to a specific application, such as those involving personnel records.
- The time between accesses to a particular application.
- The number of sensitive data items that have been accessed, such as social security number or date-of-birth information.
- Other user login characteristics such as time of day or the use of remote or on-site access.

We can also create features, or attributes, from more complex derivations involving sequential access patterns or location-based behavior. A feature is essentially a data point for a user that quantifies behavior such as the number of accesses or the average number of accesses per session. We summarize transactions over a long period of time, such as a year, by aggregating this historical access information to determine a single value that characterizes a behavioral feature for a user. Our multi-year experience of analyzing data for anomalous behavior across several domains suggests that historical data be collected for a minimal period of one to three years, depending on the resource available for data analysis. Individual information access events, such as logins or application accesses, are aggregated for time periods that are deemed appropriate for each attribute. These periods may relate to a login session, a business day, or a particular hour.

Retrospective modeling

After peer groups have been defined and a set of initial data features has been determined, the aggregated historical data is divided into datasets for each peer group and imported into the IRIS analysis environment. The IRIS workbench provides an analyst with the ability to construct a series of insider-attack hypothesis models. Hierarchical combinations of the imported features, along with scoring functions and feature weighting, define these models. In addition, the analyst can computationally derive new features from the base-level imported features. Base-level features, such as the age of an individual, are derived from unprocessed data and not from the results of mathematical computations that combine more than one feature. A set of visual data-exploration and data-mining tools enable model refinement and allow the analyst to refine peer-group definitions.

Figure 1 shows the distribution across a peer group of one of the features monitored by our solution. In this example, the "feature" is the number of accesses to a software application or similar resource. The distribution illustrates accesses across one thousand users grouped by the number of accesses to an application that retrieves employee performance data. The *x*-axis is the number of accesses per period, and the leftmost *y*-axis is the number of individuals corresponding to those accesses. This is a hypothetical distribution generated for illustrative purposes. We may consider users at the upper and lower tails of the curve as potentially exhibiting anomalous behavior, although an understanding of the business domain is required to determine whether high, low, or a combination of high and low values for this feature are indicative of bad behavior. In the example in Figure 1, the distribution across the peer group is roughly Gaussian. The red line in the diagram is a scoring curve that indicates the number of accesses that are higher than the mean and should thus be considered suspicious. In our retrospective system, the scoring of each feature allows us to scale the "badness" of the underlying data to a range from 0 to 1,000, with higher scores indicating increasingly undesirable behavior. (Only values for a feature that occur under the red curve are scored in this range.) In general, the distribution of users across each attribute, combined with the domain knowledge of the meaning of outlier values, allows us to give a user a score and ranking relative to other users. The nature of the scoring curve depends on the shape of the distribution curve and the significance attached to the outlier values. Our system has a number of standard scoring curves based on our implementation experience. We can also customize the scoring curve for each individual attribute.

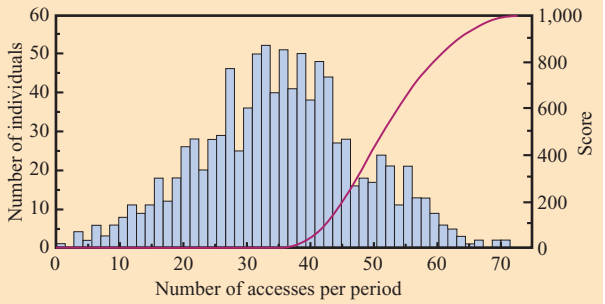

Figure 1

Distribution of accesses to an employee personnel application within the peer group. See text for a discussion of the score at the right.

Figure 2 shows a simple clustering analysis across our six example attributes. The results of this analysis can provide some insight into whether or not the attributes available from the data source are suitable for anomaly detection for this peer group. The workbench system for analysis of historical data includes several data-mining techniques, such as this cluster analysis, that might be used to dynamically locate peer groups and suggest refinements to our anomaly hypotheses.

