2016年2月17日-阅Workshops比Regular Paper稍低，但是同样索引；

2013 IEEE Security and Privacy Workshops

Multi-Domain Information Fusion for Insider Threat Detection

Hoda Eldardiry, Evgeniy Bart, Juan Liu, John Hanley, Bob Price, Oliver Brdiczka
Palo Alto Research Center (PARC)
3333 Coyote Hill Road
Palo Alto, California, 94304
hoda.eldardiry@parc.com

Xerox PARC,著名的施乐帕克研究中心，专注计算机技术研究，位于加利福尼亚的Palo Alto

***Abstract*—Malicious insiders pose significant threats to information security, and yet the capability of detecting malicious insiders is very limited. Insider threat detection is known to be a difficult problem, presenting many research challenges. In this paper we report our effort on detecting malicious insiders from large amounts of work practice data. We propose novel approaches to detect two types of insider activities: (1) *blend-in* anomalies, where malicious insiders try to behave similar to a group they do not belong to, and (2) *unusual change* anomalies, where malicious insiders exhibit changes in their behavior that are dissimilar to their peers' behavioral changes. Our first contribution focuses on detecting *blend-in* malicious insiders. We propose a novel approach by examining various activity domains, and detecting behavioral inconsistencies across these domains. Our second contribution is a method for detecting insiders with *unusual changes* in behavior. The key strength of this proposed approach is that it avoids flagging common changes that can be mistakenly detected by typical temporal anomaly detection mechanisms. Our third contribution is a method that combines anomaly indicators from multiple sources of information.**

***Keywords*—Insider threat detection; anomaly detection; information fusion**

使用的数据来源与类型
不同类型的数据作为不同的domain

I. Introduction

Our research seeks to develop anomaly detection techniques to identify malicious insiders in an accurate and timely manner. We base our anomaly detection techniques on work practice data, i.e., a user's IT traces on her workstation, such as logging on/off, accessing web sites, sending and receiving emails, accessing external devices or files, etc. Work practice data is remarkably diverse and heterogeneous. Data in different categories (which we refer to as 'domains', e.g., 'logon domain' or the 'email domain') exhibits drastically different behavior and demands for different processing and analysis techniques. Combining data from these domains is technically challenging.

不同domain的特征直接拼接是不行的，因为：
1. 数据范围不同，导致计算时影响不同；
2. 若归一化，则假设所有因素地位相同，与实际也有偏差；
3. 特征向量过长，容易造成过拟合和算法复杂度过高；

For example, simply concatenating these features into a single feature vector does not work well because features from different domains may have very different ranges (scales). The lack of proper scaling prevents the model from distinguishing between different types of activities, and limits the model's ability to treat, and reason about, different activity types appropriately. Even with standardization of variables, different frequencies of events in various domains can skew the model. In addition, a large number of features can compromise model accuracy due to overfitting or excessive model complexity, and can lead to performance degradation and scalability issues.

传统的异常检测容易根据一个领域的异常漏报，也容易根据多个domain的异常误报

Existing research on anomaly detection largely ignores the problem of data inhomogeneity and focuses on statistical outlier identification. State of the art techniques define a probability distribution over the data and classify data points with abnormally small probabilities as *anomalies* or *outliers*. Sometimes the anomalies are identified separately in each domain, and combined in an ad hoc manner, i.e. are determined manually, rather than learned automatically from the data. For example, users might be ignored who are outliers in only one domain, or be flagged as anomalous based on an extreme anomaly score.

另外一个严重的问题是：若一个用户在任何一个domain中都正常，则即便他异常也永远不会被识别。

While these techniques are successful in detecting outliers in separate domains, there are limitations. Importantly, users who are not outliers in any of the domains will never be labeled as outliers by these analysis methods. For example, consider the domains of logon data and email text. A typical company might have software engineers, who generally log on to a single machine every day and send emails about software development. It might also have system administrators, who log on to multiple machines each day and send emails about system administration issues. Thus, within the logon domain, logging into either a single or multiple machines daily might not be an anomaly. Similarly, within the email domain, neither software engineering nor system administration are anomalous topics. Therefore, a malicious software engineer who logs into multiple machines daily searching for vulnerable data will remain undetected if each domain is analyzed separately.

