Automated Insider Threat Detection System Using User and Role-Based Profile Assessment

Philip A. Legg, Oliver Buckley, Michael Goldsmith, and Sadie Creese

***Abstract*—Organizations are experiencing an ever-growing concern of how to identify and defend against insider threats. Those who have authorized access to sensitive organizational data are placed in a position of power that could well be abused and could cause significant damage to an organization. This could range from financial theft and intellectual property theft to the destruction of property and business reputation. Traditional intrusion detection systems are neither designed nor capable of identifying those who act maliciously within an organization. In this paper, we describe an automated system that is capable of detecting insider threats within an organization. We define a tree-structure profiling approach that incorporates the details of activities conducted by each user and each job role and then use this to obtain a consistent representation of features that provide a rich description of the user's behavior. Deviation can be assessed based on the amount of variance that each user exhibits across multiple attributes, compared against their peers. We have performed experimentation using ten synthetic data-driven scenarios and found that the system can identify anomalous behavior that may be indicative of a potential threat. We also show how our detection system can be combined with visual analytics tools to support further investigation by an analyst.**

***Index Terms*—Anomaly detection, cyber security, insider threat.**

I. Introduction

THE insider threat problem is one that is constantly growing in magnitude, resulting in significant damage to organizations and businesses alike. Those who operate within an organization are often trusted with highly confidential information such as intellectual property, financial records, and customer accounts, in order to perform their job. If an individual should choose to abuse this trust and act maliciously toward the organization, then their position within the organization, their knowledge of the organizational systems, and their ability to access such materials means that they can pose a serious threat to the operation of the business. The range of possible activities could be anything from taking money from a cash register to exfiltrating intellectual property from the organization to sell on to rivals, which could effectively destroy the successful operation of the organization. Capelli *et al.* from the Carnegie Mellon University Computer Emergency Response Team (CMU-CERT) group identified three main groups of insider threat: information technology sabotage, theft of intellectual property, and data fraud [1]. There are also a growing number of cases that media attention has highlighted in recent years that reveal that both businesses and governments have suffered similar experiences, whereby top secret information has been exfiltrated and passed on to oppositions. The threat posed by the insider is very real and requires serious attention from both employees and organizations.

Over the years, technological advancements have meant that the way organizations conduct business is constantly evolving. It is now common practice for employees to have access to large repositories of organization documents electronically stored on distributed file servers. Many organizations provide their employees with company laptops for working while on the move and use e-mail to organize and schedule appointments. Services such as video conferencing are frequently used for hosting meetings across the globe, and employees are constantly connected to the Internet, where they can obtain information on practically anything that they require for conducting their workload. Given the electronic nature of organizational records, these technological advancements could potentially make it easier for insiders to attack. From the organizational view, one advantage to this is the capability of capturing activity logs that may provide insight into the actions of employees. However, actually analyzing such activity logs would be infeasible for any analyst due to the sheer volume of activity being conducted by employees every day. What is required is a capability to analyze individual users who conduct business on organizational systems, to assess when users are behaving normally and when users are posing a threat.

In this paper, we present a systematic approach for insider threat detection and analysis based on the concept of anomaly detection. Given a large collection of activity log data, the system constructs tree-structured profiles that describe individual user activity and combined role activity. Using these profiles, comparisons can be clearly made to assess how the current daily observations vary from previously observed activities. In this fashion, we construct a feature set representation that describes the observations made for each day and the variations that are exhibited between the current day and the previously observed days. This large feature set is reduced into multiple anomaly assessment scores using principal component analysis (PCA) [2] decompositions on subsets of features, to identify the degree of deviation for each grouping. The anomaly assessment

P. A. Legg was with the Cyber Security Centre, University of Oxford, Oxford OX1 3QD, U.K. He is now with the Department of Computer Science, University of the West of England, Bristol BS16 1QY, U.K. (e-mail: phil.legg@uwe.ac.uk).

O. Buckley was with the Cyber Security Centre, University of Oxford, Oxford OX1 3QD, U.K. He is now with the Centre for Cyber Security and Information Systems, Cranfield University, Cranfield MK43 0AL, U.K., and also with Defence Academy of the United Kingdom, Wiltshire SN6 8LA, U.K.

M. Goldsmith and S. Creese are with the Cyber Security Centre, University of Oxford, Oxford OX1 3QD, U.K.

scores can be used either with classification schemes to produce a list of suspicious users or can be visualized using parallel coordinates plots to provide a more in-depth view. To test the performance of the approach, a red team developed ten simulated insider threat scenarios for experimentation that are designed to cover a variety of different types of insider attacks that are often observed. It was found that the system performed significantly well for detecting the attacks using the classification alerts, and the visualization enabled analysts to identify what particular attributes caused the insider to be detected. The remainder of this paper is as follows. Section II discusses the related work. Section III describes the requirements of an insider threat detection system. Section IV presents the proposed system, describing in detail the different components. Section V presents the process of constructing effective simulation data and the experimentation of the detection system, and Section VI concludes this paper.

II. Related Work

The topic of insider threat has recently received much attention in the literature. Researchers have proposed a variety of different models that are designed to prevent or detect the presence of attacks (e.g., [3] and [4]). Similarly, there is much work that considers the psychological and behavioral characteristics of insiders who may pose a threat as means for detection (e.g., [5]–[7]). Kammüller and Probst [8] considered how organizations can identify attack vectors based on policy violations, to minimize the potential of insider attacks. Likewise, Ogiela and Ogiela [9] studied how to prevent insider threats using hierarchical and threshold secret sharing. For the remainder of this section, we choose to focus particularly on studies that address the practicalities of designing and developing systems that can predict or detect the presence of insider threat.

Early work by Spitzner [10] discusses the use of honeypots (decoy machines that may lure an attack) for detecting insider attacks. However, as security awareness increases, those choosing to commit insider attacks are finding more subtle methods to cause harm or defraud their organizations, and thus, there is a need for more sophisticated prevention and detection. Early work by Magklaras and Furnell [11] considers how to estimate the level of threat that is likely to originate from a particular insider based on certain profiles of user behavior. As they acknowledge, substantial work is still required to validate the proposed solutions. Myers *et al.* [12] considered how web server log data can be used to identify malicious insiders who look to exploit internal systems. Maloof and Stephens [13] proposed a detection tool for when insiders violate need-to-know restrictions that are in place within the organization. Okolica *et al.* [14] used probabilistic latent semantic indexing with users to determine employee interests, which are used to form social graphs that can highlight insiders. Liu *et al.* [15] proposed a multilevel framework, which is called sensitive information dissemination detection, that incorporates network-level application identification, content signature generation and detection, and covert communication detection.

