Detecting Unknown Insider Threat Scenarios

William T. Young, Alex Memory, Henry G. Goldberg, Ted E. Senator
Leidos, Inc.
Arlington, VA, USA
{youngwil, memoryac, goldberghg, senatort}@leidos.com

Abstract— **This paper reports results from a set of experiments that evaluate an insider threat detection prototype on its ability to detect scenarios that have not previously been seen or contemplated by the developers of the system. We show the ability to detect a large variety of insider threat scenario instances imbedded in real data with no prior knowledge of what scenarios are present or when they occur. We report results of an ensemble-based, unsupervised technique for detecting potential insider threat instances over eight months of real monitored computer usage activity augmented with independently developed, unknown but realistic, insider threat scenarios that robustly achieves results within 5% of the best individual detectors identified after the fact. We explore factors that contribute to the success of the ensemble method, such as the number and variety of unsupervised detectors and the use of prior knowledge encoded in scenario-based detectors designed for known activity patterns. We report results over the entire period of the ensemble approach and of ablation experiments that remove the scenario-based detectors.**

Keywords -- anomaly detection; insider threat; unsupervised ensembles; experimental case study

I. INTRODUCTION

Because of the adversarial nature of the insider threat domain, malicious insiders can be expected to attempt to hide their actions by employing techniques that they believe will evade detection, at least until after they have achieved their objective. As in other adversarial domains, a useful insider threat detection system must be able to detect not only instances of known, suspected, or hypothesized insider threat scenarios, but also instances of previously unseen and novel insider threat scenarios [9]. This paper reports the results of experiments to detect instances of insider threat scenarios inserted into a real database from monitored activity on users' computers seeded with independently-developed and inserted insider threat activities superposed on the activities of real users[1].

Insider threat detection is more difficult than detection in other adversarial domains such as money laundering, stock fraud, and counter-terrorism not only because of the possibility that insiders are more aware of an organization's information protection policies and procedures than outsiders might be, but also, and far more important, because malicious insider activity is typically only a small fraction of the overall activities performed by user(s) using their organization's information systems. Insiders not only have authorized access to their organization's information systems, but they also have legitimate functions to perform that require use of these information systems – and if they did not perform these required functions, they would be easily detected not because of their malicious activities but because of the absence of their legitimate activities, i.e., because they would not be doing their jobs. Insiders who gradually adopt malicious activities can create baselines of apparently normal activity that reduces their chances of being detected by anomaly detection methods. And insiders who wish to conduct long-term or repeated malicious activities must retain their positions in their organization for the duration of the time during which they will be conducting such malicious activities.

This paper extends the state-of-the-art in validated results on real data for insider threat detection reported in references [9] and [12] by presenting:

- an unsupervised ensemble-based anomaly-detection technique whose performance is close to that of the best of a large diverse set of anomaly detectors over many months of data and multiple scenario types
- the same unsupervised ensemble-based anomaly detection technique outperforms detectors having features designed specifically for individual suspected scenarios
- initial comparisons of the best performing anomaly detectors with those included in the ensembles over multiple months of data and multiple scenario types
- composition of ensemble-based anomaly detectors with and without scenario-focused detectors
- the ability to detect starting points for detection of complex insider threat scenarios involving unknown groups of actors collaborating over days or weeks.

Experiments and results included in this paper cover eight months of data from September 2012 through April 2013.

II. BACKGROUND: DATA, SCENARIOS, DETECTORS, AND EXPERIMENTAL METHODOLOGY

A. Test Data and Red Team Scenarios

Test data for experimentation consists of a database of 5,500 users. The data collection system, SureView (reference [11]), records all user behaviors for specified activities, such as logon/logoff, email, file actions, instant message, printer, process, and URL events for a calendar month. On average, there are 1000 events per user per active day. Data are made available on a monthly basis. The data

[1] The database is from a large corporation whose identity is not allowed to be disclosed publicly. All data are used with permission in a closed facility subject to all necessary privacy protections.

DOI 10.1109/SPW.2014.42

provider anonymizes all user identification (ID) and other personally identifiable information (PII) in the data set and hashes all information related to user events to a randomly-generated but internally-consistent designator.

Separately from the data collection process, an independent evaluator acting as a Red Team develops scenarios that model known documented insider threats (reference [2]), which are text-based use cases that encapsulate specific insider threat actions. The scenarios reflect the Red Team's field experience in the public and private sectors concerning various types of actual threat behaviors [2, 7]. The Red Team develops a distinct scenario description around a documented behavior type (e.g., a disgruntled employee who sells protected information). Having specified the scenario they identify real users whose behavior exhibits characteristics of the actors in the scenario. The Red Team then creates synthetic observables for these insiders in a given data month, and superposes the inserted observables on the corresponding, real observations of those users. The Red Team inserts up to five scenarios in a data month. The instantiation of a scenario in the test data is called an instance. The Red Team may modify the number of malicious actors and their online behaviors to create scenario variants and enable the production of multiple instances of a given scenario. There are 27 total instances of Red Team scenarios corresponding to the 13 distinct scenarios listed in Table 1. In the context of a given scenario, and for that scenario only, the existing user becomes the "malicious insider"; once the scenario concludes (i.e., the user day of the last inserted observable), the existing user ID is no longer a potential Red Team actor [4]