我们的工作可以解决上面提到的特征连接和领域检测的问题，更重要的是，我们提出的全局模型在特征构建阶段是单独处理每个domain的

To address the aforementioned limitations, we propose building a global model for the entire set of available domains, and finding outliers in that global model. There are two advantages to this modeling strategy. First, the anomaly scores from multiple domains are combined not in an ad hoc, but rather in a data-driven manner. Second, this strategy allows detection of anomalous behaviors that are not by themselves anomalous in any single domain. Moreover, our proposed approach combines multiple domains at the modeling (learning and inference) and scoring (output/decision) stages, but treats the domains separately at the feature construction (input treatment) stage.

An alternative approach to anomaly detection considers users' behavioral patterns over time. Previous work has focused on detecting temporal anomalies that correspond to a sudden change in a user's behavior when compared to his past behavior. However, we note that analyzing users independently has some drawbacks, since users do exhibit changes in behavior that are not necessarily suspicious. These typical approaches will therefore have a high false positive rate, which

DOI 10.1109/SPW.2013.14

通过行为模式检测异常容易局限于用户自身合理的行为模式改变（开始新项目、新职位等），因此我们提出依据用户的对等体进行分析，通过分析用户的行为改变与对应群体的行为改变来做出判断。

前提假设：用户的工作、角色特点应当反映在数据的各个domain中，并且，相似或相同工作角色的用户的行为数据在各个domain中行为特点应当一致；的GMM分类的Cluster应当近似或一致。这里作者使用Cluster作为特征，即从用户的其余domain的Cluseter来预测某个domain数据所属的Cluster，以此表示其行为的一致性程度，显然，预测准确率越高，说明行为数据越一致。

can increase investigation costs and distract attention from the actual malicious insiders. For example, an employee starts working on a new project, or takes up a new responsibility.

To avoid mistakenly flagging users that change their behavior in a non-malicious manner, we propose modeling the activity change of similar subsets of the population, and evaluating how well a particular user conforms to change patterns that are most likely to occur within the user's subpopulation. In other words, to decide whether a user is suspicious, we compare each user's activity changes to activity changes of his peer group.

Treating each domain separately allows for modularity, as one modeling task can handle one domain independently. This accounts for several notable strengths of our proposed approaches. First, it provides a means of addressing *scalability* issues by allowing each task to be executed on a separate machine, thereby boosting *performance*. Second, treating multiple domains independently allows our algorithm to be *flexibile*, since a different type of model can be applied to each domain as appropriate. In addition, each domain model output can be weighted differently based on the relevance and/or utility of each domain to the problem, and based on the quality of data available for each domain. Moreover, domains can be disregarded if strong correlation with other domains is observed.

Third, treating the features within each domain separately improves classification *accuracy* by reducing the error due to variance in learning, as compared to combining the features from all the domains [23]. Fourth, our proposed approach is more *adaptable* since data from new domains can be considered after running previous models, and the results can be integrated without having to rerun models on previously existing domain data. Fifth, splitting the domains improves models' *simplicity*, which can reduce the risk of overfitting. In addition, it enhances models' *interpretability*.

II. Proposed approach 1: Multi-domain anomaly detection

Our proposed approach treats the domains separately during feature construction, while reasoning about the interdependence betweem the various domains. Although providing a single top-down model that explains all available data is desirable, inference is often difficult in such models due to the large size of the dataset. Often it is simpler to perform inference in stages.

目前的理解：分域单独假设高斯概率密度，然后使用最大似然方法计算出各自的概率；使用计算MAP值的方法确定最终GMM的各个Compnont的组合系数，最终确立GMM模型；

Therefore, we adopt a two-stage modeling process. First, we obtain the maximum likelihood estimate (MLE) for model parameters within each individual domain separately. Second, we use these parameters in the global model as if they were fixed. If the data in each domain is relatively unambiguous (i.e., allows to determine that domain's sub-model with sufficient accuracy), the loss in accuracy is small. Our proposed single-domain models are based on a gaussian mixture model (GMM), where the maximum a posteriori probability (MAP) values for the cluster to which each user belongs within each domain are obtained at the first stage. Our proposed global cross-domain model is based on these MAP cluster indices.

Finally, we combine the output from all the models using our proposed fusion scheme presented in section II-A3.