More recently, Eldardiry *et al.* [16] have also proposed a system for insider threat detection based on feature extraction from user activities. However, they did not factor in role-based assessment. The profiling stage that we perform allows us to extract many more features beyond the activity counts that they suggested. Brdiczka *et al.* [17] combined psychological profiling with structural anomaly detection to develop an architecture for insider threat detection. They used data collected from the multiplayer online game, i.e., World of Warcraft, to predict whether a player will quit their guild. In contrast to real-world insider threat detection, they acknowledged that the game contains obvious malicious behaviors; however, they aimed to apply these techniques to real-world enterprises. Eberle *et al.* [18] considered graph-based anomaly detection as a tool for detecting insiders, based on modifications, insertions, and deletions of activities from the graph. They used the Enron e-mail data set [19] and cellphone traffic as two preliminary cases, within the intention of extending to the CERT insider threat data sets. Senator *et al.* [20] proposed to combine structural and semantic information on user behavior to develop a real-world detection system. They used a real corporate database, gathered as part of the Anomaly Detection at Multiple Scales program; however, due to confidentiality, they cannot disclose the full details, and thus, it is difficult to compare against the work. Parveen *et al.* [21] used stream mining and graph mining to detect insider activity in large volumes of streaming data, based on ensemble-based methods, unsupervised learning, and graph-based anomaly detection. Parveen and Thuraisingham [22] extended the work with an incremental learning algorithm for insider threat detection that is based on maintaining repetitive sequences of events. They used trace files collected from real users of the Unix C shell [23]; however, this public data set is relatively dated now.

One clear observation from these related work is that access to real-world data is extremely difficult, and thus, researchers synthesize data that are similar to that of a real-world enterprise, or use a subset of data points, or apply insider threat detection techniques to other problem domains (e.g., online games). In our work, we particularly wanted to represent the variety and volume of data that would be observed in a modern real-world organization and show how this could be combined to form an overall assessment for each user and for each role. We also wanted to clearly demonstrate a wide variety of insider threat scenarios as represented by our synthetic data generation and show how our detection system would be capable of detecting the different attacks.

III. Requirements Analysis

The work described in this paper was carried out as part of a wider interdisciplinary project that includes computer scientists, security researchers, and cyber psychology experts. As the problem of insider threat continues to be of growing concern to businesses and governments alike, there becomes a critical need for practical tools to help alleviate the threat that is posed. Our understanding of what we believe to constitute as insider threat is the result of close interdisciplinary collaboration between industry, government, and academia. The system that is proposed here aims to address the majority of scenarios that are understood from the knowledge that has been shared by organizations experiencing such attacks and case studies that have been documented in research reports and the media.

Our initial work on insider threat detection was to develop a conceptual model of how a detection system could connect the actions of the real world with the hypothesis that a particular individual is an insider [3]. It is crucial that organizations looking to deploy insider threat detection tools have a clear understanding of the valuable assets of the organization and the monitored activities that relate to these assets, to therefore understand the type of attacks that could potentially arise. In developing our conceptual model, we identified the different elements that exist within organizations to understand what elements could be affected as a result of an insider attack. As a result, we can define the requirements of the detection system as given in the following.

1) The system should be able to determine a score for each user that relates to the threat that they currently pose.
2) The system should be able to deal with various forms of insider threat, including sabotage, intellectual property theft, and data fraud.
3) The system should be also able to deal with unknown cases of insider threat, whereby the threat is deemed to be an anomaly for that user and for that role.
4) The system should assess the threat that an individual poses based on how this behavior deviates from both their own previous behavior and the behavior exhibited by those in a similar job role.

While we aim for a well-defined detection system that can alleviate the presence of insider threat, to promise a system that can eradicate the problem is a bold claim that we do not try to state here. By the very nature of an insider attack, a sophisticated attacker would be conscious of covering their tracks to avoid being detected. For example, they could attempt to falsify or delete the activity logs that are reported to the detection system, or they could attempt to circumvent standard monitoring practices. In theory, the very nature of modifying or deleting log files should be detected and so should raise an alert, given that this behavior should not be deemed as normal. Such attacks would therefore most likely be detected through a combination of both online and offline behaviors, such as acting suspiciously in the workplace.

IV. System Overview

The architecture of the detection system is detailed in Fig. 1. Here, the detection system connects with a database that contains all available log records that exist for the organization. Such examples may be computer-based access logs, e-mail and web records, and physical building access (e.g., swipe card logs). All records for the current date are retrieved and parsed by the system. For each record, the user ID is used to append the activity to their daily observed profile. Likewise, the activity is also appended to the daily observed profile of their associated role, if applicable. Once the daily observation profiles are constructed, the system proceeds to assess each user based on three levels of alerts: policy violations and previously recognized attacks, threshold-based anomalies, and deviation-based anomalies. At each stage in the assessment, the system can trigger an alert to the analyst to notify of a supposed threat being observed. The analyst can investigate the alert and then decide whether this alert is correct. Should the analysts decide that the alert is not correct, then they have the capability to reject a detection result, which then refines the parameters within the system, to minimize the false positive rate for future observations.

In the following sections, we will detail how each of the key components of the system is performed to identify at-risk individuals. We consider the key components of the system to be the retrieval of records from the organizational database, user and role-based profiling, profile feature extraction, anomaly assessment from features, and classification of threat from anomaly scores. For this work, a pilot detection system was developed using the Python programming language. In addition, visualization components have been also developed, which allow the analyst to explore different components of the detection process, such as user profiles and multiple anomaly scores. Our visualization components are developed using a Python back end and the popular D3 javascript library for the front-end display [24].

A. Data Input

At the first stage of the pipeline is the Data Parser Module, which interfaces with the organization. For each day, the system requests the set of records from the log data that correspond with the current date. In theory, this could consist of many different captures of data from different sensors within the organization. Our initial work was based on the data sets provided by CMU-CERT. In these data sets, the organization activity logs consist of five different files that correspond to the different activities that can be performed: login, usb device, e-mail, web, and file access. Each record is parsed to obtain a timestamp, a user ID, a device ID (i.e., what device logged the action), and an activity name (e.g., login and e-mail). Some activities (i.e., e-mails, files, and websites) may also contain further information that we assign as the attribute, such as the e-mail recipients, the filename accessed, or the website accessed. Where an attribute is provided, the system is also capable of retrieving and analyzing content that can be assigned as the final property of the record, which is handled by the Content Parser.