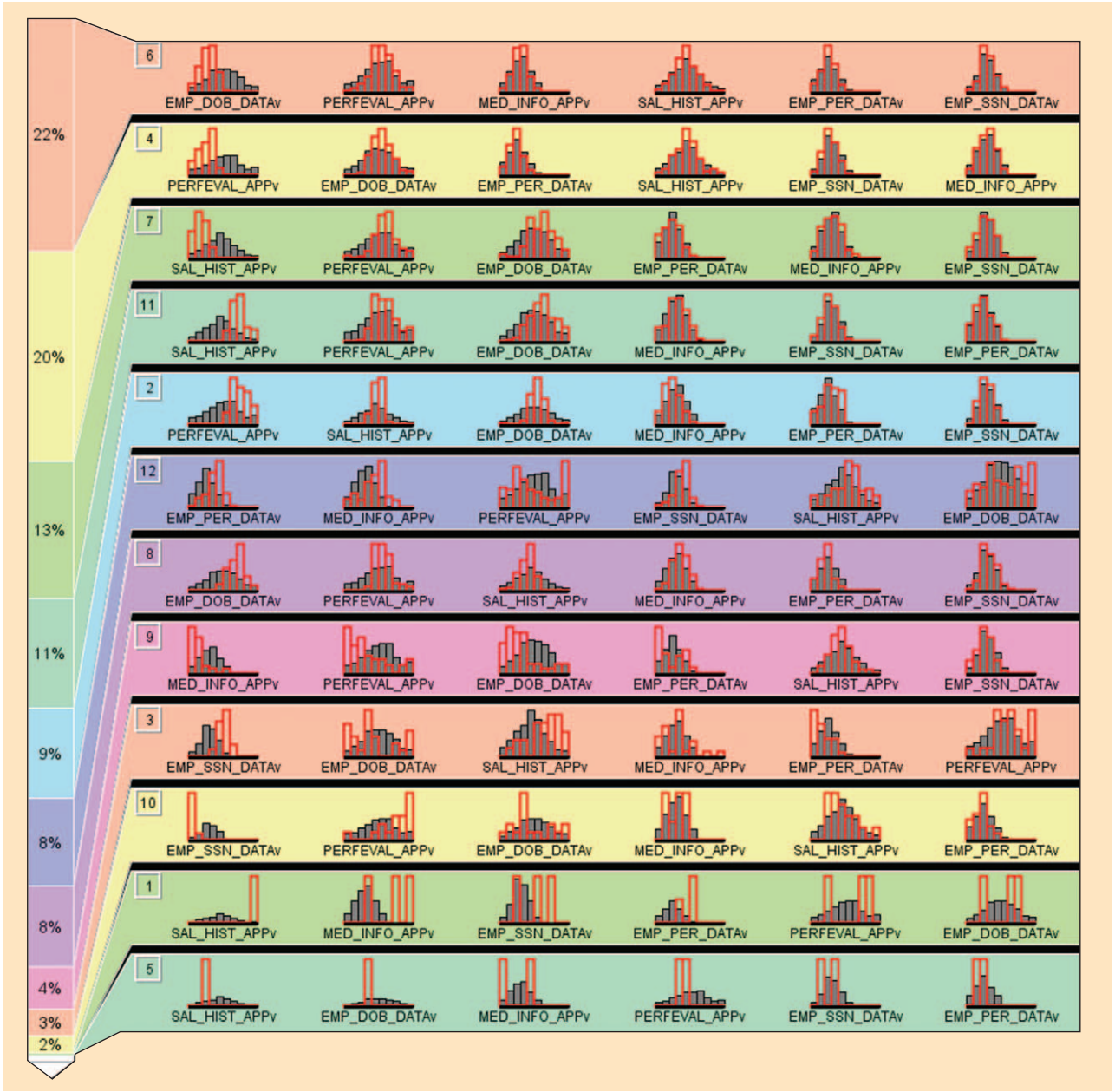

Figure 2

Cluster analysis of features. Twelve clusters are shown for six features such as the number of accesses of information relating to employee serial numbers or employee performance evaluations.

Figure 2 shows 12 clusters from the candidate records. The size of the segment of the user population, represented in the horizontal colored segments, is denoted on the left side of the segments in percentage terms. Each small histogram in the chart consists of two superimposed histograms. The gray background histogram corresponds to the population within the total population being examined; the red histogram represents the characteristics of this population—that is, the distribution of this feature across the population in the cluster. The variables (i.e., features) are also sorted in order of significance on the basis of entropy for each feature in a cluster, which is a measure of information value. Therefore, variables that have more weight or influence on the makeup of the cluster are found to the left of the chart. For example, EMP_DOB_DATAv is more important in determining