A. Methodology

We define the problem as follows. An anomalous user is one that exhibits inconsistent behavior across the domians. The intuition is that user activity should reflect the user's job role in any domain, and users with similar job roles exhibit similar behavior within each domain. We also expect that each user should belong to the cluster of the same subset of users across domains. For example, a user that behaves similar to (and belongs the the same cluster as) engineers within 'http' domain, based on her web browsing activity, should also belong to same cluster as engineers within the 'logon' domain. If such user belongs to a different cluster in the 'logon' domain (say the technical support users), this can indicate a suspicious behavior in which an engineer logs on to multiple machines. We define this as across-domain behavior inconsistency. We formulate the problem as a classification task, in which clusters are used as features. We predict a user's cluster in one domain from her cluster indices in all other domains. The prediction accuracy for a user's cluster in each domain reflects her behavior consistency across domains.

We consider the case where the job roles are latent, and we use clustering of user activities within each domain to model the hidden job roles. We expect that users with similar job roles will belong to the same cluster within each domain. The MAP cluster indices from single-domain modeling for each user u form a cluster vector c_u, where c_{u_i} is the MAP cluster index for user u in domain i. For user u, we say that domain i is consistent with other domains, if its cluster index c_{u_i} is predictable from other domains' cluster indices $\{c_{u_j}\}_{j \neq i}$. In the simplest case, we may use cluster indices of other users $w \neq u$ to learn a mapping from $\{c_{w_j}\}_{j \neq i}$ to c_{w_i}, and then check whether this mapping generalizes to user u.

In this manner, we determine whether domain i is consistent with the other domains for this user. If not, the user is labeled as an anomaly. The anomaly score in this case is determined based on the overall prediction accuracy in domain i for all other users. The idea is, if the domain is difficult to predict in general, then incorrect predictions should not be penalized as severely; in contrast, for a very predictable domain any incorrect predictions may be quite suspicious. In this case, even though the anomaly scores are computed per domain, they are informed by other domains and thus can take into account information from all domains.

Finally, we combine the anomaly information captured from each domain to make a final decision about each user. We use the fusion method described in II-A3. We present the three phases of our anomaly detection methodology next.

1) Using clustering to model user behavior and peer groups: The first phase involves clustering users in each domain separately based on their activity within a given domain. The goal of this phase is to discover peers. We apply k-means clustering to the aggregate feature vectors generated, as explained in section IV-A.

2) Multi-domain inconsistency detection: The second phase identifies the predictability of user u in domain i. The prediction is formulated as a multi-label classification task, in which a classifier is trained from the cluster information from all-but-one domains to predict the cluster information in

the remaining domain (the target domain). We propose three models that differ in the granularity of cluster information used as features for learning and evaluation.

Discrete model: (discrete features, discrete evaluation). Our discrete model uses cluster labels from the observed domains as features for learning, and predicts cluster labels to evaluate user predictability. The predictability is measured as the Hamming distance between the prediction and the observation, i.e., 0 if the prediction is correct, and 1 otherwise.

Hybrid model: (discrete features, continuous evaluation). Our hybrid model uses cluster labels from the observed domains as features for learning, and predicts cluster labels to evaluate user predictability. However the evaluation is not based on just whether or not the true cluster is predicted, but instead, how well the true cluster is predicted. This is in essence a density estimation problem. The predictability is measured as 1 minus the likelihood of observing the true cluster index given the cluster index of its peers.

Continuous model: (continuous features, continuous evaluation). The continuous model uses a vector of cluster probabilities as features. For the target domain, it also predicts the cluster probability vector.

3) Information fusion for combining anomalies from multiple domains: The multi-domain cross validation work described above characterizes the predictability of a user in a given data domain. In order to combine anomalies detected from multiple sources, we propose a method to combine predictability scores as a weighted sum. The idea is based on the TF/IDF (term frequency–inverse document frequency) scheme, reflecting the relative importance of a word to a document in a corpus. The TF/IDF value is proportional to the frequency that a word appears in the document, but is offset by the global frequency of the word in the corpus. Words that are frequent and yet unique to the document have high TF/IDF scores. This property justifies the use of TF/IDF as a weighting factor in information retrieval and text mining.