The Content Parser consists of two main techniques of analyzing textual data: bag of words and Linguistic Inquiry Word Count (LIWC) [25]. For analyzing website and file content, Content Parser will scrape the given URL and retrieve all texts that are recognized to exist within the English dictionary. Using a bag-of-words approach to construct a feature set, this feature vector is assigned to the given record. Similarly, for e-mail content, we construct a feature vector; however, rather than using the raw text content, we use features defined by LIWC. The justification of this is threefold. First, given the sensitivity of e-mail content, many organizations are concerned with directly monitoring the content of e-mails. Second, the LIWC categories have well-defined meaning with regard to psychological context, and thus could provide more meaningful information regarding the e-mail content than the raw message

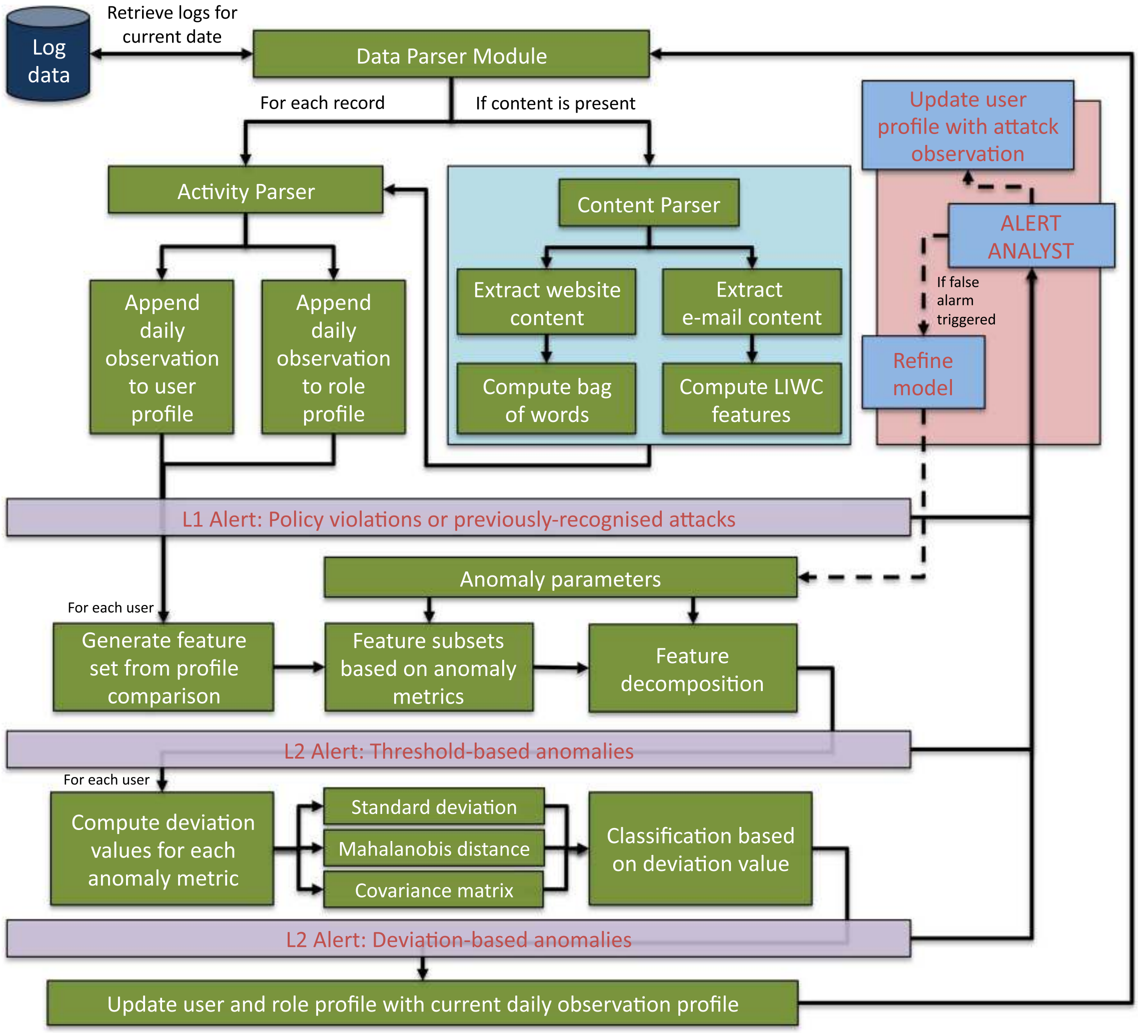

Log data
Retrieve logs for current date
Data Parser Module
For each record
If content is present
Activity Parser
Append daily observation to user profile
Append daily observation to role profile
Content Parser
Extract website content
Extract e-mail content
Compute bag of words
Compute LIWC features
Update user profile with attatck observation
ALERT ANALYST
If false alarm triggered
Refine model
L1 Alert: Policy violations or previously-recognised attacks
Anomaly parameters
For each user
Generate feature set from profile comparison
Feature subsets based on anomaly metrics
Feature decomposition
L2 Alert: Threshold-based anomalies
For each user
Compute deviation values for each anomaly metric
Standard deviation
Mahalanobis distance
Covariance matrix
Classification based on deviation value
L2 Alert: Deviation-based anomalies
Update user and role profile with current daily observation profile

Fig. 1. Architecture of the insider threat detection system. The system consists of a number of key components that process incoming log data records and construct a profile of user and role behavior for the current day and assess the level of threat posed by the invidual. Alerts can be automatically triggered at three levels: policy violations and previously recognized attacks, threshold-based anomalies, or deviation-based anomalies. Alerts are dealt with by an analyst who can then determine whether the individual does actually pose a threat or not. If deemed not to be a threat, the analyst can refine the detection model to minimize the false positive rate for future observations.

would do in any case. Finally, there are 80 features defined by the LIWC tool; thus, it means that the size of the feature vector can easily be reduced. It would be possible to use either technique for assessment of each activity; however, we make this distinction due to e-mails being user generated, rather than websites or files that are only being read by a user. Each content-based feature vector is combined with the user and role-based daily observation profiles, which we will describe further in Section IV-B.

The Content Parser serves as an optional module within our architecture. It is understood that many organizations currently do not maintain records of all content from e-mails being sent, due to privacy concerns. However, organizations may well change their position on this, particularly if it is believed that such content would help in combatting against the threat of insider attacks. For the development of our system, we have worked with a number of synthetic data sets, including CMU-CERT insider threat scenarios, the published Enron e-mail data set, sample data provided by Centre for the Protection of National Infrastructure (CPNI), and in-house generated data. One challenge with using synthetic data sets, such as CMU-CERT and our own, is that, while the data may show that e-mails were sent or files were accessed, since these are purely synthetic, there is no substantial content within the files or e-mails. E-mail content may be a collection of randomly chosen words that define a topic, rather than a meaningful communication sent by a human user. While we have been able to trial such methods on e-mail and web analysis in isolation, without these pairing up with corresponding insider threat scenarios, it is difficult to truly validate the approach. However, it is incorporated into the overall architecture since it serves as an optional complimentary anomaly metric that analysts can choose whether to utilize, based on the availability of data.

