Insider threat detection: Where and how data science applies

Derek Lin
is chief data scientist at Exabeam, building data science-driven defence against cyberthreats. His research interests and experiences include anomaly detection, insider threat detection and behaviour analytics. Derek's prior machine learning works from Pivotal Software include the consultation and building of data science-based solutions for custom security use cases for large enterprises. Before Pivotal Software, Derek was with RSA Security developing analytic systems for risk-based online banking fraud detection, data loss prevention, voice-biometrics security and speech and language processing. Derek holds numerous patents and publications in the areas of IT operations and security. He graduated with a Bachelor's and a Master's degree in electrical engineering from the University of Southern California.

Exabeam Inc., 2 Waters Park, #200, San Mateo, CA 94403, USA
Tel: (844)392-2326; E-mail: derek@exabeam.com

Abstract Insider threats are one of the top concerns of enterprise security. Traditional means of addressing general security threats, such as the use of signature matching and correlation rules, fall short when detecting insider threats. New possibilities for detecting insider threats have emerged as a result of the data-driven approach to security problems. Insider threat activities are multifaceted and require that security teams address the problem on multiple fronts. This paper introduces four areas where data science can be applied when building a system that detects threats. These four areas include the use of statistical analysis for anomaly detection, contextual information derivation for network intelligence, specific threat detection use cases, and meta learning for false positive control. Example use cases within each category are described, as well as how data science is used to approach them. The goal of this paper is to provide the general security audience with an overview of data science applications for insider threat detection.

KEYWORDS: insider threat, data science, machine learning, SIEM, user and entity behaviour analytics

INTRODUCTION

An insider threat is a threat to an organisation that comes from users who have access to the corporate network. The threat may be coming from external adversaries who have compromised internal user accounts to conduct illicit activities, or from legitimate employees and contractors who have accessed network resources with malicious intent, or even from employees who have used the network resources carelessly in ways that unintentionally violate the acceptable security policy. These insider threats can result in compromised intellectual properties, leaked sensitive data and a tarnished company reputation.

While security policies, procedures and controls[1] are the essential first line of defence against insider threats, preventative measures do not detect threats. Conventional means of detection that rely on signatures do not apply, as insider activities have no factual signatures. The traditional use of correlation rules in security information and event management systems (SIEM) offers little detection power against insider threats for

several reasons. First, correlation rules are static and therefore are ineffective against the dynamic nature of user activities. Second, the same rules are applied to all entities and do not consider the local context, which typically results in a high volume of false positives. Third, the correlation rules usually are defined over a short time horizon and are unable to meaningfully consider long-term historical profiles for detection. User activities on the network are dynamic in nature, creating a network environment that is in a constant state of change. As a result, conventional approaches that are static and applied without local contexts cannot find insider activities of interest.

On the other hand, the new generation SIEM, with its rich data already collected, is no longer limited to serving only regulatory and compliance purposes and now can provide new analytical possibilities for detecting insider threats. This new approach is generally termed 'user and entity behaviour analytics' or UEBA.[2] Myriad network and security logs can be collected if they are not already in existing SIEMs — for example, authentication and virtual private network (VPN) logs, database and file access log, cloud application log, physical badging log, as well as external security log. If analysed optimally, these types of data provide a 360-degree view of a user's behaviour. By monitoring the long-term activities of individual users, examining the local contexts and exploring relationships among users and entities, a data science-based system finds anomalies of interest with minimal false positives. A system that analyses logs needs both algorithms and an infrastructure to support the processing of the volume, variety and velocity of SIEM data. This paper describes the four categories of data science applications relevant to insider threat detection: the use of statistical analysis for anomaly detection, contextual information derivation for network intelligence, specific threat detection use cases and meta learning for false positive control. Discussions on the infrastructure and platform choices, as well as the actual mathematical formulations, are, however, beyond of the scope of this paper.

ANOMALY DETECTION

Insider threat detection is an anomaly detection problem in which labelled data indicating malicious user activities on the network are either non-existent or too few to be useful for supervised learning. Unsupervised learning that profiles a user's normal behaviour in order to alert for deviations is the essence of anomaly detection. A large body of unsupervised learning methods for anomaly detection is available, differing in ways of deriving learning features, behaviour profiling methods and outlier detection modelling techniques;[3] however, key user requirements limit the analytical framework choices for a practical insider detection system. An effective detection system would incorporate the following factors:

- The generated alerts must be self-explanatory. Without good interpretability, it is difficult for security analysts to understand and take action;
- The learning is continuous and adapts to the dynamic activities of users and entities on the network;
- The framework allows for flexibility in accommodating new data sources;
- The framework facilitates the addition or change of detection indicators by analysts.