Our problem is similar in essence: given multiple anomaly scores for each user, drawn from multiple sources of information provided by the various domains, the goal is to combine the scores into a final score for each user. We develop an approach similar to TF/IDF. Given m scores from m sources for each of n users, the algorithm outlined below proceeds in two steps. The first step calculates the weights for each source s to reflect the differences discussed in the challenges above. The second step computes, for each user i, the weighted anomaly score w for each source s, then aggregates the weighted anomaly scores from each source to compute the final anomaly score f.

Algorithm 1 Fusion Algorithm

for $s = 1$ **to** m **do**
  for $i = 1$ **to** n **do**
    $m_i^s = Miss\, prediction\, score\, for\, user\, i\, in\, source\, s$
  $p^s = \log \frac{n}{\sum_{j=1}^{n} m_j^s}$ {source predictabililty}
for $i = 1$ **to** n **do**
  for $s = 1$ **to** m **do**
    $a_i^s = m_i^s * p^s$ {adjust to reflect source predictability}
$F \leftarrow \emptyset$
for $i = 1$ **to** n **do**
  $f_i = \sum_{s=1}^{m} a_i^s$ {final user score}
  $F = F \cup \{f_i\}$
return F

III. Proposed approach 2: Unusual change detection

We note that while a particular behavior may not be suspicious, a change in behavior that is rare can be. In this section we propose a method that detects users with unusual changes in behavior. In this context, changes that are common among peers are considered common changes, and the goal is to detect changes that are less likely to happen within the group to which a user belongs. The key strength of our proposed approach is that it avoids detecting common changes that can be mistakenly detected by typical temporal anomaly detection mechanisms.

A. Methodology

We define the problem as follows. An anomalous user is one that exhibits changes in behavior that are unusual compared to that user's peers. The intuition is that user activity should reflect the user's job role in any domain, and users with similar job roles should exhibit similar behavior changes within each domain, over time. That is to say, peers will not be expected to exhibit similar changes in behavior at each similar time episodes, but they will be expected to do so over longer time intervals. Our model does consider that peers are expected to experience similar changes, that do not necessarily have to take place at the same time.

We clustering users based on their activities, such that a cluster that a user is assigned to indicates the type of behavior this user exhibits. In addition, a change in user behavior is indicated by a chnage in the cluster that this user gets assigned to. Over a period of time, peers are expected to transition between the same subset of clusters. For examples, engineers will be seen to transition between clusters 2 and 4 in the logon domain, and clusters 3, 4 and 5 in the email domain. So an engineer that transitions between 2 and 5 in the logon domain is considered suspicious. The less likely this transition is among peers, the more suspicious it is. We build an empirical model describing the statistical distribution of clusters and cluster transitions over time.

1) Using clustering to model user behavior and peer groups: Similar to our earlier discussion in section II-A1, we consider the case where the job roles are latent, and use clustering of user activities within each domain to model the hidden job roles. However, in this setting we model user temporal state by clustering users' daily behavior features instead of clustering based on features that are aggregated over the entire time span. We construct a transition probability matrix Q_d for each domain d by computing the transition probabililties $q_d(c_k, c_m)$ between each possible cluster pair (c_k,c_m) by counting the number of such changes aggregating over all users and each time instance. Then we use two methods to model user changes and detect temporal anomalies.

2) Modeling behavior change: We model behavior change within each domain separately. For each domain, we look at the

TABLE I: Domain features

Domain	Features
Logon	#Logons, #PCs logged on, #after hour logons, #logons on user's PC, #logons on other PC(s)
Device	#device accesses, #PCs with device access, #after hour device accesses, #device accesses on user's PC, #device usage on other PC(s)
File	#file accesses, #PCs with file accesses, #distinct files, #after hour file accesses, #file access on user's PC, #file accesses on other PC(s)
HTTP	#web visits, #PCs with web visits, #URLs visited, #after hour web visits, #URLs visited from other PC(s)
Email Sent	#emails, #distinct recipients, #internal emails, #internal recipients, #emails sent after hour, #emails with attachment(s), #emails sent from other PC(s)
Email Received	Similar to email sent

cluster to which a user belongs at each day. We then compute the likelihood of transitions between clusters from one day to the next. We use two approaches to model behavior change.