Scenario Name (No. of Instances)	Scenario Synopsis
Anomalous Encryption (2)	An insider passes proprietary information to an outsider, secretly encrypting files and emailing them from his work email to his personal email.
Bona Fides (2)	Espionage volunteer prints a bona fides package and takes it to a foreign embassy.
Circumventing SureView (2)	A user circumvents SureView monitoring to commit a crime.
Hiding Undue Affluence (1)	An employee possesses undue affluence because of ongoing espionage activity. They need to hide the existence of the money from investigators and they perform research on how to do so.
Indecent RFP (1)	An insider uses an inappropriate relationship with another insider to influence illegally vendor selection for a contract for personal financial gain.
Insider Startup (6)	Three co-conspirators collude to steal company IP. They coordinate the synchronized theft of proprietary information before leaving the company.
Layoff Logic Bomb (2)	An insider, concerned about worried layoffs, uploads a logic bomb into the IT system that will "detonate" unless he disables it.
Manning Up (1)	An insider emulates Bradley Manning and researching similar techniques while at work.
Manning Up Redux (1)	An insider emulates Bradley Manning and researches detection counter-measures and scripts code that will upload large amounts of files through a custom DNS
Masquerading 2 (2)	Subject sets up a rogue SSH server on another user's machine. They also make a copy of the local Windows password file and copy the file off over the network.
Outsourcer's Apprentice (2)	An insider in a software development role outsources tasks (and provides his credentials and remote access to the network) to an outsider.
Selling Login Credentials (1)	One insider who has system privileges sells (with the unwitting help of two other insiders) dummy accounts and temporary access to outsiders.
Survivor's Burden (3)	An insider, after experiencing a logic bomb and being passed over for promotion, steals IP that he then uploads to DropBox.

Table 1: Red Team Scenarios

The Red Team, in an effort to avoid evaluation bias, designs its scenarios without in-depth, technical knowledge of our of detection methods. Likewise, we do not review scenario specifics nor train our detectors on the test data so as to avoid over-fitting to a particular signal. Further, note that Table 1 contains scenario descriptions provided by the Red Team after the fact for purposes of evaluating the performance of our methods; the Red Team is continually adding new scenarios as the research is ongoing. Neither the Red Team nor our team claims that the set of scenarios included in Table 1 are a complete set of real insider threat scenarios.

B. Detectors

Our approach builds on the techniques described in references [9] and [12]. We employ a large number of diverse detectors of three main types: (1) indicator-based, (2) anomaly-based, and (3) scenario-based. Indicator-based detectors use statistical outlier techniques based on single features. Anomaly detectors employ a set of complex algorithms which focus on different aspects of the data – e.g., structural features, semantic features, temporal features – and search through the feature space to identify potential anomalies. Features typically consist of observed actions, aggregates, and ratios, such as URLs accessed by a user, the number of print jobs by a user, the ratio of the number of print jobs by a user to his/her average over some time period, or the ratio of the number of files copied to removable media compared to the number of files copied to the hard drive. Graphical features such as the email and text-message communication graphs are also employed. Scenario-based detectors are inspired by the scenarios described in reference [1], but are developed independently of the Red Team scenario descriptions and inserts.

A particular detector specification incorporates, in addition to the algorithm, a set of features, a baseline population for comparison (i.e., a peer group), a time period for the baseline activity, a time granularity for potential detection, and other relevant aspects. Baselines for comparison may be cross-sectional (i.e., compare a user's actions over a particular time period with that of other users in a "peer group" over some time comparable time period) or temporal (i.e., compare a user with his/her own behavior over different time periods), or both.

Indicator and scenario-based detectors are described in references [9] and [12]; the remainder of this section provides more detail about additional scenario-based detectors that are not described in these references. Scenario-based detectors consist of a combination of indicator-based and anomaly-based detectors and classifiers in a specified workflow, structured to reflect a hypothesized sequence of real world actions that are likely to discriminate between the scenario of interest and other, mostly legitimate, actions. In addition to the textual description of these scenario-based detectors, we include their specification using the Anomaly Detection Language [ADL] that was introduced in these references. The ADL provides a construct by which an analyst may understand better what constitutes normal insider behavior for a given entity extent, peer group (i.e., baseline population), or temporal extent. Table 2 describes the scenario-based detectors used in this paper and maps each to RT scenarios to which we believe they best correspond. Note that there are no Red Team scenarios corresponding to some of our detectors; this is because we do not know what scenarios the Red Team will choose to model and insert, so we constructed and applied a broad set of detectors to cover as many envisioned scenarios as possible, while recognizing that this set will not cover all possible IT scenarios.

Scenario-Based Detector	Description	Corresponding RT Scenario(s)
Saboteur	An insider uses corporate information technology (IT) resources to harm an organization or an individual. Saboteurs are technical, such as a system administrator, and have privileged access to systems. The saboteur plans his attack before leaving the organization and executes the attack as he leaves.	Circumventing SureView Layoff Logic Bomb Survivor's Burden
IP Thief	An insider uses corporate IT resources to steal IP. IP thieves are generally scientists, engineers, or salespeople, they generally steal what they consider to be their own work for their own private gain.	Anomalous Encryption Bona Fides Manning Up Manning Up (Redux)
Fraudster	An insider uses IT for destroying, denying, or degrading an organization's information or systems for personal gain or to commit a crime. Fraudsters are lower-level employees, are often motivated by financial need - hardship, greed, etc. Sometimes they are recruited by outsiders in collusion with other insiders.	Hiding undue Affluence Indecent RFP Masquerading (2) Outsourcer's Apprentice
Ambitious Leader	The Ambitious Leader is an IP thief who is motivated by ambition to steal as much as possible before leaving the organization. To do so, he recruits other insiders to get access to all parts of the IP being stolen.	Insider Startup Selling Login Credentials
Careless User	The insider is not intentionally malicious but, through blatant disregard of corporate IT policies, exposes the group to a comparable level of risk similar to the Saboteur scenario.	None
Rager	The insider has outbursts of strong, vociferous, abusive, and threatening language in Emai/Webmail/IM toward other insiders or against the organization in general. These outbursts coincide with anomalies in other data types, e.g., Logons, URL, indicating a potential fundamental change in behavior.	None

Table 2: Scenario-Based Detectors, Synopses, and Corresponding RT Scenarios

Scenario-based detectors not described in references [9] and [12] include Fraudster and Saboteur. Distinguishing characteristics of the Fraudster scenario is that of insider in