These requirements dictate a white-box, analytical framework that is self-learning, extensible and easily configured and modified by security analysts. Some options include anomaly detection frameworks that are suitable for enterprise operations and are based on a collection of statistics-based and fact-based behaviour indicators[4,5] that can be easily added to and changed. For example, statistical indicators reveal whether there is

an abnormal application logon for the user, or whether the number of bytes transferred is abnormal. Across available enterprise logs, tens to hundreds of indicators can potentially be designed and implemented, each capturing a certain aspect of a user's behaviour. Then the alerts from triggered indicators are aggregated together with scoring mechanisms that derive a user's session score for prioritisation.

For such a framework, each statistical indicator measures the degree of abnormality of a current user event against the user's behaviour profile. A profile is a mathematical representation of certain aspects of a user's behaviour. For example, a profile that tracks a user's application usage behaviour can be a simple count-based histogram that shows the frequency distribution of applications a user has logged into. Or it can be a Markov graph structure that preserves the temporal or sequential information of how a user moves from one application to another in sequence. Or it can be a tree of hierarchical parent–child relationships among application processes. Count-based histogram profiles are a good choice based on their simplicity and effectiveness.

There are three data types to consider when building histograms. Some require machine-learning treatment. Figure 1 illustrates histograms for the categorical data, numerical data and time-of-week data.

Categorical data

Histograms are most naturally defined by categorical data. Examples include: frequency counts on locations from which a user has logged in, past network devices accessed by the user, applications a user has logged on, and command and processes executed by a user.

Numerical data

Examples of numerical data for profiling include the number of servers accessed in a day, the number of bytes transferred via ftp, the number of e-mails sent and the duration of a user's session determined by the time between log in and log out. Unlike the categorical data, the value for the numerical data is potentially unbounded. Fitting the non-discrete data into histograms first requires clustering the numerical data points into bins or ranges of numbers, and then converting the ranges into discrete bins to construct the histogram. For example, an algorithm, such as agglomerative clustering, is applied. In addition to such a nonparametric approach of the histogram-based profiling of numerical data, the parametric approach for profiling the data offers another option. In this case, a few parameters are to be calculated from the data to describe a distribution function that fits the numerical data distribution.

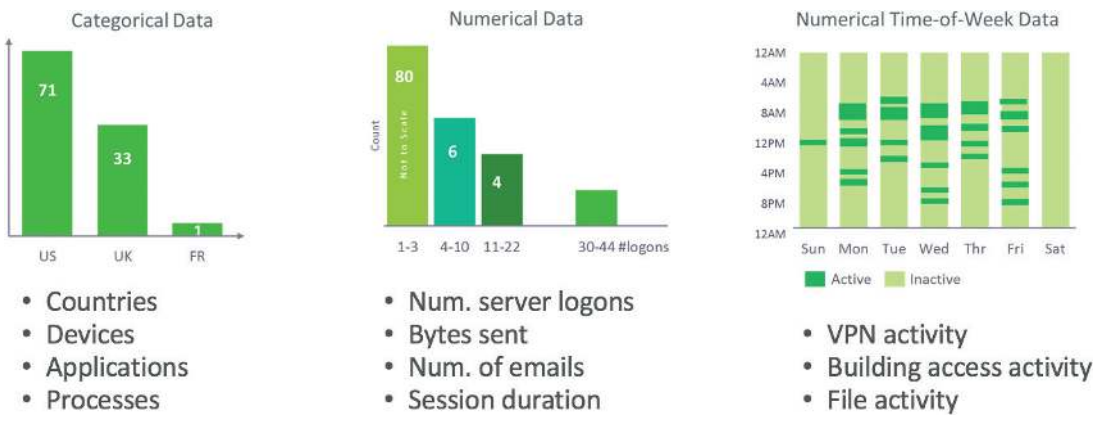

Figure 1: Histograms with different data types

Time-of-the-day data

Examples of time-of-the-day data include the hours and weekdays a user connects via VPN, the time a user enters a building and the time a user has a file activity. Profiles based on this data aim to learn the normal active time blocks of a user's activity. Histogram-based profiling of the time-of-the-day data is similar to that of numerical data, except that there is a need to take the daily and weekly boundaries into consideration, so that time activities crossing the boundaries are not split into two clusters.