B. *User and Role-Based Profiling*

The second stage of the system is user and role-based profiling. Each user and each role that exist within the organization

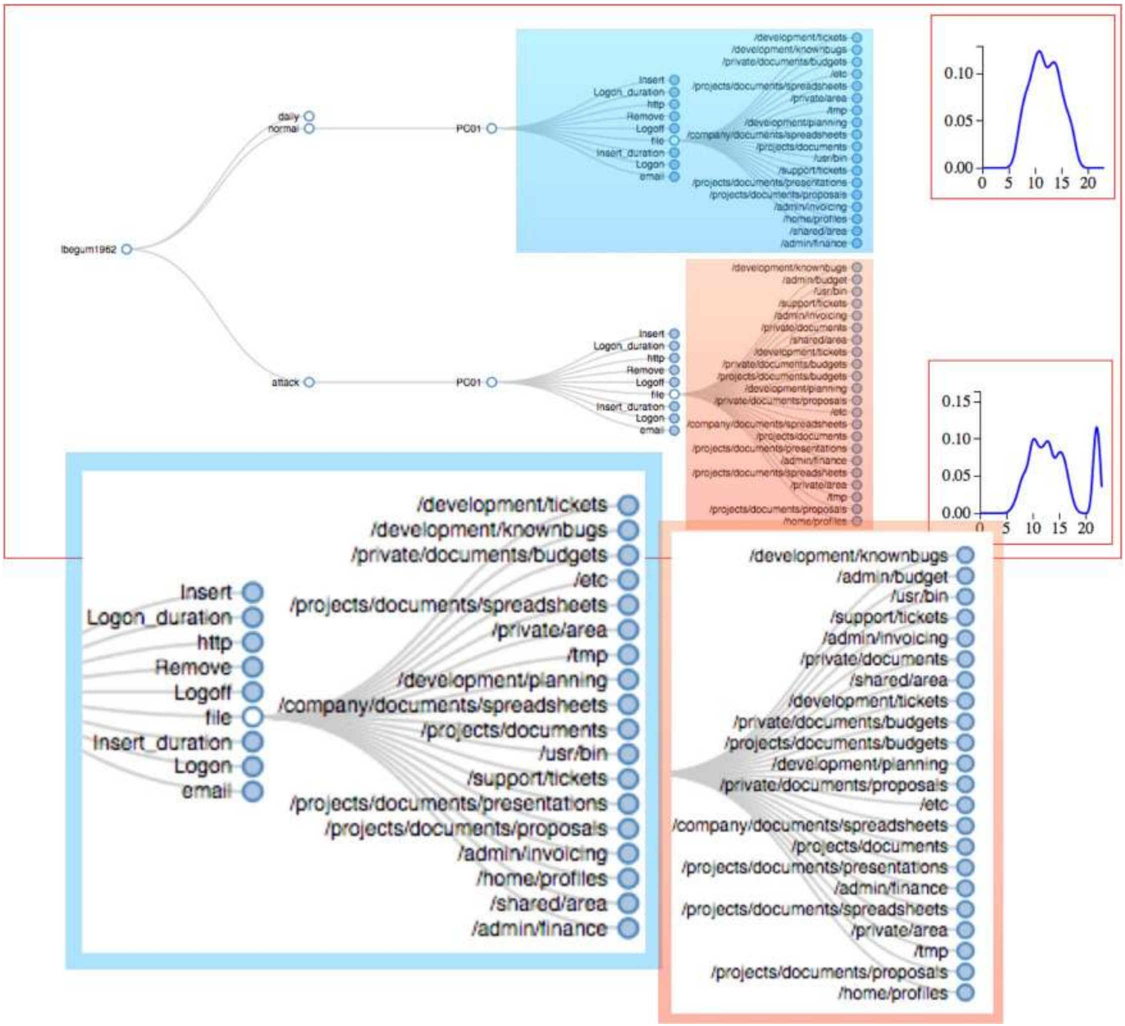


Fig. 2. Tree-structured profiles of user and role behaviors. The profile shows all the devices, activities, and attributes that the user has been observed performing. The probability distribution for normal hourly usage is given in the top right, and the distribution for the detected attack is given in the bottom right. Here, it can be seen that the user has accessed resources late at night.

are defined by a tree-structured profile that describes the different devices, activities, and attributes that they have been observed on. This notion of a tree-structured profile provides a consistent representation for all users and for all roles that can be used for comparative assessment, either between multiple employees or between multiple time steps for a particular employee. Fig. 2 illustrates the profile of a typical user using our tree visualization tool. At the root of the tree is the user ID (or in the case of a role tree, the role title), which can consist of three child nodes: observations made for the current date (daily), observations that have previously been made and exist within the normal profile (normal), and, if applicable, observations that have been deemed to be suspicious (attack). For each of these branches, we define the same hierarchical structure to facilitate comparison. At the first level down, all devices that the user has been observed on are given. In the case of a role tree, this would be all devices observed by all users who act within this particular role. These typically would be computers; however, this could well be extended to other electronic devices such as printers or door locks. The next level of the tree shows all the activities that the user has performed on each of these devices. The level below this then shows the attributes, if applicable, such as the files or websites accessed or the e-mail addresses that the user has contacted. Each node in the profile maintains a 24-bin histogram that denotes the hourly usage for that particular state, based on the observed records. In addition, attributes can also maintain the results of the Content Parser as a cumulative histogram.

For each record, the system first compares this record against the state of the user's current daily profile. If the device–activity–attribute tuple does not exist within the tree-structured profile, then a new node is created at the appropriate location within the daily profile tree. The associated histogram for the node is then updated based on the timestamp of the observed record. Similarly, the tuple is also compared against the corresponding role profile that the user belongs to. This provides a profile that describes the currently daily activity for all users and for all roles. If the user belongs to multiple roles, then the system can be configured to either populate all roles that the user belongs to or to create a specific role type that is then populated (e.g., *technician-engineer* could define the set of users who act in both roles). Once all daily records have been observed, the next stage of the system is to derive a feature set that provides a comprehensive and comparable description of each user's profile. To do this, the system compares the current daily profile against the existing previous profile.

Once the daily observation profile is constructed, the system can perform a comparison against organizational policies and previously recognized attack patterns. A rule-based approach can be specified using a policy language that can be used to state how particular observations should be treated (e.g., all logins out of hours should be flagged to the analyst with a medium severity level). If there are no violations flagged up at this stage, then the system proceeds to the next level.