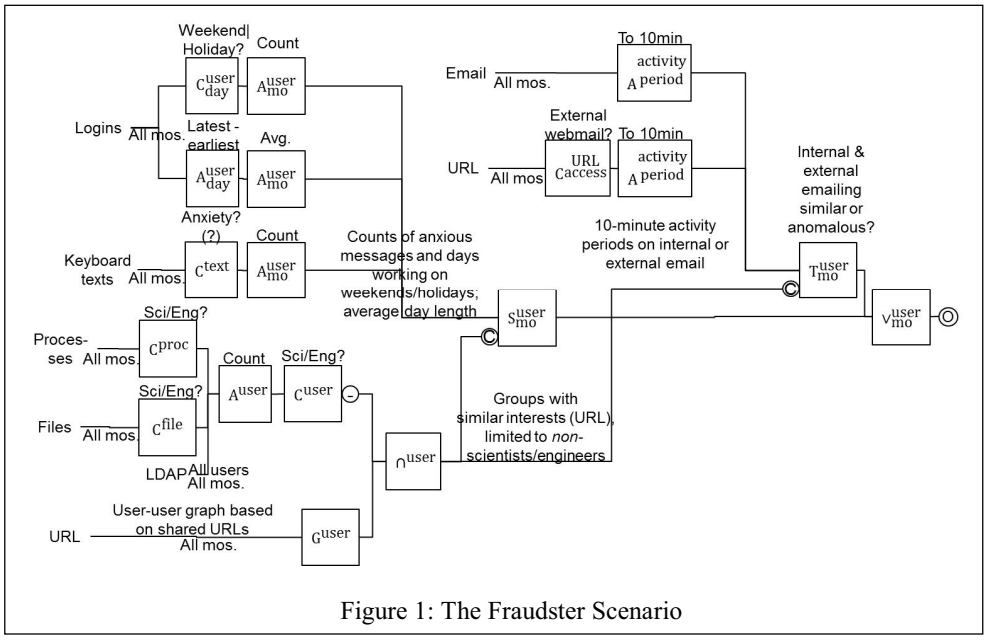

Figure 1: The Fraudster Scenario

(1) a non-technical role, (2) who is under a great deal of personal stress and commits nefarious actions but (4) wishes to minimize detectable signal and (4) does not intend to leave the organization immediately. These characteristics are important to the Fraudster because continued access to protected information, systems, or privileges are critical to the Fraudster's ultimate goals; having his activities discovered or abruptly leave the organization would cease his fraudulent activity and source of secondary income. The initial scenario definition depicts the Fraudster as someone in a non-technical role, such a program management or support staff, but the scenario could be re-configured to identify individuals in more technical roles (i.e., scientists and engineers). Figure 1 below expresses the Fraudster scenario in the ADL.

The Fraudster scenario uses login, instant message/text from the corporate instant message service, processes run by the user, file events, and URL events as the primary inputs. The detectors use process, file, and URL features to classify the user as someone in either a technical or non-technical role, comparing the user to his LDAP group (for files and processes) and the user's browsing patterns to his group by shared URL. The detector also establishes the baseline for the user's general work computer activity using login/logout events and looks for increases in anxiousness in the user's work day patterns (e.g., repeated early logins/late logouts in the same day) or messages. It then looks for anomalies in the user's email and browsing patterns (to include webmail), unusual behavior with respect to file, and process events with respect to the user's LDAP group and group with shared URLs.

The Saboteur scenario looks for a (1) user who performs a technical role and (2) displays either open hostility toward the organization or whose internal communication graph is increasingly and drastically shrinking (3) who abruptly destroys, disrupts, or denies access to protected information or systems.

The Saboteur scenario looks for personnel who perform technical actions (e.g., a system administrator or someone who acts like a system administrator) whose behavior shows unusualness with respect to access to high-risk files (e.g., proprietary information, corporate human resources records), processes (e.g., attempting to disable the monitoring agent, altering log files), and whose communication graph fits with that of a system administrator. Figure 2 depicts the features, extents, and algorithm types included in this scenario.

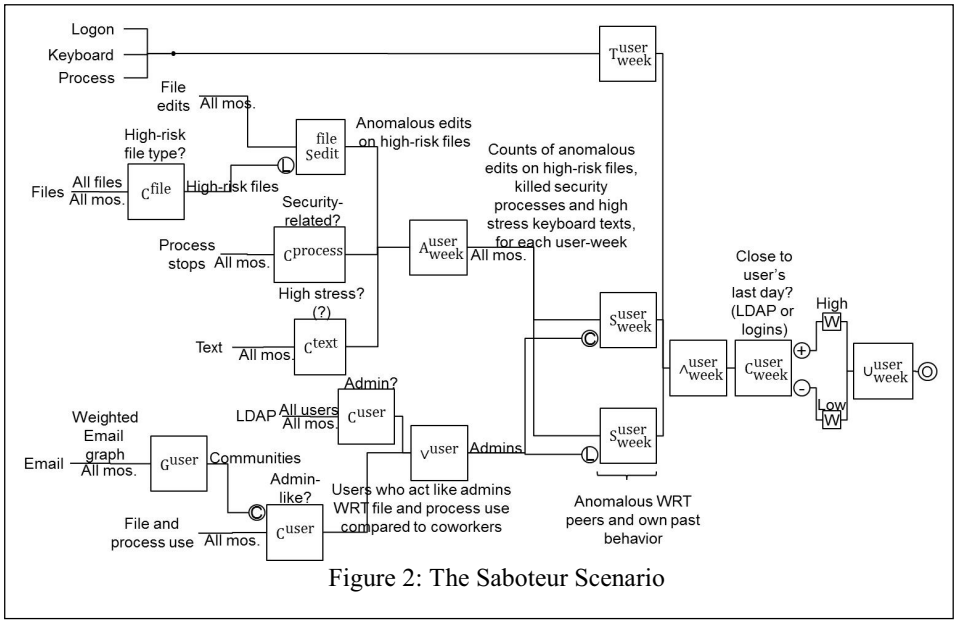

Figure 2: The Saboteur Scenario

C. Experimental Methodology

Section A describes the data environment in which our prototype operates. Instances of Red Team scenarios are limited to one month duration and inserted as targets each calendar month. (CERT has found that 2/3 of known insider threat scenarios evolve over less than one month.) This allows for consistent, independent experiments. We have leveraged the variety of inserted scenarios within this uniform data environment to demonstrate robustness of our detection methods. From month to month, within a uniform data environment, very different inserted scenarios are detected with consistent accuracy.