Markov model. The first method models user behavior over time as a Markov sequence, where a user will belong to one cluster (or state) each day, and transition between clusters (or states) each day. A cluster (or state) reflects user's behavior on a particular day. For each user, the total likelihood of all the transitions made by that user over the entire time span is computed using Q_d. The anomaly score S_d^u for each user u within domain d is calculated by estimating the user's total likelihood. The anomaly $S_d^u = p_d(c_0)\prod_{t=1}^{n-1} q_d(c_t^u, c_{t+1}^u)$ where $p_d(c_0)$ is the prior probablility of being in cluster c_0 which is the start state for user u. Users are ranked based on their scores, thus a user with the rarest transitions compared to her peers will be the most suspicious.

Rarest change model. The second method penalizes a user for the least likely transition behavior change, and uses that "rarest" transition to compute the anomaly score: $S_d^u = min_{t=1}^{n-1} q_d(c_t^u, c_{t+1}^u)$.

3) Information fusion for combining anomalies from multiple domains: Given the two approaches discussed above, we have multiple rankings for the same set of users from the different domains. To combine information from different domains we generate the final rank for each user based on the worst rank from all domains. $S_{final}^u = min_d\{S_d^u\}$. The final ranking for each user thus reflects the highest suspicion indicator score across all the domains.

IV. Experiments

For our first experiment, we applied the discrete and hybrid models on the domains described below in section IV-A. We computed the predictability for each individual user in any given domain. This enables the identification of anomalies. Users with lots of mispredictions are labeled as anomalous users. For our second experiment, we applied the Markov and rarest change models on the domains described below in section IV-A. We computed the likelihoods for each user transitions and used them to rank users such that more suspicious users are ranked higher than less suspicious ones.

A. Datasets

We validate our first multi-domain anomaly detection method using a synthetic dataset (dataset 1) of 1000 users, and we validate our unusual change detection method using a real dataset (dataset 2) of about 4600 users. Due to the lack of ground truth, a common methodology is to inject synthetic anomalies. Both datasets contain synthetically injected anomalies based on several real world malicious behavior scenarios. Note that the scenario labeling was not made available to the learning or modeling algorithm.

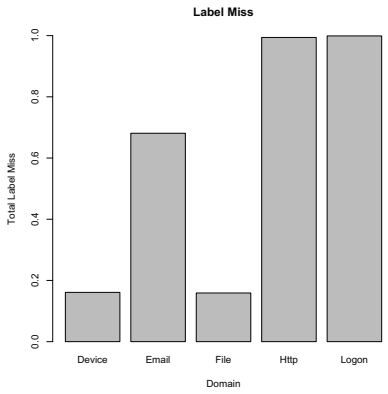
Work pratice data falls into five categories, as listed below. Each event is tagged with auxilary information such as user id, host PC id, activity code (whether it is a logon/logoff, or file upload/download etc), and a timestamp.

- Logon and logoff events.
- Usage of removable device such as USB drives or removable hard disks. Device name and type are logged with each usage event.
- File access events: e.g., file created, copied, moved, written, renamed, or deleted. For each file access record, file name, path, type, and content are logged.
- Http access events, tagged with URL and domain information, activity codes (upload or download), browser information (internet explorer, firefox, or chrome), and whether the website is encrypted.
- Email sent and viewed are tagged with from address, to address, cc/bcc address, subject line, sent date, text, attachment info, and whether the email is encrypted.

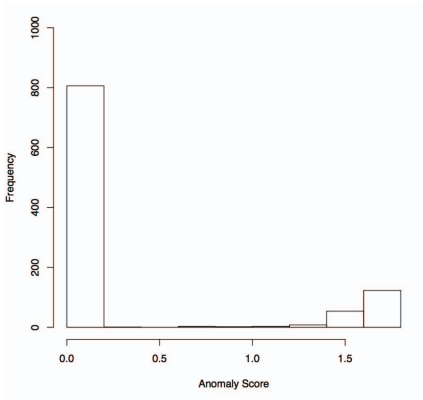
Furthermore, our system associates a set of tags to raw events. For instance, we label (1) whether the event happens after normal working hours and (2) whether the event happens on a user's own designated PC, someone else's designated PC, or a shared PC. As malicious insiders often need to steal information from their colleagues, labeling the host PC is semantically important. In addition, events concerning activities external to the organization (e.g., email sent to or received from external addresses, file upload/download from external URLs) are labeled. In real world setting, users often have a multitude of events. For instance, in our dataset 2 of 4600 users, the data volume is approximately 89 million records per day. To simplify processing, we bin events into user day records. For each (user, day) pair, we compute aggregated statstics as shown in Table I.