C. Feature Extraction

Once we have computed the current daily profile for each user and for each role, we perform our feature extraction. Since the profile structure is well defined, it means that a wide variety of comparisons between users, roles, or time steps can be easily made. We define a series of features that consider new observations across devices, activities, and attributes, for the user compared against their previous behaviors, and for the user compared against the previous behavior of all users within the same role (e.g., *New device for user*, *New activity for device for role*, and *New activity for any device for user*). We also define a series of features that assess the hourly and daily usage counts for each particular device, activity, and attribute (e.g., *Hourly usage count for device*, *Hourly usage count for activity*, *Hourly usage count for attribute*, and *Daily usage count for activity*). Finally, we define time-based features for each particular device, activity, and attribute (e.g., *Latest logon time for user*, *Earliest USB time for user*, and *USB duration for user*). The full feature matrix that we currently consider consists of 168 columns (the full list of extracted features is available in [26]). The complete set of features allows for assessment of three key areas: the user's daily observations, comparisons between the user's daily activity and their previous activity, and comparisons between the user's daily activity and the previous activity of their role.

D. Threat Assessment

Once the feature set for the current daily observation has been computed, the next stage of the system is to determine whether these features show significant deviation in behavior compared with all previously accepted observations. To do this, an $n \times m$ matrix is constructed for each user, where n is the total number of sessions (or days) being considered, and m is the number of features that have been obtained from the profile. The bottom row of the matrix represents the current daily observation, with the remainder of the matrix being all previous observation features. To derive the amount of variation that is exhibited in the multivariate feature space, we perform PCA to obtain a projection of the features into lower dimensional space based on the amount of variance exhibited by each feature. What this means is that features that have a higher variance can be projected into a lower dimensional space while preserving separability between similar and dissimilar features. It is often used to enable visualization and understanding of large data sets using only two or three dimensions, to observe the clustering of similar data records. For our application, we also allow a weight to be associated with each feature so that features of greater importance can be emphasized, as dictated by an analyst. This way, the analyst can generate different models for analysis based on different configurations of weighted combinations. If no weights are specified, then the weight is taken to be $1/f$, where f is the total number of features. All feature columns are normalized before the PCA decomposition is performed. By default, we consider a decomposition of the features to a 2-D space. If all feature observations were identical, then all points in the new space would be clustered at the origin. However, given the deviation that is expected of human behavior, points are likely to be clustered near to, but not directly at, the center. For the new matrix, we consider only the current observation, which is the bottommost record in the matrix. We compute the distance of this point from the origin in the new space and take this to be the anomaly score of this metric at this observation. This process is performed for each of the anomaly metrics, where each metric consists of a subset of the overall feature set and, if specified, a corresponding weighting function for each feature. Each anomaly metric can be configured to alert if the score obtained for that particular metric is above a particular threshold.

The anomaly metrics that are currently considered include the following: *Login_anomaly*, *Login_duration_anomaly*, *Logoff_anomaly*, *USB_inserstion_anomaly*, *USB_duration_anomaly*, *Email_anomaly*, *Web_anomaly*, *File_anomaly*, *This_anomaly*, *Any_anomaly*, *New_anomaly*, *Hourly_anomaly*, *Number_anomaly*, *User_anomaly*, *Role_anomaly*, and *Total_anomaly*. The system could easily support the addition of further anomaly metrics, based on the observation of different activity types. From our research into case studies of insider threat, most cases could be associated with either performing a new activity, performing an existing activity at a new time of day, or performing an existing activity more or less often than previously. These define our "new," "hourly," and "number" metrics. The combination of multiple metrics also provides support for greater confidence in the result obtained regarding an individual. For example, we may observe that a particular individual scores higher than other users not only on "hourly_anomaly" or "total_anomaly" but also on "file_anomaly" and "e-mail_anomaly." By considering how the different subsets of features score, rather than a single overall score, it allows an assessment to be made on not only that an individual is posing as a threat but also on what attack vectors they are acting on. Here, we observe that the user is logging in at an unusual time to access new files and e-mail new contacts.

E. Classification of Threat

The final stage of the system is to provide assessment of the threat that is posed by an individual, given the observation of their activity, and the collection of anomaly scores that have been assigned to their daily observation profile. One approach to this as a relatively effective measure is to simply normalize each column of the anomaly score matrix and then take the maximum standard deviation as an integer classification of importance. We would expect most data to exist within two standard deviations of the norm; thus, anything above this should certainly be investigated. Likewise, we can also compute the Mahalanobis distance to assess how far away an individual's observations are from the rest of the distribution. As a third approach that can be deployed, we compute the covariance matrix of a user's anomaly scores and, on each daily observation, assess the signed differences between the covariances. The system could well be extended to support other classification schemes in the future as desired by the analyst. The classification can be used to flag up users to the analyst and to determine whether a user's daily observation profile should be included within their previously observed normal profile. If the observation is deemed to be too much of an anomaly, then the observation is recorded as an attack rather than their normal. This is a vital stage so as to not contaminate a user's previously observed profile with malicious behavior, while also providing the capability for each daily observation to contribute toward the previously observed profile.

V. Experimentation

To be able to assess the performance of the detection system, we conduct a series of experimentation scenarios using the prototype system. As part of the wider project on insider threat, ten scenarios have been developed that cover the broad range of possible attacks that an insider could perform against their organization. For each scenario, a narrative has been devised that explains what has happened, including why the individual has chosen to act against the organization, and what they have done. Each scenario is modeled within a unique synthetically generated data set that represents the normal activity of the organization. The data contain all employee activities within the organization for the period of 365 days, including that of the insider. We consider the first 15 days as training data, where no attacks are initiated, so that an initial normal baseline can be obtained. The remaining 350 days are then used as testing, whereby each newly observed day that is deemed to be normal then contributes toward the normal baseline. The scenarios were developed in isolation of the detection system, so not to have been bias by this, and have been designed to test a variety of different scenarios that could occur over different attack vectors. In addition to our own synthetic data, we have also used third-party data sets generated by CMU-CERT to further validate the performance of the approach described.

A. Constructing Experimentation Data

The creation of the synthetic data sets was conducted in isolation of the detection system so as to not introduce any bias. The premise of the activity was to craft a synthetic organization for each scenario and insert a malicious employee in such a way that their behaviors correspond with those that have been documented by the various case studies of previously observed attacks. All the while, the intent was to create different scenarios with the objective of beating the detection system, within the confines of the data points available as described in Section IV-A.

TABLE I
Characteristics of the Ten Insider Threat Scenarios, Including the Volume and Type of Inserted Malicious Activity. Each Scenario Consists of 365 Days of Activity Logs

#	# users	# records	# days modified	Malicious activities
1	100	2486663	4	(20 file, 4 email)
2	305	8452267	5	(34 file, 5 http)
3	12	115745	10	(34 file, 20 usb, 10 login)
4	50	905053	5	(10 login, 9 http)
5	200	2692373	1	(200 email)
6	100	2514792	2	(2 file, 1 login, 2 usb, 4 email)
7	305	8458402	2	(3 file, 6 email)
8	12	117195	10	(10 login, 24 file)
9	50	893700	4	(6 login, 1 file, 2 usb)
10	200	2697772	1	(2 http, 1 file)

The approach used to generate the data sets involved an automated system to generate the normal day-to-day activity (the background noise), and then, the attack data were manually injected into the log files. The method used to create the normal activity has focused on the notion of defining a "virtual organization." In our system, an organization is composed of a number of staff roles (e.g., manager and developer), with a number of employees in each of the roles.