Given the constructed behaviour profiles, statistical indicators are defined to evaluate whether the current user activity is abnormal when compared to the profiles; alerts are raised accordingly. If profiles are histogram-based, as described above, the abnormality of an event against a histogram is measured by the *p*-value — a common statistics metric used in hypothesis testing.

As the detection system raises alerts from triggered indicators throughout a user's session, a mechanism is chosen to combine the alerts to produce the final session score. As an option for a scoring mechanism, each indicator is preassigned an expert-determined score. A session's total score is simply the cumulated scores from the triggered alerts. Based on the scores, analysts prioritise the sessions to review. While assigning human-determined scores to alerts may seem rigid, a later section of this paper describes a data-driven scoring method to complement the expert-assigned scoring, as illustrated in Figure 2. The detection system benefits from combining both human knowledge and data-driven information.

In short, different choices for anomaly detection frameworks are possible. This section describes a viable framework for making a detection system operational.

CONTEXT ESTIMATION

Context information, such as labelled attributes and properties of network users and entities, serves an important role in the triage and review of alerts. In addition, this information is leveraged as an integral part of anomaly detection indicator design. The context information is not always available, however, because it is either not maintained or not up to date due to its low operational priority in IT administration. Despite any lack of information, in many cases SIEM's logged data can be leveraged to derive and estimate the contextual information needed for the detection system. And this bridges the gap. Some data science-use cases for context estimation are given below.

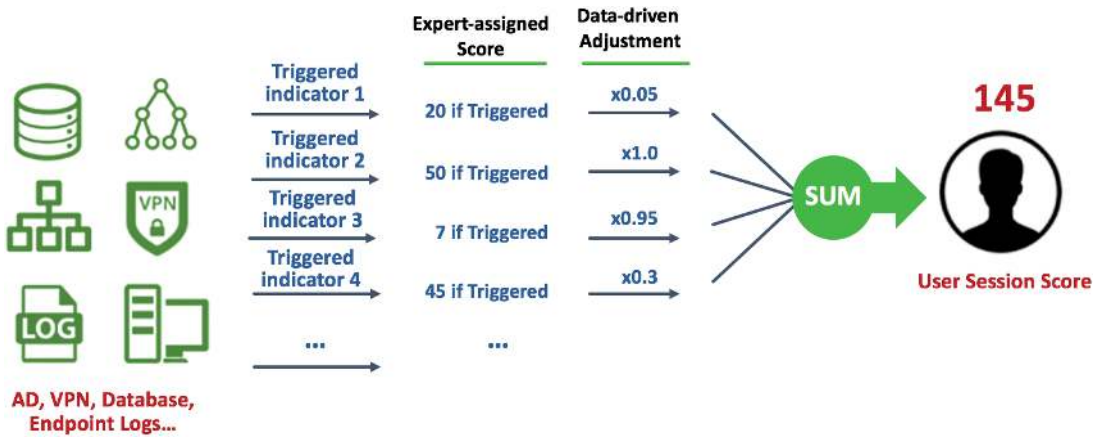

Figure 2: Scoring based on expert knowledge and data-driven adjustment

Service account estimation

Service accounts and human user accounts show different behaviours. Knowing the nature of the account helps teams to design better statistical indicators. For example, a popular anomaly detection indicator is whether an account has accessed an asset for the first time. But it is not unusual to see a service account connecting to new assets for the purpose of software updates. Without considering the context of the account type, this indicator will generate a high rate of false positives. In large IT environments, it is not surprising that only some of the service accounts are tracked and known. The analytical goal is to mine the data to identify the unknown service accounts. One method analyses the textual data in active directory (AD) by looking for additional accounts that share similar account properties to those in the AD's key-value attribute descriptions and to those in the known service accounts. Another method performs the account classification based on the behaviour cues that service accounts likely generate higher volume of events than human user accounts, or they exhibit periodic activities, or they are active throughout the day, etc.

Host type estimation

Knowing the type of machine a user connects to, whether a personal workstation or a server, provides a useful context for assessing risk; however, explicit labelling of machine types is not always an option in IT practice. Nonetheless, it is useful to explore the behaviour differences of the machines to classify the machine types. For example, a server is expected to have different behaviour cues from a laptop, such as in the number of users and other machines it connects to, or in the volume of events it has generated.