B. Multi-domain cross validation method results

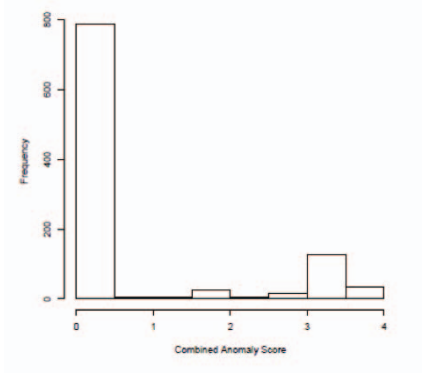
Figure 1a shows the general domain predictability over the entire population based on unadjusted scores. The varying level of predictability explains that using different domains have different levels of importance in identifying an anomaly. Device and File domains are predictable (with the least amount of mispredictions over the user population), while Logon and



(a) Misclassification of the discrete model on the various domains

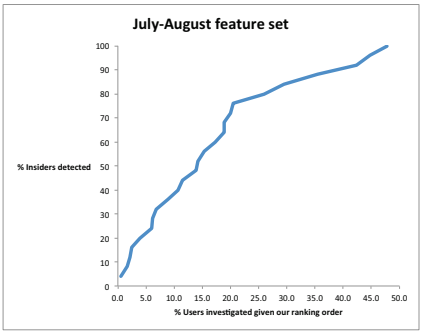


(b) Anomaly detection using the device domain

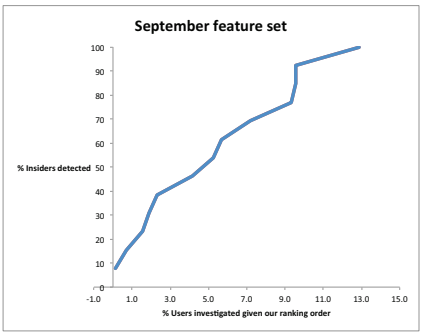


(c) Anomaly detection using the combined anomaly scores

Fig. 1: Multi-domain cross validation results



(a) All anomalous users are detected by scanning 50% of the population



(b) All anomalous users are detected by scanning 13% of the population

Fig. 2: Unusual change detection: investigation effort and anomaly detection return

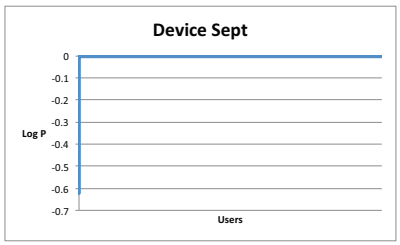
HTTP domains seems hard to predict. This is probably because users differ a lot in their logon and http behavior, but are more uniform in device usage and file access. This emphasizes the necessity of our fusion approach, which considers this domain variation when computing the final anomaly score.

Next, we show how we identify individual anomalies by looking at the anomaly score distribution. Figure 1b plots the anomaly scores for the entire user population for the device domain, and Figure 1c plots the combined anomaly score using our fusion approach. Most users have low anomaly scores. The users on the tail with high anomaly scores are identified as suspicious.

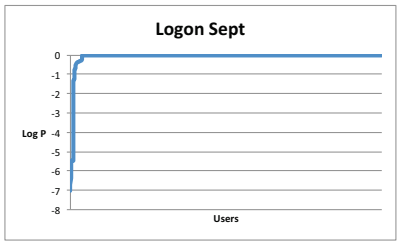
C. *Unusual change method results*

We show the results of applying the Markov model approach to 2 sets of data collected on two different months, and the injected anomalies exhibit one of twelve scenarios. Figures 2a and 2b show the investigation effort on the x-axis and the return on the y-axis. The effort is represented by the percentage population that needs to be investigated and the return is the percentage insiders detected (which is the suspicious rank in the population of each insider). Figure 2b shows that if we stop at screening the top 2% of the population, we would find 4 out of 13 anomalous users, and 4 out of 5 malicious scenarios.