The experiments reported here measure detection performance on user-day entity extents – a data structure derived from the collection of activities of one user over one day. (We limit entity extents to this size for these experiments, although our prototype is capable of representing many others.) Each month, several instances of a few scenarios are inserted, each instance comprising several user-days containing inserted activities as well as actual activities by the selected users. We consider a hit to be the ranking of such a user-day above some threshold, and can measure the hit-false alarm tradeoffs using well known measures of sensitivity or recall and false positive rate or fall-out.

Metrics were chosen to measure both detection accuracy of the individual algorithms and their contribution to the overall task of providing leads to an analyst. For the former, we compute the Receiver Operator Characteristic (ROC) curve and area under the curve (AUC) as well as the Approximate Lift Curve and Average Lift. ROCs are generally used to show detection/classification tradeoffs over a set of uniform, independent trials. We reason that, while users and their activities may not fit this condition, our transformation to "entity extents" does. AUC estimates discrimination, or the probability that a randomly chosen positive entity extent will be ranked higher than a randomly chosen negative one. Cost estimates of a system utilizing these decisions can be readily made. Lift metrics, such as Average Lift estimate the improvement in target density delivered to later stages of a multi-stage detection process. In addition, we compute the number of positive hits ranked in the top k scored entity extents (for k = 5, 10, 50, 100, 500, etc.) and in the top p% of all scored entity extents (for p =

Month	Ensemble AUC	Best Detector	Best Detector AUC	Ratio (Ensemble/ Best Detector)	RT Scenario(s)
Sep-12	0.8821	RPAD up feature normalization	0.9703	90.94%	Insider Startup (3x) Circumventing Sureview
Oct-12	0.8878	RPAD up feature normalization	0.9804	90.59%	Insider Startup (2x) Anomalous Encryption Layoff Logic Bomb
Nov-12	0.7212	RDE alpha version; raw feature set; 10k training	0.7469	96.56%	Anomalous Encryption Masquerading 2 (2x) Layoff Logic Bomb
Dec-12	0.8396	GMM Density Estimation via unusualness of counts, vs company	0.8676	96.77%	Anomalous Encryption Layoff Logic Bomb Outsourcer's Apprentice
Jan-13	0.9014	RIDE using Raw Counts	0.9014	100.00%	Outsourcer's Apprentice Survivor's Burden Hiding Undue Affluence
Feb-13	0.7637	Ensemble GMM via unusualness of counts, vs company	0.7792	98.01%	Survivor's Burden Bona Fides Manning Up
Mar-13	0.8734	Ensemble GMM Density Estimation via unusualness of counts, vs company	0.8903	98.10%	Manning Up Redux Hiding Undue Affluence Survivor's Burden
Apr-13	0.8862	RIDE using Raw Counts	0.8862	100.00%	Survivor's Burden Circumventing Sureview Selling Login Credentials Indecent RFP

Table 3: Ensemble and Best Detector Results by Month

.01, .05, .1, .5, 1, 5, etc.). The latter allow us to estimate anticipated detection success for fixed analyst workloads.

III. Unsupervised Ensemble-Based Anomaly Detection

A. Methods

Having multiple anomaly detection results for the same data naturally leads to the need to combine those results to take advantage of the distinct perspectives embodied in different detectors. Further, an analyst responsible for insider threat detection desires a single ranked list rather than many different result sets from different detectors whose detailed operation he/she may not fully understand. Anomaly detector ensembles combine the results (i.e., scores) from multiple detectors in a way that is analogous to how classifier ensembles combine predictions from multiple classifiers [3].

Because when building ensembles we assume that we do not have access to ground truth, it is not known whether one of the individual detectors always performs well; in fact, in our experiments where we have ground truth we found that the best detector varied across data sets. Therefore, one goal of ensemble building is to perform as well as the best detector. It is also possible for an ensemble to outperform all of the individual detectors, which is analogous to how some classifier ensembles are able to outperform their individual classifiers.

Selecting an approach for building ensembles depends upon the types of detectors that are used. If all the detectors share an underlying model, then the ensemble approach can leverage that commonality to improve performance, e.g., the method reported in [6] varies the features used as input to a single anomaly detection model to build an ensemble. Another way of leveraging a common model is to use the same input features, but alter hyperparameters, which determine how the model is built in each detector [6, 3].

If, however, the detectors do not share a common model then the ensemble-building approach may only assume that the scores from the detectors are given as input, i.e., the features used as input to detectors and the hyperparameters of the detectors are unknown. Because our individual detectors employ a variety of models, we chose an approach that is consistent with this setting [8]. The following paragraphs describe the approach.

Some approaches for ensemble building, including the method we used, employ the following two high-level heuristics. First, if a consensus about which points are most anomalous can be drawn from the individual detectors, then that consensus should be preserved in the final ensemble. Second, because each individual detector is subject to unavoidable biases stemming from the choice of model, choice of input features, hyperparameter settings, etc., the ensemble should prefer combinations of results from detectors with uncorrelated biases.

These heuristics are implemented in two distinct phases in this method. In the first phase they extract a consensus across all detectors from the union of the top k most-anomalous points from each detector. All points in this union are given a score value of 1 and all others are given a score of 0. We chose a value for k for each dataset that included the top 1% of the points. The method then initializes the ensemble with scores from the detector that is most correlated with the consensus. The correlation between detectors and the consensus is found by viewing each as n-length vectors of scores, where there are n points in the dataset, and then using a simple correlation metric to compare the vectors. We used the Pearson's r correlation metric for this.