The employee's role is used as the seed for the data generation process and determines the boundaries of normal behavior of an employee undertaking that role. An employee's role, within our virtual organization, defines the normal boundaries of behavior over a number of data dimensions, including the following: log-in times, USB device insertions, HTTP requests, e-mail contacts, e-mails sent, and file system accesses. The role does not provide entirely uniform behavior; there are only average values for an employee in that role. For example, an employee in an administrative role may typically log in to the system between 8 A.M. and 10 A.M. and log out between 4 P.M. and 6 P.M. The data generation system would, typically, assign the employee a log-in time within the specified window, but there is also the provision to generate occasional anomalous values outside of this window.

Once the normal activity has been generated, then the malicious activity is manually inserted into the data sets. For each of the attacks inserted, an attack scenario was written, specifying the type of employee (i.e., the employee's role) and describing the nature of the attack. For example, a scenario may specify that a manager, within the organization, had been arriving at work earlier than they normally would and browsing to new areas of the corporate network that they had not previously visited. Once a scenario is created, then an employee in the correct role is selected, and the attack data are inserted into the log files. Owing to the random nature of the data generation process, very little was known about the behavior of the "malicious" employee prior to the insertion of the attack data. The data inserted, about an attack, relate directly to the employee's behavior, rather than that of the role. If we consider the earlier example of a manager who begins to log in earlier than before and accesses new areas of the corporate network, then the

TABLE II
Results From Ten Insider Threat Scenarios for Level-2 and Level-3 Alerts

#	L2 alerts	L2 alerts	L2 alerts	L3 alerts	L3 alerts	L2 anomaly vectors	L3 anomaly vectors
	($\sigma > 0.1$)	($\sigma > 0.2$)	($\sigma > 0.3$)	($\sigma = 1.0$)	($\sigma = 2.0$)		
1	935	415	352	276	75	n/a	n/a
2	3033	1481	1259	964	293	n/a	n/a
3	145	92	88	63	24	logon, logon_duration	insert, file, hourly, user, total
4	373	120	83	68	14	logoff, user, role, number, new	logon, logon_duration, total
5	1573	553	474	391	82	user, this	new, email
6	906	394	340	287	52	user, new, this	n/a
7	3068	1462	1254	977	276	n/a	user, file
8	160	94	90	64	25	logon, logon_duration	user, file, total
9	358	125	89	68	20	logon, logon_duration	n/a
10	1645	610	526	452	73	n/a	new, number, user, role, hourly, total

earlier logins would be early for that particular employee, not the role as a whole. This makes the attack insertion slightly more subtle and harder to identify. Details of each synthetic data set are provided in Table I.

B. Results

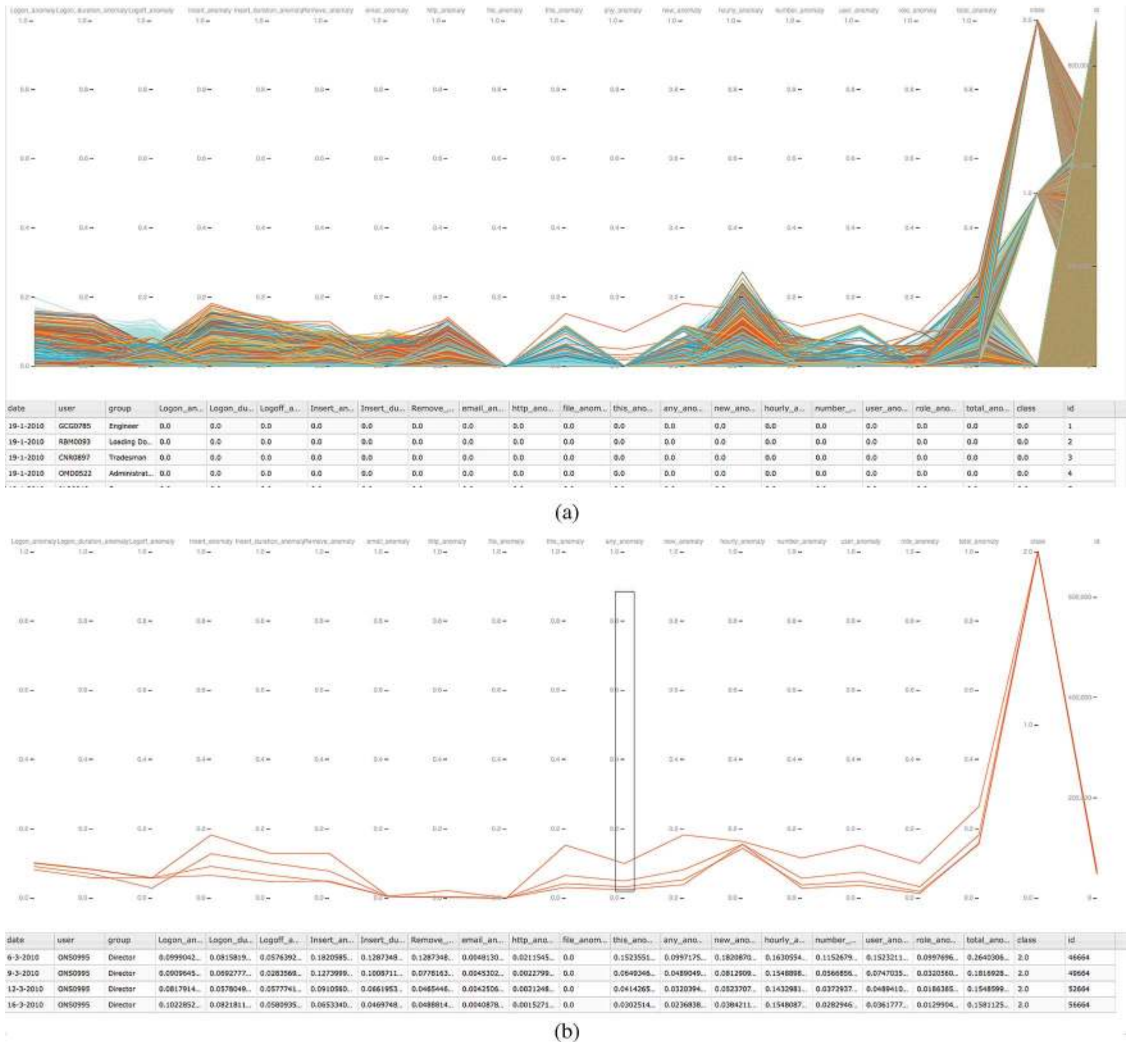
Table II shows the results from the detection system. We show the number of alerts that are generated under different operational schemes for Level-2 and Level-3 alerts. In addition, we show the anomaly metrics that the alerts were triggered for. In this experiment, it is clear that L3 alerts with a deviation of $\sigma = 2.0$ gave the fewest alerts. In real-world operation, it may well be beneficial to preserve alerts generated under different operational schemes, for instance, to observe that an employee is consistently scoring just below a particular threshold. This knowledge, coupled with offline behaviors, could well reveal the employee to be a threat, which would have been missed otherwise. From these results, the best result is obtained from scenario 3. Here, there are 4200 daily assessments made (12 employees for 350 days), of which 24 are flagged as anomalies. From Table I, we see that ten of the days consisted of malicious activity, of which all ten days are within the set of 24 detected anomalies. Based on precision and recall, this gives a precision of 42% and a recall of 100%. While it is clear that the system still presents some error, the effort of an analyst to investigate 24 results rather than 4200 is still clearly advantageous. The classification of either being an insider threat or not is somewhat of an ambiguous task, since it is highly dependent on context, and it also involves the analyst or managers to determine what the next course of action should be regarding the individual. What is perhaps most important from any insider threat detection system is that recall is ensured over precision. In this sense, the detection system serves as a means to filter a substantial number of assessments to alleviate the efforts of the analyst.