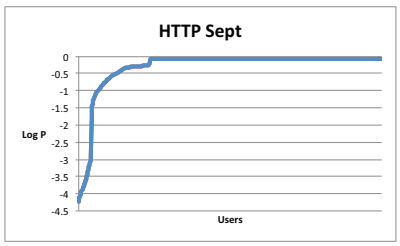
Figure 3 plots the result of the Markov model. The plot shows the transition likelihood for each user. Here, the



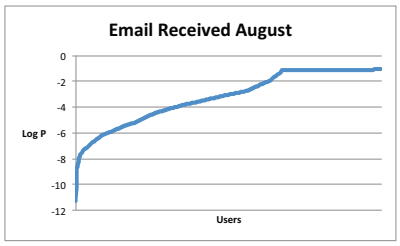
(a) Device



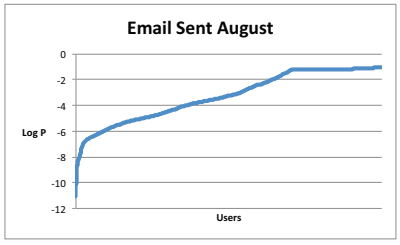
(b) Logon



(c) HTTP



(d) Email received



(e) Email sent

Fig. 3: Markov model log probability plot for user transition likelihoods

users are sorted based on their likelihoods. Users are ranked separately within each domain based on these likelihoods. The ranking indicates the suspicious level. Some domains are highly discriminative, for example 'device' and 'logon'. Other domains are weakly discriminative, for example 'email-sent' and 'email-received'.

V. Conclusion

Our proposed approach improves prediction accuracy by combining information from multiple domains, and is able to detect anomalies that are not apparent in any single domain. Previous anomaly detection methods for work practice data treated each domain separately. The main novelty of our proposed method is that it combines model information from each domain, rather than only anomaly scores from each domain. As a result, it is able to determine anomalies that are not apparent in any single domain, but only manifest in discrepancies across domains. In addition, how anomaly scores are combined across domains is determined automatically, in a data-driven manner. As a result, discrepancies that are common in the data will be filtered out. For example, if (due to high volumes of the data) it is common for a user to be an outlier in at least one domain, then users who are anomalous in only one domain will not be flagged as anomalous overall.

VI. Related work

There are many novel technologies for detecting malicious insider behavior. Such behaviors are relatively rare in the broader user population and so techniques for anomaly detection can be applied. For example, [1] uses machine learning to recognize malicious intent in information gathering commands, [2] detects anomalies in document accesses and queries with respect to a Hidden Markov Model of text content, and [3] models user processes and flags deviations from the model. There are also many commercial tools that used rule-based techniques (e.g. see, for example, [4], [5], [6]). These techniques detect malicious insider behavior through monitoring network activity and the use of enterprise applications.

Despite these tools, the number of incidences of insider attacks continues to rise in the government and commercial sectors. As an example, a recent survey found that 28% of respondents would take sensitive enterprise data to negotiate a new position in the event their employer terminated their current position[7]. Indeed, insider attacks have been the most frequent (CSI 2007, [8]) or second most frequent (CSI 2008, [9]) source of security incidents in recent years in the United States.

In attempt to augment this basic rule-based approach, some works have introduced decoys onto the network to entrap adversarial insiders[14], [15]. Please see [16] for a survey of this work. In addition, various models of adversarial insiders have been developed. These models include physical behaviors that are indicators of adversarial intent (e.g. foreign travel, signs of wealth) [17], as well as variables related to motivation, personality, and emotion [18], [19], [20], [21]. While all these models are valuable, none incorporate all of the possible situational triggers, context variables and indicators. We believe such attributes are necessary to establish a close connection between psychology and behavior.

Social network analysis (SNA) is now a well-established research tool [10] with a long track record in identifying key individuals in organizations based on their communication patterns (e.g., [11]). It has been fruitfully used by the defense and intelligence community to study covert networks (e.g.,[12]), in an attempt to target the most important enemies and disrupt their organization [13]. The concept of combining data from multiple domains has been mentioned in previous work [22]. However, the work we present here extends beyond the conceptual arguments presented in earlier published work, to yield methods and algorithms that are implemented and applied in real-world scenarios.