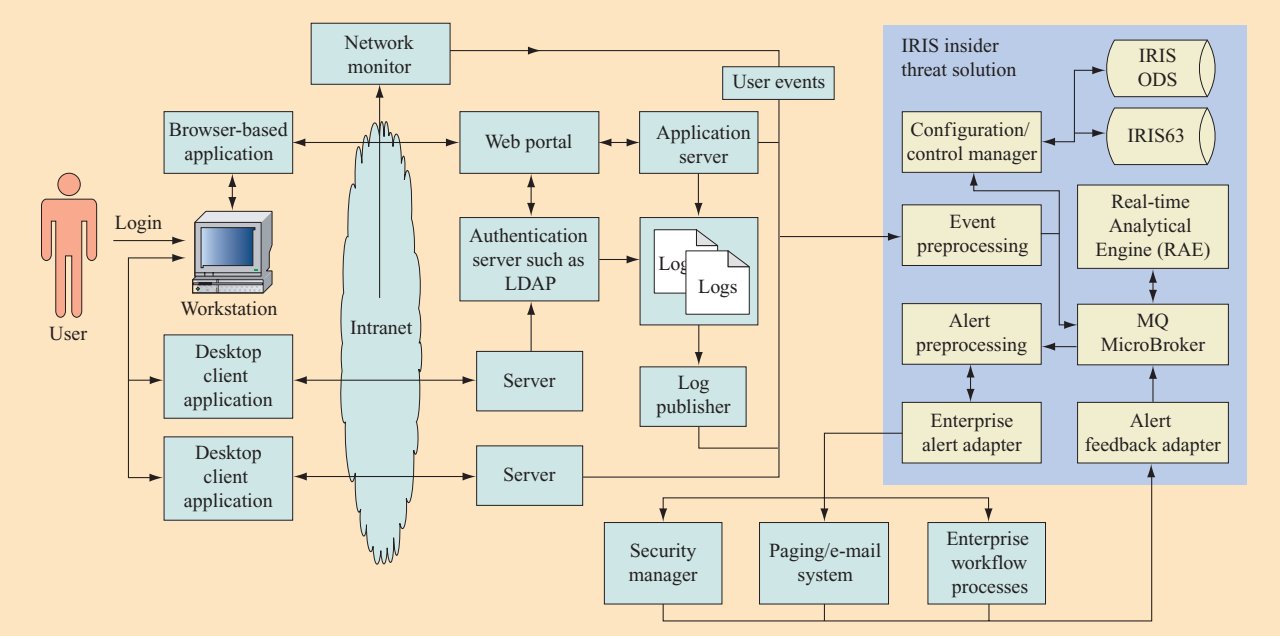

Figure 3

Example of integration architecture. (LDAP: Lightweight Directory Access Protocol; MQ: message queuing; ODS: operational data store; IRIS63: a database and software version for the IBM Entity Profile Management System.)

the placement of individuals in the uppermost cluster than MED_INFO_APPv.

Transitioning from retrospective to real-time analysis

The retrospective analysis techniques discussed in the previous section allow an analyst who understands the enterprise processes and data to develop hypotheses of behavior that indicates anomalous or improper access to enterprise data. Our analytical workbench allows analysts to verify the hypotheses by loading additional datasets and to use the hypothesis models they have developed to calculate risk scores for the users in this new data. In fact, the analytical workbench can be used to process new datasets periodically if the intent is to detect anomalous behavior from historical data. However, as we discussed at the outset, insider attacks are a significant financial threat to the enterprise and must be detected as closely as possible to the occurrence of the incident. To accomplish this early detection, we have developed a real-time data-mining solution targeted specifically at insider attacks. This solution builds on the information gained from the retrospective analysis.

Insider threat solution architecture

Figure 3 shows an example of the IRIS insider threat solution integrated with an extremely simplified enterprise infrastructure. This illustrates that the events related to a user's behavior may originate from diverse applications and via a variety of delivery channels. In this example, we have shown input sources such as an application server, either providing a user event directly or via a log file, client-server applications sending user audit events directly, and a network monitor component that may monitor network traffic at the packet level and gather information related to the user actions from the transaction payload, that is, the contents of the transaction data stream. This network monitor approach is exemplified by software developed by Intellinx, which intelligently extracts information on user activity from traffic between a web server and browser client, or a mainframe application screen and a mainframe server. Given the appropriate application adapters, this model can easily be extended to very diverse event sources such as radio frequency identification (RFID) sensors that provide location awareness. Our reference implementation, that is, our IRIS insider threat solution, is shown in the simplified architectural representation in the inset in Figure 3. With the exception of the Real-time Analytical Engine, which is discussed in detail in the next section, our solution components are listed in the following short sections.

