INSIDER THREAT INDICATOR COST MATRIX

Bob Ditmore
Carrie Gardner
Derrick Spooner

February 2021

DOI: 10.1184/R1/13406651

1 Introduction

Insider threat programs look for early warning signs of potential insider threats by applying analytics to various data sources to identify indicators of concerning behavior [1]. The analytics used in these programs vary in capability from simple to complex. On the simple end of the spectrum, the analytic might look for a pattern in the data. On the complex end, the state of some user's behavior is compared against a baseline looking for any abrupt changes. A method to classify these various levels of analytic capabilities would help insider threat program decision makers select and prioritize analytic requirements for detecting and preventing insider threats.

Other research suggests various approaches for classifying analytics [2], [3]. Some suggest categorizing analytics according to detective or predictive qualities versus whether the analytic is traceable or is a "black box." Initially, this categorization seemed like a useful abstraction. We tried to run various types of potential risk indicators from our database, but those attempts raised more questions than they answered.

Others suggest categorizing analytics based on their purpose (e.g., determine activity or determine content, or infer some behavior). However, while this broad approach lends itself well for understanding the types of indicators and their effectiveness against different types of threats, it does not directly address the complexity of the analytics.

These attempts at classifying analytics are usually too broadly scoped because they try to determine what potential indicators are without first understanding a key underlying concept: *the transition of indicators as they move through various stages in the data model*. These attempts to classify analytics have led to confusion about what an analytic refers to, especially as it relates to potential insider threat indicators.

In this report, we explain how data transformation mappings are used to refine which analytics apply to which transform.[1] Using this data transform model, we can refine what is meant by an analytic for insider threat indicators. Then we discuss the dimensions that make up the analytic space.

From those dimensions, we develop a cost matrix that can be used by an insider threat program to prioritize analytic indicator development. We provide examples of how the data transforms and cost matrix help clear up some of the confusion around current insider threat analytic development. Finally, we discuss some future areas of investigation.

2 Data Mapping Transforms

Analytics can be defined as the discovery, interpretation, and communication of meaningful patterns in data [4]. A critical part of an insider threat program is data source selection. The sources of data should include information related to the organization's population and that can be used to look for socio-technical signals of potential insider threat activity. Understanding how personnel in the insider threat hub ranges from data to behavior is similar to the knowledge management's progress from data to knowledge and understanding. Abstracting the predictive adaptive classification model [5], Figure 1 illustrates how data model mapping moves from data input (the left side) to behavioral modeling (the right side).

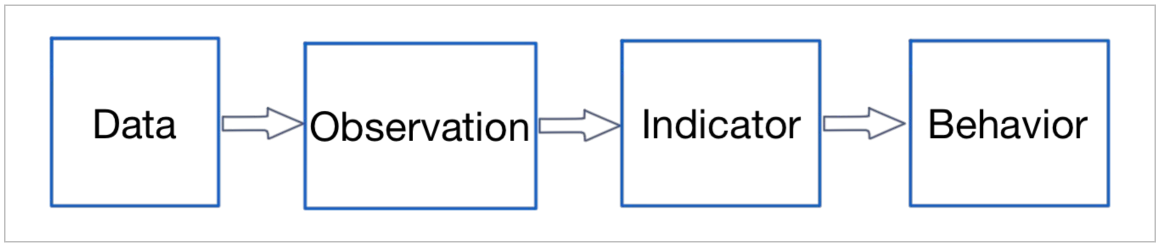


Figure 1: Data Model Mapping Transforms

The data that enters the model can be technical or non-technical. The technical data is the telemetry coming in from technical data sources (e.g., Windows event logs). In the predictive adaptive classification model, information transforms from one stage to the next. Our abstraction of this transformation is represented as arrows between the various stages of the model.

In this report, we present an approach that suggests viewing these stages in the model as *mapping transforms* or, more specifically, *mathematical transform functions*. Viewing the approach in these terms enables us to ask if these mappings are independent of the previous level of the model once computed.

[1] A *transform* is a mathematical transform function, in this case, applying to data model mapping.