ACKNOWLEDGMENT

The authors gratefully acknowledge support for this work from DARPA through the ADAMS (Anomaly Detection At Multiple Scales) program funded project GLAD-PC (Graph Learning for Anomaly Detection using Psychological Context). Any opinions, findings, and conclusions or recommendations in this material are those of the authors and do not necessarily reflect the views of the government funding agencies.

REFERENCES

[1] M. Salem and S. Stolfo. Masquerade attack detection using a search-behavior modeling approach. Columbia University Computer Science Department, Technical Report # cucs-027-09, 2009.

[2] P. Thompson. Weak models for insider threat detection. Proceedings of the SPIE Vol. 5403, Sensors and Command, Control, Communications and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense III, 2004.

[3] P. Bradford and N. Hu. A layered approach to insider threat detection and proactive forensics. ACSAC 2005.

[4] Raytheon Oakley Systems SureView. http://www.raytheon.com/ capabilities/products/cybersecurity/insiderthreat/products/surview/ (Retrieved Feb 14, 2012)

[5] Lanxoma. Intelligent Desktop Surveillance. http://www.lanxoma.com/ (Retrieved Feb 14, 2012)

[6] PacketMotion. http://www.packetmotion.com/ (Retrieved Feb 14, 2012)

[7] Cyber-Ark Press Release, November 23,2009. http://www.cyber-ark.com/news-events/pr_20091123.asp (Retrieved Feb 14, 2012)

[8] Computer Security Institute (CSI) Computer Crime and Security Survey, 2007. http://i.cmpnet.com/v2.gocsi.com/pdf/CSISurvey2007.pdf (Retrieved Feb 14, 2012)

[9] Computer Security Institute (CSI) Computer Crime and Security Survey, 2008. http://www.docstoc.com/docs/9484795/CSI-Computer-Crime-and-Security-Survey-2008 (Retrieved Feb 14, 2012)

[10] Wasserman, S. and K. Faust (1994). Social network analysis: methods and applications. Cambridge, UK, Cambridge University Press.

[11] Burt, R. (1995). Structural holes: the social structure of competition, Harvard University Press.

[12] Sparrow, M. (1991). The application of network analysis to criminal intelligence: an assessment of the prospects. Social Networks 13: 251-274.

[13] Carley, K. (2003). Destabilizing terrorist networks. 8th International Command and Control Research and Technology Symposium, National Defense War College, Washington DC.

[14] L. Spitzner. Honeypots: Catching the insider threat. ACSAC 2003.

[15] B. Bowen, S. Hershkop, A. Keromytis and S. Stolfo. Baiting inside attackers using decoy documents. SecureComm 2009.

[16] M. Salem, S. Hershkop and S. Stolfo. A survey of insider attack detection research. Insider Attack and Cyber Security: Beyond the Hacker, Springer, 2008.

[17] M. Marbury, P. Chase, B. Cheikes, D. Brackney, S. Matzner, T. Hetherington. B. Wood, C. Sibley, J. Marin, T. Longstaff, L. Spitzner, J. Haile, J. Copeland and S. Lewandowski. Analysis and Detection of Malicious Insiders. Technical Paper, Case #05-0207.

[18] K. Herbig. Changes in espionage by Americans: 1947-2007. Department of Defense Technical Report 08-05, March 2008.

[19] K. Herbig and M. Wiskoff. Espionage against the United States by American citizens 1947-2001. PERSEREC Technical Report 02-5, July 2002.

[20] S. Band, D. Capppelli, L. Fischer, A. Mooore, E. Shaw and R. Trzeciak. Comparing Insider IT Sabotage and Espionage: A Model-Based Analysis. Technical Report CMU/SEI-2006-TR-026.

[21] M. Keeney, E. Kowalski, D. Cappelli, A. Moore, T. Shimeall, S. Rogers. Insider Threat Study: Comuter System Sabotage in Critical Infrastructure Sectors. U. S. Secret Service and CERT Coordination Center/SEI.

[22] E. E. Schultz. A framework for understanding insider attacks. University of California-Berkeley Lab. Compsec 2002, London, England.

[23] H. Eldardiry and J. Neville. An analysis of how ensembles of collective classifiers improve predictions in graphs. The 21st ACM International Conference on Information and Knowledge Management (CIKM 2012). 29 October 2012