Event preprocessing

Because we handle events from a variety of sources, our solution attempts to limit the computational and communication-bandwidth burden on the enterprise

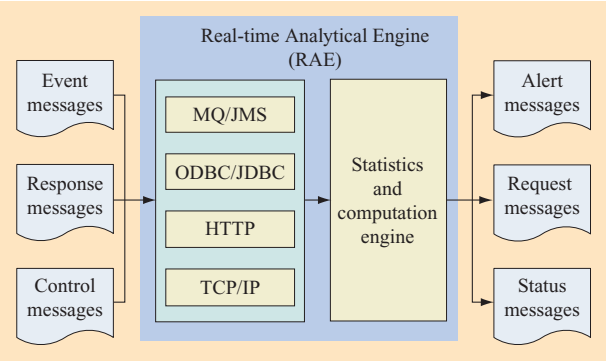

Figure 4

Real-time Analytical Engine component. (MQ: message queuing; JMS: Java message service; ODBC: Open database connectivity; JDBC: Java database connectivity.)

infrastructure by performing some preprocessing on events prior to submission to the event monitor. Key tasks performed by this preprocessing component, which may be implemented as a set of distributed components, include correlating user IDs with diverse event sources and normalizing the structure of the messages to simplify handling by the event-monitoring component.

Component integration hub

The components in our IRIS solution are independent and loosely coupled, making use of the MQ Telemetry Transport protocol (MQTT) implemented by the IBM MQ MicroBroker. (MQ stands for *message queuing*.) The MQ MicroBroker is a high-performance, small-footprint publish/subscribe message broker that allows our solution components to communicate via XML-based messages using a number of concurrent topic channels. It also allows multiple components in the solution to monitor topics (i.e., message categorizations) and take independent but complementary action. Although not shown in our diagram, this technology allows our solution to scale to large numbers of messages, and hence users, by distributing the responsibility for event preprocessing or allowing the event monitoring to be performed by multiple instances of the components that may be running on different servers or even in different physical locations across the enterprise.

Configuration/control manager

Because the RAE component is a general-purpose processing engine, it requires a controller that provides initialization information and configures it with data such as peer group membership, attributes to be monitored, and attribute scoring and aggregation models developed during the historical data mining and hypothesis modeling processes discussed earlier.

Business logic processor

While the basic event monitoring is performed by the RAE, our solution provides the capability to analyze the alerts generated by the RAE in greater depth using a business logic processing module. This component can gather additional information on the user in question from the operational data store (ODS) associated with the IRIS analytical client or other data sources, and it can apply business rules and/or extended calculations. It may also be used to analyze the cumulative alerts that have been issued by the RAE with respect to that user, or to identify some type of insider attack being carried out by multiple users.

Enterprise alert adapter

This component provides transformation and routing capabilities for publishing the RAE alerts to enterprise systems, such as workflow, e-mail, and paging gateways, as well as the access management infrastructure.

Alert feedback adapter

Much like the enterprise alert adapter, this component connects to processes and/or components in the enterprise infrastructure in order to accept feedback on the alerts that were issued by the RAE. As we discuss later, the RAE component uses this feedback to learn and adapt to user behavior that triggers an alert that a knowledge worker in the enterprise determines to be acceptable. In our reference implementation of this solution, the feedback component is handled by a Java**-enabled telephone that receives alerts via an application called Universal Inbox, which is itself an application built around MQTT and the MQ MicroBroker.

Real-time analytical engine

The real-time data-mining engine should be viewed as a software appliance that can be plugged into the enterprise infrastructure. **Figure 4** illustrates the input interfaces and message input/output (I/O) at a high level. The engine presents various software interface "facades" to the enterprise infrastructure, and data submitted to the statistics engine is made consistent across all input channels. It is expected that each of the event messages will contain information related to one of the monitored attributes and will include a user-identification token as well as a peer group association token. The peer group association token is optional in cases in which the user is already being monitored by the engine. In our reference implementation, given a user-identification token, an associated component is available to provide the peer group association. Since RAE is a general computation

engine, it has the capability to make such synchronous requests.

The RAE provides a general-purpose processing framework that has been tailored for use in our IRIS insider threat solution. As shown in **Figure 5**, the component level of the RAE includes the following:

1. A variety of input channels to provide flexibility for integration with the enterprise (upper left in Figure 5).
2. A command interpreter that can be configured for specific analytical domains.
3. A work queue that can interleave tasks originating from all of the input channels.
4. An embedded Array Processing Language and high-performance mathematical calculation engine optimized for large array-oriented datasets [11].
5. An in-memory database configurable dynamically and with a persistence mechanism (i.e., a means for placing data in nonvolatile storage).

At an implementation level in the context of our anomaly-detection solution, the engine is tuned (i.e., customized) to the domain by code and configuration properties that encapsulate the expected message and data structures that are used in the data mining of user behavior. The objective of this environment from a services perspective is to facilitate flexible and easy changes to both mathematical transformations and domain-specific functionality.

Data mining user behavior in real time

IRIS, which is our reference implementation of the insider-attack solution, uses the RAE to monitor user access in real time and trigger alerts based on either statistics related to individual attributes or scores derived from calculations made across the multiple attributes of our analytical models. In many ways, the RAE performs a triage function related to insider attack in that it attempts to compare and group events on the basis of the need for, or likely benefit from, immediate attention.