Peer group estimation

Peer analysis is an important part of user and entity behaviour analytics. While it is noteworthy that a user is observed to perform some task for the first time, the risk is elevated if the user is also the first in their peer group doing so for the first time. Performing peer analysis is predicated on knowing to which peer groups a user best belongs. Although AD maintains the data of user peer attributes — department, location, title, security group, etc. — the information is not optimal. Some peer groups are very large and some are singleton groups — and some peer attributes have less coverage than others. For the purpose of peer analysis, the challenge is in determining which are the best peer groups for a user. One analytical possibility compares a user's past behaviour in machine access patterns to those of members in peer groups. The peer groups whose members have, on average, the closest matching behaviours to the user are suitable for peer analysis.

Account resolution

A user on the network may own more than one network account. For example, an administrative user normally maintains multiple accounts: one for their normal user-level, day-to-day activity and another for their administrative tasks. If the multiple accounts are not recognised as belonging to one person, the detection system will analyse their activity as separate streams. An individual's account activity streams may seem innocuous when analysed separately but can be risky when combined and viewed together as a single-user activity, which is a problem. Therefore, it is important to resolve multiple accounts to a single user. One analytical approach employs market basket analysis, in which two separate accounts that tend to co-occur on the same machine in history are likely to belong to the same person.

Personal e-mail identification

Associating or linking an external personal e-mail address to an employee

provides an important context for the data exfiltration-use case. For example, the risk from an 'unusually large attachment is sent' e-mail alert from a user should be further elevated if the destination e-mail is their personal e-mail address. Maintaining a database for personal e-mail addresses is not a common corporate IT practice. But from an analytical point of view, by examining the co-occurring frequency of a pair of sender and receiver addresses in e-mails in history, it is possible to establish relationships between external and work e-mail accounts. In short, new or additional contextual information discovery can improve the precision of a statistical indicator and can help in alert evaluation and understanding.

TARGETED DETECTION

Anomaly detection can find malicious or otherwise interesting activities of an unknown nature. In contrast, targeted detection finds malicious or otherwise interesting activities with known behaviour characteristics, either provided by the domain knowledge or through observation of past repeating activities that exhibit similar behaviour properties. Such detection-use cases have a narrowly defined scope on specific data sets, and they can benefit from machine learning for targeted learning and detection. A large body of research works[6] on various detection scenarios exists. Enumeration of the past works is beyond the scope of this paper. For the purpose of illustration, however, some examples of use cases are described below to explain how data science applies.

Many initial attacks from external adversaries start with the use of phishing domains, which can compromise accounts. Past work in phishing domain detection[7] uses supervised learning to build classifiers to learn both lexical and connectivity features from known populations of phishing and non-phishing domains. As another example in the targeted detection, domain names created by domain generation algorithms are known to be used by malware to communicate to external command-and-control servers. Deep learning[8] or simpler bigram frequency analysis[9] has been used for their detection. Other targeted detection-use cases focus on specific data types, requiring more specialised learning methods (for example, database queries,[10] system file access events[11] or user commands[12]).

FALSE POSITIVE CONTROL

The effectiveness of an anomaly detection system depends on the quality of designed statistical indicators. Indicators have varied performance, however; some are more prone to false positive alerts than others. In aggregating alerts to generate the final risk scores, a high volume of false positives from the noisy alerts can render the detection system ineffectual — or, at its worst, useless for analysts.

There are at least two directions to address the false positive issue though meta learning where we learn from the detection system's own behaviour to improve itself. The first method reduces the scoring contribution of noisy alerts to an overall score. As illustrated in Figure 2, expert-assigned scores for triggered indicators can be sum-aggregated to make up the user's overall session-level or day-level score. But an additional scoring adjustment helps improve the scoring quality. For example, the data loss prevention alerts from security vendors are known to be noisy in practice. Scores from these noisy alerts dominate and crowd out the detection signal from other higher-precision alerts. The net effect is a high volume of false positive user sessions for analysts to review. One solution leverages the historical information of the triggering and frequency of alerts across the population and within a user's history. If an alert is observed to trigger often across many users, it is deemed to be less interesting. It is then reasonable to adjust the expert-assigned score with a penal weight. If an alert has

rarely been triggered in the past, it makes sense to impose little or no penalty at all to its score contribution. Additionally, a penal weight can be further imposed at the user level. If an alert is frequently triggered for a particular user but not across the general user population, then the penal score should be adjusted and made heavier for that user than for others. Allowing such global (per population) and local (per user) adjustments helps to reduce the scoring contribution from noisy alerts while preserving the scoring contribution from infrequent alerts. This scheme provides analysts with higher-quality results.