Month	Ensemble	Red Team Scenarios (no. of instances)	Corresponding Scenario Detector	Scenario Detector AUC	Ratio (Ensemble/ Scenario Detector)
Sep-12	0.8821	Insider Startup (4x)	Ambitious Leader	0.8388	105%
		Circumventing SureView (1x)	Saboteur	0.7117	124%
Oct-12	0.8878	Insider Startup (2)	Ambitious Leader	0.7890	112%
Mar-13	0.8734	Manning Up Redux (1x) Bona Fides (1x)	IP Thief	0.5970	104%
		Hiding Undue Affluence (1x)	Fraudster	0.6231	140%
Apr-13	0.8862	Survivor's Burden (1x) Circumventing SureView (1x)	Saboteur	0.5766	142%
		Selling Login Credentials (1x)	Ambitious Leader	0.3905	226%
		Indecent RFP (1x)	Fraudster	0.6218	142%

Table 4: Comparison of Scenario-Based Detectors Performance to Ensemble Performance, by Red Team Scenario

In the second phase, the method greedily selects candidate detectors to combine with the initial ensemble by preferring the detectors that are least correlated with the current ensemble. The same correlation metric used before is used again here. The candidate detector's scores are combined with the current ensemble using a point-wise combination function. For this the method uses the average over scores for each point; we also experimented with other functions including the maximum of scores. The algorithm proceeds to accept a candidate detector if the resulting ensemble is no less correlated with the consensus than the previous ensemble; if it is, then the candidate detector is discarded. This phase continues until all detectors are either accepted or discarded.

B. Results

Our initial experiments evaluated our detection results based on our ability to detect user-days with red-team inserted activity. We used a wide variety of detectors, described in reference [9], and the ensemble technique described above. Results are summarized in Table 3. For each month, we report the area under the ROC curve for our best detectors, for the ensemble, and for the detectors that correspond most closely to the inserted scenarios. The AUCs reflect the ability of the detectors to find all of the user-days of the union of all scenarios present in the month. These results illustrate how the unsupervised ensemble-based anomaly-detection technique had performance that is close to that of the best of the individual anomaly detectors. The AUC for the ensemble technique was consistently above 90% of the AUC of the best detector and frequently approached 100%. Interestingly, the ensemble-based technique appeared to have results that were similar across datasets, while the performance of the best detector varied widely on the same datasets. Figures 3 through 10 illustrate the differences in performance between the ensemble and the best-performing detector for each month.

Additionally, we see that the ensemble generally out-performed the scenario-focused detectors, including the scenario-focused detectors that we determined later to have been a likely fit to the red team scenario that was actually inserted. In table 4 we see that the ensemble consistently outperforms the most relevant scenario-focused detector; in two of the months, the ensemble exceeds the scenario-focused detectors by 40%. The table reports results on months where the comparison was possible, not to select only months where the ensemble performed well. Figures 11 through 14 depict the performance of the ensemble method compared to the corresponding scenario-focused detectors for these months.

Table 5 identifies the sets of detectors selected by the ensemble each month and compares them to the best performing detectors for that month. The best-performing detector was included in the ensemble in only one of the eight months of data. And because in that month – Nov-2012 – there are six detectors accepted in the ensemble, and all ensembles comprise equally-weighted detectors, the best detector is never given more than one sixth of the weight in this ensemble result. Therefore, the ensemble technique is able to achieve comparable performance to the best detector by combining detectors and with those detectors generally excluding the best-performing detector. Recall that the heuristics followed by the ensemble technique favor detectors that are either most close to the consensus or those that are able to add diversity to the ensemble (least correlation with the ensemble) without reducing correlation with the consensus. Thus in these data sets the best-performing detector generally disagrees with the consensus from other detectors, yet a combination of those other detectors can be built automatically that performs nearly as well as that best detector.

In five out of eight months at least one of the accepted detectors used an underlying model that was shared with the best-performing detector. For example, in Dec-12 the best performing algorithm is GMM Density Estimation via unusualness of counts vs company, which shares the same underlying model – Gaussian mixture models – as one of the accepted detectors, GMM Density Estimation using Raw Counts; the difference between the detectors is the input features used by the two detectors.

Month	Ensemble Selection	Best detector
Sep-12	RDE alpha version; raw feature set; 10k training RDE alpha version; up feature set; 10k training GFADD fileCreationCnt distFilesCnt 0 GFADD fixedEventCnt netEventCnt 0 GFADD remEventCnt netEventCnt 8 GMM Density Estimation using Raw Counts RIDE using Raw Counts RIDE via unusualness of counts vs. company	RPAD up feature normalization

Oct-12	RDE alpha version; raw feature set; 10k training RDE alpha version; up feature set; 10k training GFADD fileCreationCnt distFilesCnt 0 GFADD copiesToRem copiesFromRem 8 GFADD fixedEventCnt netEventCnt 0 GFADD fixedEventCnt netEventCnt 8 GFADD remEventCnt netEventCnt 8 RIDE using Raw Counts RIDE via unusualness of counts vs. company	RPAD up feature normalization
Nov-12	RDE alpha version; raw feature set; 10k training RDE alpha version; up feature set; 10k training GFADD fileCreationCnt distFilesCnt 0 GFADD copiesToRem copiesFromRem 8 RIDE using Raw Counts RIDE via unusualness of counts vs. company	RDE alpha version; raw feature set; 10k training
Dec-12	RDE alpha version; raw feature set; 10k training RDE alpha version; up feature set; 10k training GFADD fileEvents remEventCnt 8 GFADD remEventCnt netEventCnt 8 RIDE using Raw Counts RIDE via unusualness of counts vs. company	GMM Density Estimation via unusualness of counts, vs company
Jan-13	RDE alpha version; raw feature set; 10k training RDE alpha version; up feature set; 10k training GFADD fileCreationCnt distFilesCnt 0 GFADD copiesToRem copiesFromRem 8 GFADD fixedEventCnt netEventCnt 0 GFADD fixedEventCnt netEventCnt 8 RIDE using Raw Counts RIDE via unusualness of counts vs. company	RIDE using Raw Counts
Feb-13	RDE alpha version; raw feature set; 10k training RDE alpha version; up feature set; 10k training GFADD fileCreationCnt distFilesCnt 0 GFADD copiesToRem copiesFromRem 8 GFADD fixedEventCnt netEventCnt 0 GFADD fixedEventCnt netEventCnt 8 GFADD fixedEventCnt remEventCnt 8 RIDE using Raw Counts RIDE via unusualness of counts vs. company	Ensemble GMM via unusualness of counts, vs company
Mar-13	RDE alpha version; raw feature set; 10k training RDE alpha version; up feature set; 10k training GFADD fixedEventCnt netEventCnt 8 GFADD remEventCnt distRemDrivesCnt 8 GFADD remEventCnt netEventCnt 8 RIDE using Raw Counts RIDE via unusualness of counts vs. company	Ensemble GMM Density Estimation via unusualness of counts, vs company
Apr-13	RDE alpha version; raw feature set; 10k training RDE alpha version; up feature set; 10k training GFADD copiesToRem copiesFromRem 8 GFADD fixedEventCnt netEventCnt 0 GFADD remEventCnt distRemDrivesCnt 8 GFADD remEventCnt netEventCnt 8 RIDE using Raw Counts RIDE via unusualness of counts vs. company	RIDE using Raw Counts