Assessment calculations and triggering

Because the RAE data mining uses time-based statistical features, the concept of analyzing events in discrete time slices is an important one. We are concerned with the time between analytical assessments for each of the monitored attributes. In general, we might anticipate that all attributes would be assessed after the same interval of time instead of independently, although the timing of the assessment can be controlled independently for each attribute. At intervals determined either by the nature of the assessment interval for that attribute or the configuration of the specific implementation of IRIS,

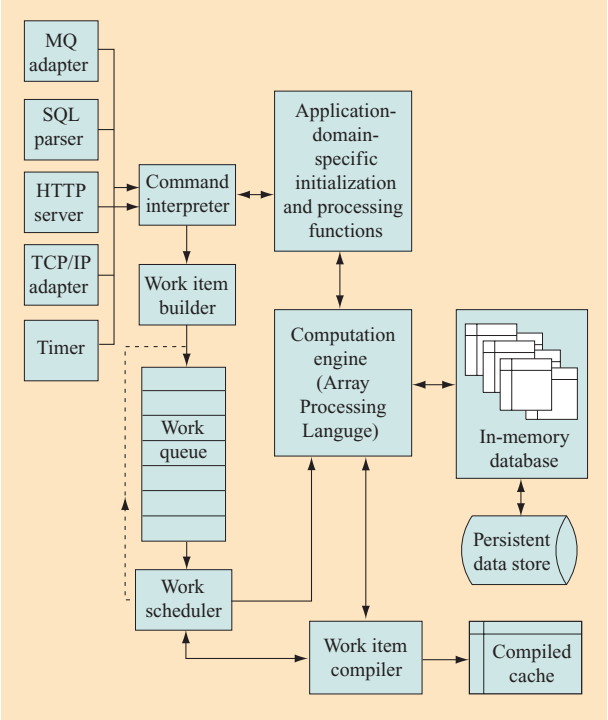

Figure 5

RAE internal architecture. (SQL: Structured Query Language.)

a calculation is made to determine the deviation of the user's data from the peer group baseline. In addition, the data across all attributes being monitored for that user are scored using our analytical model. The results of these calculations are then analyzed to determine whether an alert will be issued. The criteria for issuing an alert include one or more of the following:

1. The statistics associated with an individual attribute deviate by more than a specified amount from the peer group baseline.
2. The score calculated for an individual attribute exceeds the specified threshold.
3. The aggregate score across all of the attributes being monitored exceeds a specified threshold.
4. The extrapolated statistics for an attribute exceed some expectation. (For instance, if historical data suggests that 20 records are normally accessed over a 24-hour period, the user accessing 15 records in the first five minutes may be an indication for an alert.)

The RAE is an analysis and detection engine; it does not suggest remedial action or integrate directly with enterprise workflow. Once an alert is issued, the responsibility for handling the alert is transferred to

components handling the integration with the enterprise infrastructure. However, the RAE does expect to receive feedback that is associated with the alert and uses this feedback as a learning and adjustment mechanism for the detection process. The statistics associated with the user's accesses are unchanged until the knowledge worker or other responsible individual accepts or rejects the alert.

Learning and adaptation by the real-time data mining

If, at the end of a time-slice assessment, the user statistics are within the current allowable boundaries, the data from this time slice is added to the continually changing tally for that user, effectively modifying the baseline against which the next assessment will be done. This allows the system to adapt to gradual changes in a user's behavior and is an example of the learning and adaptation that is built into the real-time data mining. Once an alert is issued from the real-time monitoring system, the system flags the user and continues to accumulate data for the user and issue additional alerts. Each alert is placed in a pending area awaiting an asynchronous response to the alert through the message-based input mechanism from the enterprise assessment tools. If the response is to ignore the alert, the statistics that generated the alert are incorporated into the baseline statistics for that user, allowing the system to adjust for the approved changes in behavior. Since this mechanism is itself a potential area of fraud and/or abuse, an audit trail is kept for these events.

Our solution provides two approaches to minimizing the impact of false positives with respect to intrusion alerts. The first includes the functionality described above, in which a knowledge worker responds to alerts, and if the user behavior is deemed acceptable, the monitoring tool automatically adjusts for future monitoring. The second feature that actually minimizes the number of false positives is the ability to configure the alert thresholds in the tool. This allows the monitoring to start at a more tolerant level and be tuned until an acceptable balance between false positives and false negatives is reached.