Futhermore, we also present our results using a parallel coordinates plot that shows each of the anomaly metrics as an individual axis (as shown in Fig. 3). This example is shown for a scenario generated by CMU-CERT, where the data set consists of 1000 employees, of which 1 is an insider. Fig. 3(a) shows 691 000 daily observations on the plot, and yet, there is a distinct polyline, which is seen to be an outlier on multiple axes. By brushing the axis, as shown in Fig. 3(b), the analyst can filter the data and reveal information on this particular case. Here, there are now only four observations, all of which are the actions of the inserted insider who copied data to a USB drive during unusual work hours. By coupling the detection results with a visual analytics approach, this empowers the analyst much more to be able to identify anomalous behavior within the daily observation records.

In our experimentation, we found that seven of the ten cases were clearly identifiable when using the parallel coordinates plot. Of the cases that did fail, there are a number of factors that could impact on the performance of the system. First, one of the scenarios that failed was dependent on the content of a website, rather than the unusual access of it. While the architecture of the system supports content, we did not include this in for the synthetic experimentation as the website addresses were randomly created, and thus, access to these would not be feasible. Second, despite the synthetic data being modeled to reflect human behavior, it is difficult to truly capture the intentions and motivations of the employees who are supposedly acting normally. Therefore, there is possibility that the normal background data exhibit noise and randomness that real data should not have. Having said this, it is also possible that the opposite could be true for some organizations and that, in fact, the synthetic data are too simple and not truly reflecting the dynamic nature of real human behavior. Nevertheless, we believe that the results presented here, for threshold and deviation-based assessment and for visual assessment of anomalies, are encouraging for our initial experimentation. We are currently working with a large international corporation to deploy our experimentation system within their environment to test our system against real-world activity data.

VI. Conclusion and Future Work

In this paper, we have presented an effective approach for insider threat detection. From the organizational log data, the system generates user and role-based profiles that can describe the full extent of activities that users perform within the organization. The tree-structured profiles are designed to be easily comparable against other users, role types, and temporal observations. From each daily observation, the system constructs a large set of features that describe the state of the current daily profile and the previously observed profiles for all users. The system then creates subsets of the features that describe particular anomalies of interest and computes a PCA decomposition on this to identify features that exhibit high deviation. Alerts are generated when anomaly scores are deemed to be over a particular threshold, measured as the standard deviation from the normalized anomaly scores. From an alert,



date	user	group	Logon_an…	Logon_du…	Logoff_a…	Insert_an…	Insert_du…	Remove_…	email_an…	http_ano…	file_anom…	this_ano…	any_ano…	new_ano…	hourly_a…	number_…	user_ano…	role_ano…	total_ano…	class	id
19-1-2010	GCG0785	Engineer	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1
19-1-2010	RBM0093	Loading Do…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
19-1-2010	CNR0897	Tradesman	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3
19-1-2010	OMD0522	Administrat…	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4

(a)

date	user	group	Logon_an…	Logon_du…	Logoff_a…	Insert_an…	Insert_du…	Remove_…	email_an…	http_ano…	file_anom…	this_ano…	any_ano…	new_ano…	hourly_a…	number_…	user_ano…	role_ano…	total_ano…	class	id
6-3-2010	ONS0995	Director	0.0999042…	0.0815819…	0.0576392…	0.1820585…	0.1287348…	0.1287348…	0.0048130…	0.0211545…	0.0	0.1523551…	0.0997175…	0.1820870…	0.1630554…	0.1152679…	0.1523211…	0.0997696…	0.2640306…	2.0	46664
9-3-2010	ONS0995	Director	0.0909645…	0.0692777…	0.0283568…	0.1273999…	0.1008711…	0.0778163…	0.0045302…	0.0022799…	0.0	0.0649346…	0.0489049…	0.0812909…	0.1548898…	0.0566856…	0.0747035…	0.0320560…	0.1816928…	2.0	49664
12-3-2010	ONS0995	Director	0.0817914…	0.0578049…	0.0577741…	0.0910560…	0.0661953…	0.0465446…	0.0042506…	0.0031248…	0.0	0.0414265…	0.0320394…	0.0523707…	0.1432981…	0.0372937…	0.0489410…	0.0186385…	0.1548599…	2.0	52664
16-3-2010	ONS0995	Director	0.1022852…	0.0821811…	0.0580935…	0.0653340…	0.0469748…	0.0488814…	0.0040878…	0.0015271…	0.0	0.0302514…	0.0236838…	0.0384211…	0.1548087…	0.0282946…	0.0361777…	0.0129904…	0.1581125…	2.0	56664

(b)

Fig. 3. Parallel coordinates plot of the multiple anomaly results that are generated by the detection process. (a) Plot shows 691000 results for a year and a half of monitoring 1000 employees. It can be seen that there exists a record that can be described as an outlier over at least four of the different metrics. (b) Plot shows four results that have been highlighted by the *any_anomaly* metric that all correspond to a particular individual. In this example, this invidual was the inserted malicious insider who began using a USB device to copy sensitive records earlier than they would normally act.

the analyst can visualize how the user differs from their normal behavior, or from other users, using a range of visualization techniques. We demonstrate this approach for a variety of synthetically generated insider threat scenarios, both from our own development and from CMU-CERT, and find that the system performs well for identifying these attacks across the range of anomaly metrics that are considered.