Table 5: Ensemble Composition and Best-Performing Detector, by Month

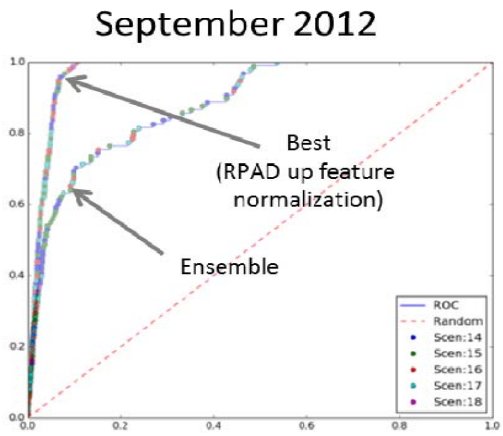

Figure 3: ROC curves for the Ensemble and the Best Detector for September 2012

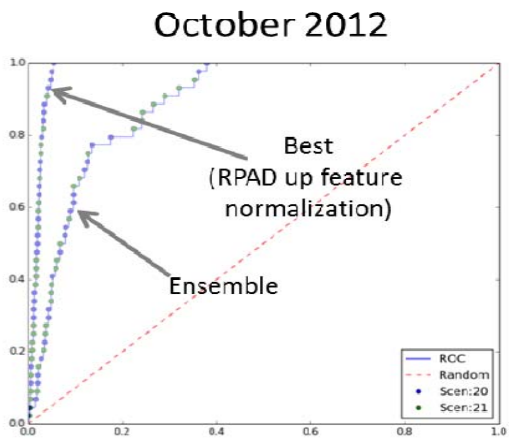

Figure 4: ROC curves for the Ensemble and the Best Detector for October 2012

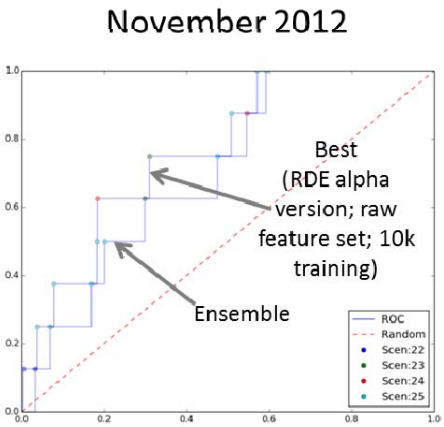

Figure 5: ROC curves for the Ensemble and the Best Detector for November 2012

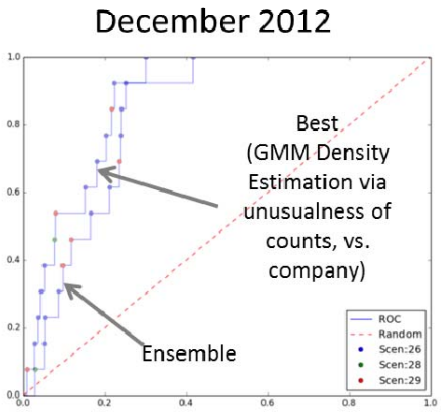

Figure 6: ROC curves for the Ensemble and the Best Detector for December 2012

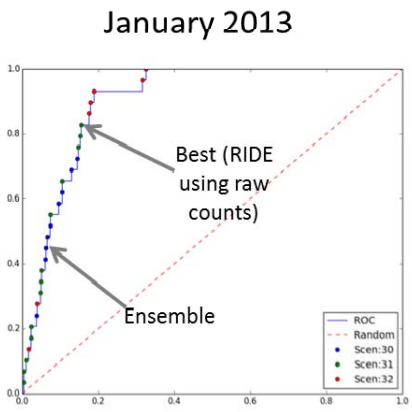

Figure 7: ROC curves for the Ensemble and the Best Detector for January 2013

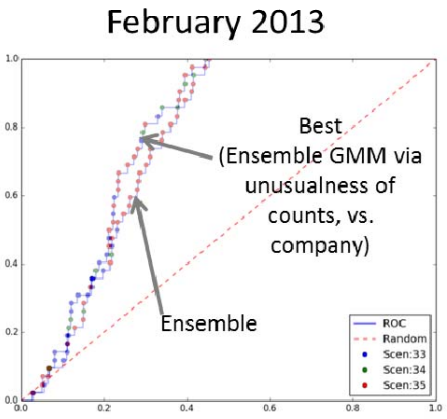

Figure 8: ROC curves for the Ensemble and the Best Detector for February 2013

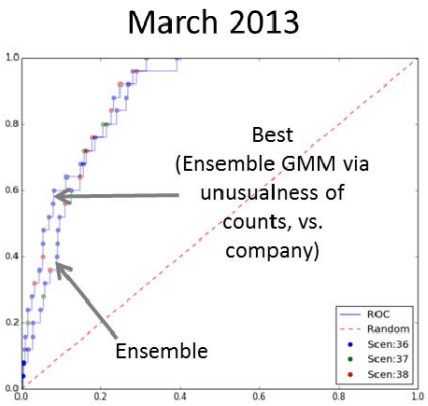

Figure 9: ROC curves for the Ensemble and the Best Detector for March 2013

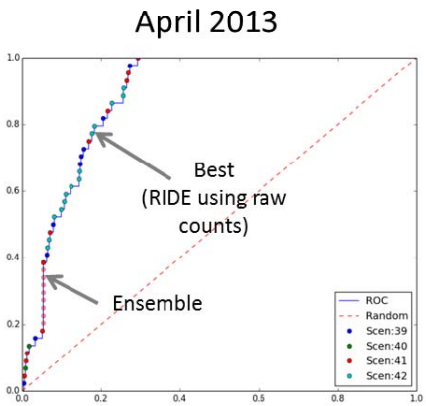

Figure 10: ROC curves for the Ensemble and the Best Detector for April 2013

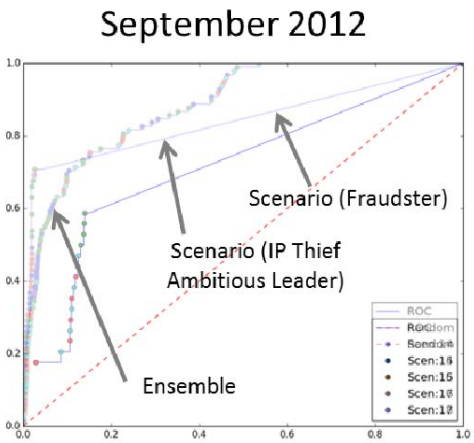

Figure 11: ROC curves for the Ensemble and the Scenario-Based Detectors for September 2012

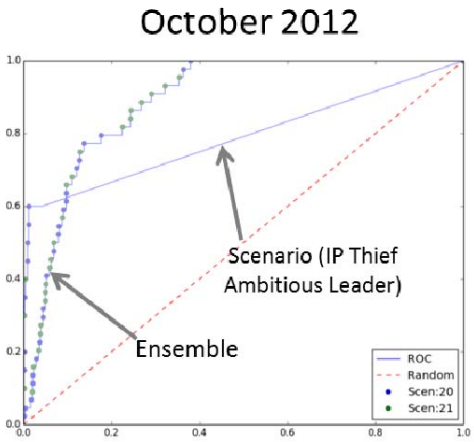

Figure 12: ROC curves for the Ensemble and the Scenario-Based Detector for October 2012

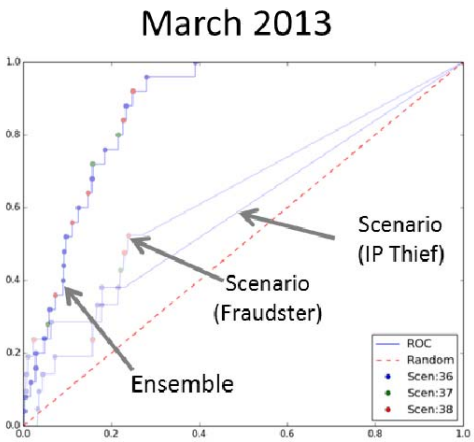

Figure 13: ROC curves for the Ensemble and the Scenario-Based Detectors for March 2013

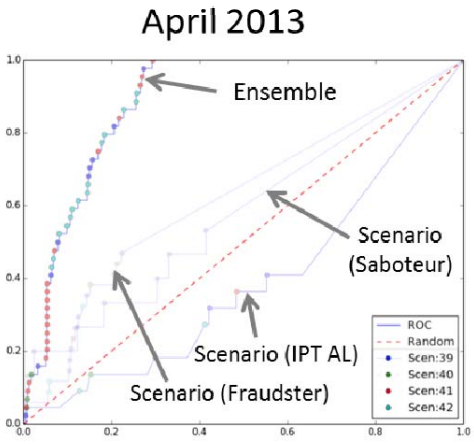

Figure 14: ROC curves for the Ensemble and the Scenario-Based Detector for April 2013

IV. COMPLEX SCENARIO DETECTION

Insider threat scenarios that form the subject of our work typically involve complex inter-related sets small numbers of actions by multiple individuals with diverse types of relationships occurring over multiple days to weeks. These actions are superposed on their "normal" activities. (These threat scenarios are contrasted with what is often described as "cyber threats" in so far as the latter tend to be executed by software at time scales of seconds or below. Network-based monitoring is typically used to detect such cyber threats; by contrast, we rely on host-based monitoring of user actions.) The essential challenge to detect these actions is that the effective signal-to-noise ratio is dependent not only on the fraction of actions by the malicious individuals that are improper but