A second method that addresses the false positive issue for anomaly detection requires intelligently suppressing some noisy alerts altogether. A good example shows the suppression of the user's first-time asset access alerts. A user's first-time asset access is a common indicator deployed in UEBA. As noteworthy as it may be, such an alert is known to be noisy in practice. Not surprisingly, a user's behaviour on a network is highly dynamic. Project changes, job role changes, or other personal factors can trigger this alert. One way[13] to reduce false positives is through meta learning from the history to build a recommender system that predicts whether the user's first access to the asset could have been predicted. If so, the corresponding alert is suppressed with no scoring contribution.

CONCLUSION

Due to a lack of signatures or known behaviours, insider threat detection cannot be addressed by the conventional means of blacklisting or correlation rules. The statistical and behaviour-based approaches in UEBA offer the most promising solutions for detecting such threats; however, there is no single silver bullet algorithm. It takes multi-pronged, data-centric approaches to find insider threats. This paper introduced four categories where data science applies to insider threat detection. Profile building and statistical analysis provide the backbone framework for behaviour anomaly detection. Machine learning applies to developing contextual intelligence that is otherwise non-existent or difficult to acquire. Targeted detection-use cases with a narrow data scope can benefit from machine learning as well. Finally, system-wide false positives can be controlled by meta learning by analysing the detection system's own historical outputs to improve the system itself. In summary, recent use of data science to analyse existing data available in enterprise SIEM creates new possibilities for threat detection. While many use cases are already meeting success, research in the community will continue to answer the challenges from new or existing use cases in insider threat detection.

Endnotes

1. Trzeciak, R. (November 2017), '5 Best Practices to Prevent Insider Threat', Carnegie Mellon University, available at https://insights.sei.cmu.edu/sei_blog/2017/11/5-best-practices-to-prevent-insider-threat.html (accessed 29th November, 2018).
2. Phillips, T. (April 2018), 'Market Guide for User and Entity Behavior Analytics', Gartner, available at https://www.gartner.com/doc/3872885/market-guide-user-entity-behavior (accessed 29th November, 2018).
3. Chandola, V., Banerjee, A. and Kumar, V. (September 2009), 'Anomaly detection: A survey', *ACM Computing Surveys*, Vol. 41, No. 3, p. 15.
4. Maloof, M. A. and Stephens, G. D. (2007), 'Elicit: A System for Detecting Insiders Who Violate Need-to-Know', in *Recent Advances in Intrusion Detections*, Springer, Berlin, pp. 146–166.
5. Legg, P. A., Buckley, O., Goldsmith, O. and Creese. S. (2015), 'Caught in the act of an insider attack: Detection and assessment of insider threat', *2015 IEEE Internation Symposium Technologies for Homeland Security (HST).*
6. Liu, L., De Vel, O., Han, Q-L., Zhang, J. and Xiang, Y. (February 2018), 'Detecting and preventing cyber insider threats: A survey', *IEEE Communications Surveys & Tutorials,* Vol. 20, No. 2, pp. 1397–1417.
7. Le, A., Markpoulou, A. and Faloutsos, M. (Sepetember 2010), 'Phishdef: Url names say it all', *InINFOCOM 2011 Proceedings IEEE,* pp. 191–195.
8. Woodbridge, J., Anderson, H. S., Ahuja, A. and Grant, D. (November 2016), 'Predicting domain generation algorithms with long short-term memory networks', *arXiv prepring,* Vol. arXiv:1611.00791.

9. Security List Network, available at http://seclist.us/dga-detection-dga-domain-detection-using-bigram-frequency-analysis.html (accessed 29th November, 2018).
10. Kamra, A., Terzi, E. and Bertino, E. (2008), 'Detecting anomalous access patterns in relational databases', *The International Journal on Very Large Data Bases,* Vol. 17, No. 5, pp. 1063–1077.
11. Toffalini, F., Homoliak, I., Harilal, A. and Binder, A. (2018), 'Detection of Masqueraders Based on Graph Partitioning of File System Access Events', *2018 IEEE Security and Privacy Workshops (SPW).*
12. Kudłacik, P., Porwik, P. and Wesołłowski, T. E. (2016), 'Fuzzy approach for intrusion detection based on user's commands', *Soft Computing,* Vol. 20, No. 7, pp. 2705–2719.
13. Tang, B., Hu, Q. and Lin, D. (2017), 'Reducing False Positives of User-to-Entity First-Access Alerts for User Behavior Analytics', *2017 IEEE International Conference on Data Mining Workshops (ICDMW).*