System performance

Performance measurements in this distributed system are extremely dependent on the deployment environment. Two key performance measurements are the number of raw events that can be processed per second, and the number of users and features that can be monitored concurrently. Because our monitoring data is resident in memory (although with a persistence mechanism for recovery in times of system failure) and we are maintaining a very small data footprint for each user, our system can monitor several hundred thousand to one million users, even using low-end server hardware found in many enterprises. In terms of event processing, one benchmark on a system consisting of an IBM POWER5* pSeries* (running AIX* 5.3 and with 4 GB of memory), shows that the system processed approximately ten thousand user events per second.

Implementing the IRIS insider threat solution

We have described components and technology that form the basis of our IRIS insider threat solution. This solution always involves integration with both the information technology processes of an enterprise and the business processes and policies that govern access to enterprise information. **Figure 6** shows the typical steps needed to achieve success in implementing an insider threat solution using our technology. Each of these steps is discussed below.

Assessment

This step involves an overall assessment and prioritization of the data assets to be monitored and protected. The participation of an individual with strong knowledge of the business value and regulatory issues associated with a broad spectrum of business data is important for this task. In addition to protecting the enterprise data from internal user misuse, the enterprise may decide that usage of other computing resources must be monitored. Here, the enterprise user base is assessed and priority areas determined for the types of users that have authorization to access the critical data. In other words, the enterprise must prioritize the types of employees that should be monitored on the basis of such concerns as access to sensitive data.

Identification

Suitable sources of user access events are identified that can be monitored by the system. Some initial determination is made regarding the feasibility of publishing these events in a suitable form and timeframe. An initial scheme for classifying users into peer groups is developed. Peer group members should exhibit similar patterns of information access in terms of both frequency and quantity. This scheme is refined and augmented during the analysis and data-mining phase of the process.

Data collection

Historical data access patterns for the peer groups are collected across the information domain to be monitored. The information domain may include database log files, login audit records, or any files that indicate user access to enterprise data. If this information is not available, it can be collected during a training phase. Additional information on users is collected (e.g., job role, length of employment, and other factors that might be useful

in developing peer groups and performing rule-based in-depth evaluation of the alert messages).

Analysis (historical data mining)
Historical data is analyzed in order to determine normal values and the type of distribution that is typical for each attribute. Hypotheses are developed with respect to patterns of behavior that may constitute anomalous behavior indicative of insider attack. Multiple-variant modes are developed for purposes of risk scoring.

Design and operation
The attributes to be monitored are determined (e.g., number of accesses to an application per time period, number of sensitive data records retrieved, time of day for login, and login location). A workflow process is designed and developed for evaluating and handling the alerts issued from the real-time data-mining monitoring system. This may involve some business process transformation and development of new policies and audit procedures.

Integration
The sources for generating access events across the enterprise are identified and integrated. The real-time data-mining appliance and its infrastructure are connected to the event stream and the workflow processes that monitor the alert.

Execution
User access events are monitored and alerts issued on the basis of statistical analysis and multiple-variant scoring of user behavior. The alerts generated are reviewed by the monitoring processes, and the behavior is checked to see whether it is within normal boundaries.

Validation analysis

The approach used by our insider threat solution creates a challenge in terms of measuring such parameters as false-positive rates, although false-negative rates should be inherently minimal due to use of peer groups. In our system, abnormal behavior is detected by comparison with a large base of historical user activity. When current monitored activity deviates from this historical base, it may mean that the user is improperly classified as a member of the peer group with which he or she is compared, or the user is properly classified in a peer group but is currently behaving in an anomalous manner. If the behavior corresponds to the second category, it is either acceptable behavior that can be explained in the context of the business process, or it is anomalous behavior that is improper and must be stopped. The alert feedback process determines which case is true, and this result allows the system to self-tune.

To properly determine the occurrence of false positives and false negatives, we require a suitable reference dataset. While existing datasets for evaluating intrusion attacks exist, we have not found datasets that are designed to model the type of user behavior in terms of peer groups and deviations from peer groups which would be required to test our detection strategy. Insider