also on the combinatorics of the grouping of individuals. The Insider Startup scenario and the Indecent RFP scenario inserted by the red team and described earlier in this paper illustrate the new detection capabilities. The insider startup scenario for the month of October 2012 involved two separate instances each of three distinct users who engaged in malicious activity on 22 user-days on 12 distinct calendar days in a calendar month. The first of these instances comprised 527 distinct observations of user activity, of which 151 involved file accesses, 19 involved text messages, 42 involved processes, and 12 involved URL accesses. The second of these instances involved 468 inserted observations, of which one was an email event, 148 were file events, 41 process events, 18 keyboard events, and 278 URL events. These 527/468 inserted events correspond to roughly 0.5-0.5-0.6% of each user's total activity over the course of a month, or about 0.0003% of the total observations in the database of about 5500 users each month. The scenario was designed with minimal communications between the three malicious users to make it realistically difficult to detect; the key challenge was to aggregate behavior to detect any of the conspirators. It was the surge in file events distributed roughly equally between the three users over five consecutive calendar days was detectable. The highest-ranked user-day from our best detector for the two instances of this scenario; i.e. the number of user-days that would have to be reviewed by an analyst to detect this malicious behavior, was 17 and 13 out of the approximately 150,000 user-days analyzed that month.

For a single instance of the Indecent RFP scenario inserted during the month of April 2013, there were two users whose malicious activities occurred on 15 distinct calendar days for 14 distinct user-days, with a total of 228 inserted observations comprising 52 email events, 133 process events, and 53 text messages. These activities consisted of less than 0.5% of the malicious users' activities over the course of the month, or less than 0.0002% of the total monthly observations in the database. For this scenario, all red team inserted activity occurred in the communications between the two users. The highest-ranked inserted user-day for this month was the highest-ranked user-day overall by our best detector.