Clearly, by the very nature of an insider threat, the individual in question is purposely attempting to stay below the radar; and thus, to guarantee 100% detection success is difficult since there could be a number of attacks that are not considered by the designers of the detection system. Our future work is to explore the notion of model evolution and how multiple detection models could operate in parallel. In our current architecture, we have shown the process of refining the current model, but what if the analyst chose to maintain both models and compare the two? The analyst would then need to be able to assess the performance of each model over time, to decide whether it is worth utilizing all models or whether some models should be discarded. There are also organizational-dependant characteristics that may need to be considered; however, the approach described is designed to be flexible to the forms of data that different organizations may collect. We are currently conducting experiments with a large real-world organization to see how effective the tools can be when studying real users and, in particular, the differences between real normal and real threats. We are also exploring whether decomposition to different levels of dimensionality can improve the precision results for the detection system, to further alleviate analyst efforts. What is very clear, however, is that organizations recognize that real threats exist and that such systems as this could well detect and alleviate the efforts that are required of organizational security analysts.

Acknowledgment

This research was conducted in the context of a collaborative project on Corporate Insider Threat Detection, sponsored by the U.K. National Cyber Security Programme in conjunction with the Centre for the Protection of National Infrastructure, whose support is gratefully acknowledged. The project brings together three departments of the University of Oxford, the University of Leicester, and Cardiff University.

References

[1] D. M. Cappelli, A. P. Moore, and R. F. Trzeciak, *The CERT Guide to Insider Threats: How to Prevent, Detect, and Respond to Information Technology Crimes*, 1st ed. Reading, MA, USA: Addison-Wesley, 2012.

[2] I. Jolliffe, *Principal Component Analysis*. Hoboken, NJ, USA: Wiley, 2005.

[3] P. A. Legg *et al.*, "Towards a conceptual model and reasoning structure for insider threat detection," *J. Wireless Mobile Netw., Ubiquitous Comput., Dependable Appl.*, vol. 4, no. 4, pp. 20–37, Dec. 2013.

[4] M. Bishop *et al.*, "Insider threat detection by process analysis," in *Proc. IEEE SPW*, 2014, pp. 251–264.

[5] M. Bishop, S. Engle, S. Peisert, S. Whalen, and C. Gates, "We have met the enemy and he is us," in *Proc. NSPW*, Lake Tahoe, CA, USA, Sep. 2008, pp. 1–12.

[6] F. L. Greitzer and R. E. Hohimer, "Modeling human behavior to anticipate insider attacks," *J. Strategic Security*, vol. 4, no. 2, pp. 25–48, May 2011.

[7] J. R. C. Nurse *et al.*, "Understanding insider threat: A framework for characterising attacks," in *Proc. IEEE SPW*, 2014, pp. 214–228.

[8] F. Kammueller and C. W. Probst, "Invalidating policies using structural information," *J. Wireless Mobile Netw., Ubiquitous Comput., Dependable Appl.*, vol. 5, no. 2, pp. 59–79, Jun. 2014.

[9] M. R. Ogiela and U. Ogiela, "Linguistic protocols for secure information management and sharing," *Comput. Math. Appl.*, vol. 63, no. 2, pp. 564–572, Jan. 2012.

[10] L. Spitzner, "Honeypots: Catching the insider threat," in *Proc. 19th IEEE ACSAC*, Las Vegas, NV, USA, Dec. 2003, pp. 170–179.

[11] G. B. Magklaras and S. M. Furnell, "Insider threat prediction tool: Evaluating the probability of IT misuse," *Comput. Security*, vol. 21, no. 1, pp. 62–73, 1st Quart. 2002.

[12] J. Myers, M. R. Grimaila, and R. F. Mills, "Towards insider threat detection using web server logs," in *Proc. 5th Annu. CSIIRW—Cyber Security Inf. Intell. Challenges Strategies*, New York, NY, USA, 2009, pp. 54:1–54:4.

[13] M. A. Maloof and G. D. Stephens, "Elicit: A system for detecting insiders who violate need-to-know," in *Recent Advances in Intrusion Detection*, vol. 4637, Lecture Notes in Computer Science, C. Kruegel, R. Lippmann, and A. Clark, Eds. Berlin, Germany: Springer-Verlag, 2007, pp. 146–166.

[14] J. S. Okolica, G. L. Peterson, and R. F. Mills, "Using PLSI-U to detect insider threats by datamining e-mail," *Int. J. Security Netw.*, vol. 3, no. 2, pp. 114–121, 2008.

[15] Y. Liu *et al.*, "SIDD: A framework for detecting sensitive data exfiltration by an insider attack," in *Proc. 42nd HICSS*, Jan. 2009, pp. 1–10.

[16] H. Eldardiry *et al.*, "Multi-domain information fusion for insider threat detection," in *Proc. IEEE SPW*, May 2013, pp. 45–51.

[17] O. Brdiczka *et al.*, "Proactive insider threat detection through graph learning and psychological context," in *Proc. IEEE Symp. SPW*, San Francisco, CA, USA, May 2012, pp. 142–149.

[18] W. Eberle, J. Graves, and L. Holder, "Insider threat detection using a graph-based approach," *J. Appl. Security Res.*, vol. 6, no. 1, pp. 32–81, Dec. 2010.

[19] B. Klimt and Y. Yang, "The enron corpus: A new dataset for email classification research," in *Machine Learning: ECML 2004*, vol. 3201, Lecture Notes in Computer Science, J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, Eds. Berlin, Germany: Springer-Verlag, 2004, pp. 217–226.

[20] T. E. Senator *et al.*, "Detecting insider threats in a real corporate database of computer usage activity," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2013, pp. 1393–1401.

[21] P. Parveen, J. Evans, B. Thuraisingham, K. W. Hamlen, and L. Khan, "Insider threat detection using stream mining and graph mining," in *Proc. IEEE 3rd Int. Conf. Social Comput. PASSAT*, Oct. 2011, pp. 1102–1110.

[22] P. Parveen and B. Thuraisingham, "Unsupervised incremental sequence learning for insider threat detection," in *Proc. IEEE Int. Conf. ISI*, Jun. 2012, pp. 141–143.

[23] S. Greenberg, Using unix: Collected traces of 168 users, Univ. Calgary, Calgary, AB, Canada, Tech. Rep., 1988.

[24] M. Bostock, V. Ogievetsky, and J. Heer, "D3 data-driven documents," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011.

[25] Y. R. Tausczik and J. W. Pennebaker, "The psychological meaning of words: LIWC and computerized text analysis methods," *J. Lang. Social Psychol.*, vol. 29, no. 1, pp. 24–54, Mar. 2010.

[26] P. A. Legg, O. Buckley, M. Goldsmith, and S. Creese, "Caught in the act of an insider attack: Detection and assessment of insider threat," in *Proc. IEEE Int. Symp. HST*, Waltham, MA, USA, 2015, in press.