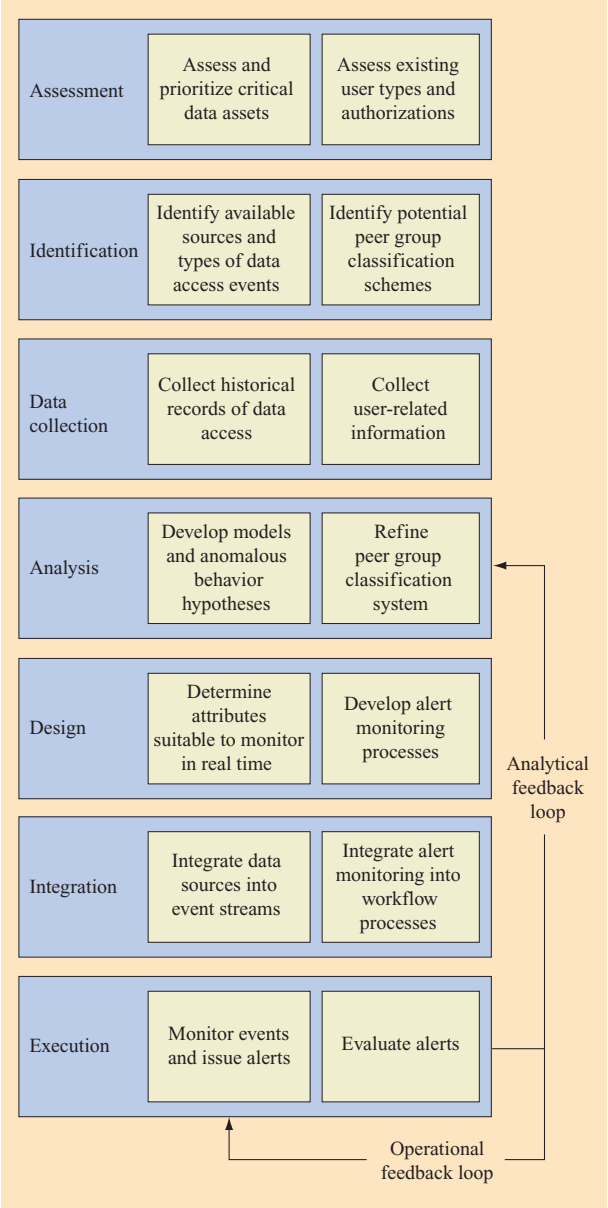

Figure 6

IRIS implementation process. The steps indicated in the boxes are performed in sequence from top to bottom. The operational feedback loop involves the alert response that indicates when a user behavior is acceptable even though an alert has been issued. The analytical feedback loop refers to the re-performing of the historical analysis to make the system more accurate on the basis of recent experience gained from the real-time mining.

attack, unlike external attack, is an emerging area of investigation, and for this reason we have also discovered no commercial applications that implement our strategy of behavior anomaly detection for insider attacks. Note that our approach is not designed to work on heterogeneous user data records in the absence of peer group analysis and business domain modeling efforts.

We have in fact built a coherent set of test data that simulates six thousand users in two peer groups that contain a small set of users who are behaving anomalously. This dataset has been analyzed with our retrospective analytics in order to determine the users who would be expected to be detected as anomalous users. When data for these users was subjected to our real-time detection system, it in fact triggered the expected alerts. However, this may be considered more of a system or regression test than an effectiveness test. Because of our particular approach to insider threat detection, the effectiveness (e.g., as measured by false positives and false negatives) of the system is fully determined by the quality of the retrospective analysis and peer group classification.

Although the nature of our approach does not at this point easily lend itself to determining its inherent effectiveness, the real-time, analytical engine self-tuning based on alert feedback from knowledge workers suggests that the system will immediately start reducing the false-positive rates. False-negative rates are somewhat mitigated by the peer group classification approach that we have adopted. If a user is properly classified into a peer group and behaving within the expected statistical bounds, it seems unlikely that that user is involved in insider attack behavior. Of course, accuracy is dependent on the system monitoring a comprehensive set of user behavior. In summary, the effectiveness of our solution is primarily affected by the quality of the historical data supplied, the domain knowledge used in the behavioral modeling, and the accuracy of the peer group classification process.

Conclusion

An effective technique for combating insider threats requires a combination of historical user access analysis, modeling to support the development of peer group statistics, and real-time proactive monitoring of user behavior. Results of the real-time monitoring must be processed using a combination of an automated rule-based approach and monitoring by a knowledge worker. We have demonstrated a set of technologies that can accomplish this process.

Detecting insider misuse of enterprise computing assets and data is critical in the current business environment. Developing an effective detection system requires a cooperative effort from the business, security, and technical parts of an organization. A business analysis function determines what is normal behavior for a group of users. Integrating the tools necessary for implementing a system that detects deviation from that expected behavior is a technical solution integration problem. Monitoring the results of this detection system and taking action is a problem for an operational security organization that may encompass a mix of business and technical workers. While the implementation of these technologies and the process changes that must accompany them are not inexpensive, the financial and business cost of insider attacks is substantial and cannot be ignored, nor can it be dealt with by using traditional computer security policies and techniques.

*Trademark, service mark, or registered trademark of International Business Machines Corporation in the United States, other countries, or both.