V. ABLATION EXPERIMENTS

Although the scenario-focused detectors are not accepted into the ensembles on the data sets we tested, it is still possible that the consensus formed in the first phase of the ensemble method is affected by the presence of the scenario-focused detectors. We ran a separate set of experiments with the scenario-focused detectors disabled to test whether this would affect the resulting ensembles. We did this for all months of data and varied which detectors were disabled in each month based on which related red team scenarios had been inserted in the month. In these experiments we found that removing scenario-based detectors does affect the consensus as well as affecting which other detectors are accepted into the ensembles. For example, in data sets for several months, disabling scenario-focused detectors caused many of the GFADD algorithms to flip from being discarded by the ensemble to becoming accepted by ensemble, or vice versa. Interestingly, the GFADD algorithms, like the scenario-focused detectors, are focused on a narrow set of features compared to the other individual detectors. The GFADD algorithms also tend to assign zero scores to many data points as a result of the narrow range of features. This may begin to explain why these algorithms are sensitive to the presence of points added to the consensus by the scenario-focused detectors than the other detectors, and so be more likely to have their status in the ensemble affected. Removing scenario-focused detectors did not, however, substantially affect the AUCs of the resulting ensembles.

VI. CONCLUSIONS AND FUTURE WORK

Real insider threats are complex and adversarial, which leads us to conclude that an effective system for detecting these threats must detect scenarios that builders of the system never planned for or contemplated. Therefore, it is important to evaluate systems on their ability to detect previously-unknown scenarios in real data. In this paper we evaluate our prototype in this setting and show that by using a variety of diverse individual detectors combined using an anomaly detection ensemble technique, we achieve a final detection result with performance that consistently approaches that of the unidentified detector among the set tested that was found to perform best on each dataset in after-the-fact analysis. This result holds on many data sets, including ones containing scenarios we had not contemplated when designing the detectors. The ensemble result also outperforms many anomaly detectors that are specifically focused on the scenarios that are known, on data sets containing those scenarios.

Furthermore, we investigate the composition of the ensembles chosen by the technique we use and find that the ensemble achieves consistent performance without relying any single detector or the best unidentified detector for each dataset. We also disable scenario-focused detectors in the prototype and find that the ensemble continues to perform well on a variety of scenarios. These observations suggest

that our approach is robust to gaps between the scenarios contemplated during detector design time and unexpected scenarios that appear in real data, so long as the available detectors are still diverse and numerous as we have in our prototype.

This result is one that we will continue to study in future work. Specifically, we are interested in developing more advanced ensemble techniques than the one we used that are able to incorporate scenario-focused detectors effectively to increase confidence in results when known scenarios do match with ones in the data. We will also begin incorporating explanation capabilities with the ensemble approach so that underlying reasons for detection from individual detectors can be combined in the final result presented to analysts.

VII. ACKNOWLEDGEMENTS

The work reported herein was performed as part of the Anomaly Detection at Multiple Scales (ADAMS) program sponsored by the Defense Advanced Research Projects Agency (DARPA). We thank our collaborators at Oregon State University, University of Massachusetts, Georgia Institute of Technology, and Carnegie Mellon University, and Leidos who developed many of the algorithms that were used for detection and who developed and operate the software that configures and executes these algorithms in the ADAMS testbed. We also thank the ADAMS Red Team from CERT who develops the scenarios, inserts them into the real data, and provides the answer keys against which we are evaluated. Finally, we thank the data provider and testbed operator for the ADAMS program for making the real data available for research and for operating the testbed in which it occurs.Funding was provided by the U.S. Army Research Office (ARO) and DARPA under Contract Number W911NF-11-C-0088. The content of the information in this document does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

VIII. REFERENCES

[1] C. Agarwal. “Outlier Ensembles.” In Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description, 6–6. ODD ’13. New York, NY, USA: ACM, 2013. doi:10.1145/2500853.2500855.

[2] D. Cappelli, A. Moore, R. Trzeciak, The CERT Guide to Insider Threats: How to Detect, Prevent, and Respond to Information Technology Crimes. Addison-Wesley Professional. 2012.

[3] T. Dietterich. “Ensemble Methods in Machine Learning.” In Multiple Classifier Systems, 1–15. Springer, 2000.

[4] J. Glasser and B. Lindauer. “Bridging the Gap: A Pragmatic Approach to Gneerating Insider Threat Data,” in Proceedings of the Workshop on Research for Insider Threat, IEEE CS Security and Privacy Workshops, San Francisco, CA, 23-24 May 2013.

[5] E. Kowalski, et al., “Insider threat study: illicit cyber activity in the government sector”, United States Secret Service & the Software Engineering Institute, Carnegie Mellon University, January 2008.

[6] A. Lazarevic and V. Kumar. “Feature Bagging for Outlier Detection.” In Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, 157–166, 2005.

[7] F. T. Liu, K. M. Ting, and Z. H. Zhou. “Isolation Forest.” In Data Mining, 2008. ICDM’08. Eighth IEEE International Conference On, 413–422, 2008.

[8] E. Schubert et. al., “On Evaluation of Outlier Rankings and Outlier Scores,” in *Proceedings of the 12th SIAM International Conference on Data Mining (SDM), Anaheim, CA, 2012*, 1047–1058, 2012.

[9] T. E. Senator et. al., “Detecting Insider Threats in a Real Corporate Database of Computer Usage Activity,” in Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, page 1393-1401, ACM (2013)

[10] T. E. Senator, “On the Efficacy of Data Mining for Security Applications” in Proceedings of the ACM SIGKDD Workshop on CyberSecurity and Intelligence Informatics, Paris, France, June 28, 2009

[11] “SureView Proactive Endpoint Information Protection”, Raytheon, February 13, 2013. Webpage: http://www.raytheon.com/capabilities/rtnwcm/groups/iis/documents/content/rtn_iis_sureview_datasheet.pdf

[12] W T. Young et. al, “Use of Domain Knowledge to Detect Insider Threats in Computer Activities,” in Proceedings of the Workshop on Research for Insider Threat, IEEE CS Security and Privacy Workshops, San Francisco, CA, 23-24 May 2013