**Trademark, service mark, or registered trademark of Sun Microsystems, Inc. in the United States, other countries, or both.

References

1. J. P. Anderson, "Computer Security Threat Monitoring and Surveillance," Technical Report, James P. Anderson Company, Fort Washington, PA, April 1980; see *http://csrc.nist.gov/publications/history/ande80.pdf*.
2. A. H. Phyo and S. M. Furnell, "A Detection-Oriented Classification of Insider IT Misuse," *Proceedings of the Third Security Conference*, Las Vegas, NV, April 14–15, 2004.
3. D. E. Denning, "An Intrusion Detection Model," *IEEE Trans. Software Eng.* **13**, No. 2, 222–232 (1987).
4. S. Kumar, "Classification and Detection of Computer Intrusions," Ph.D. Dissertation, Purdue University, Lafayette, IN, August 1995.
5. A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava, "A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection," *Proceedings of the Third SIAM Conference on Data Mining*, San Francisco, CA, 2003; see *http://www.siam.org/meetings/sdm03/proceedings/sdm03_03.pdf*.
6. S. Berinato, "The Global State of Information Security 2005," September 15, 2005; see *http://www.cio.com/archive/091505/global.html*.
7. *CSO* magazine,"2005 E-Crime Watch™ Survey, Summary of Findings"; see *http://www.cert.org/archive/pdf/ecrimesummary05.pdf*.
8. M. Randazzo, M. Keeney, E. Kowalski, D. Cappelli, and A. Moore, "Insider Threat Study: Illicit Cyber Activity in the Banking and Finance Sector," Carnegie Mellon University Software Engineering Institute, *Technical Report CMU/SEI-2004-TR-021*, 2004; see *http://www.sei.cmu.edu/publications/documents/04.reports/04tr021.html*.
9. A. Litan, "ChoicePoint Fraud Case Shows Need for Security Controls," February 22, 2005, *ID No. G00126448*, Gartner, Inc.; see *http://www.gartner.com/resources/126400/126448/choicepoint_fra.pdf*.
10. D. Yachin, "Combating Insider Threats: The Application-Level User Behavior Tracking Approach," IDC White Paper sponsored by Intellinx, 2006; see *http://www.intellinx-sw.com/pdf/IDC_White_Paper_Combating_Insider_Threats.pdf*.
11. J. A. Brown, "A Development of *APL2* Syntax," *IBM J. Res. & Dev.* **29**, No. 1, 37–48 (1985).

Received September 15, 2006; accepted for publication October 15, 2006; Internet publication May 23, 2007

Gary F. Anderson *IBM Global Business Services, Center for Business Optimization, 150 South Kettletown Road, Southbury, Connecticut 06488 (gfa@us.ibm.com.)*. Mr. Anderson received his B.S. degree in chemistry from the University of Washington in 1970. He joined the IBM Insurance Research Center in 1995, working on leading-edge Internet-based solutions for the insurance industry. He has been awarded several patents in the areas of distributed computing and Internet-based solutions. Mr. Anderson is currently a solutions architect, working in anomaly-based detection systems focused on the areas of tax fraud, insurance fraud, and insider misuse of enterprise data.

David A. Selby *IBM Global Business Services, Center for Business Optimization, Hursley Park, Winchester, Hampshire SO21 2JN, England (David_Selby@uk.ibm.com)*. Mr. Selby received a Higher National Certificate (HNC) in computer science from Portsmouth Polytechnique in 1982. He is a Fellow of the British Computer Society. Since joining IBM in 1977 in the area of manufacturing, he has held many positions, including microcode engineer, software engineer, research scientist, and consultant. Mr. Selby has been awarded several patents in the field of advanced analytics, and he was named an IBM Master Inventor in 2004. He has received an IBM Outstanding Innovation Award for sustained innovation in real-time analytics. Mr. Selby is currently a solution architect for both the real-time fraud and abuse solutions and the IBM Marketing Event Optimization Solution.

Mark Ramsey *IBM Global Business Services, Center for Business Optimization, 1507 LBJ Freeway, Dallas, Texas 76245 (mramsey@us.ibm.com)*. Mr. Ramsey received his B.A. degree in computer science and business administration from Capital University, Columbus, Ohio. He joined IBM in 1995 to launch a consulting and services team focused on business intelligence in the insurance industry. He has been awarded several patents in areas ranging from the deployment of data analytics into relational databases to the optimization of marketing events. Mr. Ramsey is currently the Global Data Analytics leader responsible for the development, sales, and delivery of solutions leveraging the expertise of IBM in the use of analytics and optimization to address various business challenges.