The answer is not immediately clear. Looking at various indicator analytics, it appears as though the analyst would need to know the data source that generated the observation that led to the indicator. If this is the case, the insider threat program must keep the indicator's provenance to compute the analytic in the next stage. However, if the analytic is a mathematical mapping transform function, it should be independent of the previous stage once computed. To ground this discussion, we use an analytic related to printing as an example.

In this example, assume the insider threat analyst in the insider threat hub needs two pieces of information for their analysis: (1) the print time and (2) the machine that printed it. In the data stage, the analyst might use a data source such as print server logs, netflow, or output from a data loss prevention (DLP) tool. The analyst can use any of these data sources to extract the two required pieces of information.

However, extracting the information may require different levels of effort. Using the print log and DLP, the information can be readily available. For the netflow case, you might need to know the IP address of the print server or port where the print job occurred. Once the two bits of information are extracted and placed in the insider threat hub, the output from the previous stages can be discarded.

This process lends itself to the mathematical mapping transformation approach. However, this approach breaks down when the analysts need additional information. In this example, they might need the name of the print job. Using a print log or DLP, they can go back to the data sources and get the information. In the case of netflow, the information may not be available without getting addition data from a packet capture.

For more in-depth investigations, analysts may want to see the document. In that case, the packet capture might be able to recover the document, the DLP might be configured to store the document, and the print server might be configured to keep the document available for further review. In this approach, the insider threat analysts want to keep all of the data just in case it is needed. This approach forces data source provenance in the event the analyst needs to gather additional data for the analytic. If the insider threat program is more intentional about its analytic needs, these cases could be made to use an independent mathematical mapping transform approach.[2]

We suggest that an insider threat program treat the steps in the process as mapping transformation functions that provide the program with a more accurate understanding of the fields needed from the data source [6].

Let's apply a bit more formalism to the arrows between the stages that represent the mapping transform functions. From left to right, they can be thought of as the following:

$$O = f(D) + \epsilon$$

$$I = f(O) + \epsilon$$

$$B = f(I) + \epsilon$$

[2] This approach also supports the insider threat program's legal, privacy, and civil liberty requirements by being intentional in the program's request for data.

D is the data source(s), *O* is the observation(s) transformed from the data, *I* is the indicator(s) transformed from observation(s), and *B* is the behavior(s) transformed from the indicator(s).

The epsilon (ϵ) associated with each transform represents the amount of uncertainty. This uncertainty can represent the number of false positives or false negatives introduced by the transform. Overall, it represents the risk associated with the analytics used by the insider threat program that are associated with these data mapping transform functions.

Expanding on each of these transforms, let's look at the first function. The first mapping transform—from the data to the observation stage—can perform one of two major activities:

1. The first activity can be thought of as a traditional extract, transform, and load (ETL) from data management for technical data sources that provide telemetry about the organization's environment and can provide some entity resolution.
2. The other activity, for non-technical data sources, measures and weighs the reliability and credibility of the sources.

The next mapping transform, from the observation to the indicator stage, is the main focus of this report.[3]

The final mapping transform is from the indicator to the behavior stage. This stage models sequences of indicators to determine user behavior.

This model shows that the analytic indicators are transforms associated with the mapping transformation from the observation stage to the indicator stage. Insider threat programs should look at this observation-to-indicator transform when doing their analysis. Unfortunately, in the publications and various programs, there tends to be some confusion about what stage in the pipeline is defined as *analysis*. This confusion leads to a poorly defined list of potential risk indicators. For example, some programs include the observations or behavior transform stages as indicator analytics.

Consider how an insider threat program might use this approach to refine what is meant by a term like *disgruntlement*. A data source for the program might be a tip line. As the tip line data source is processed, a disgruntled user could become an observable. However, with our approach, this observable is made more precise by signifying, "Person A observed Person B being disgruntled." Next the observable could become an indicator after the mapping function has properly vetted the report to verify it. Finally, the observable could be used in a behavioral model, where the observation is combined with other indicators as suggestive evidence of potential insider activity.

[3] We expand on this mapping transform in upcoming sections.

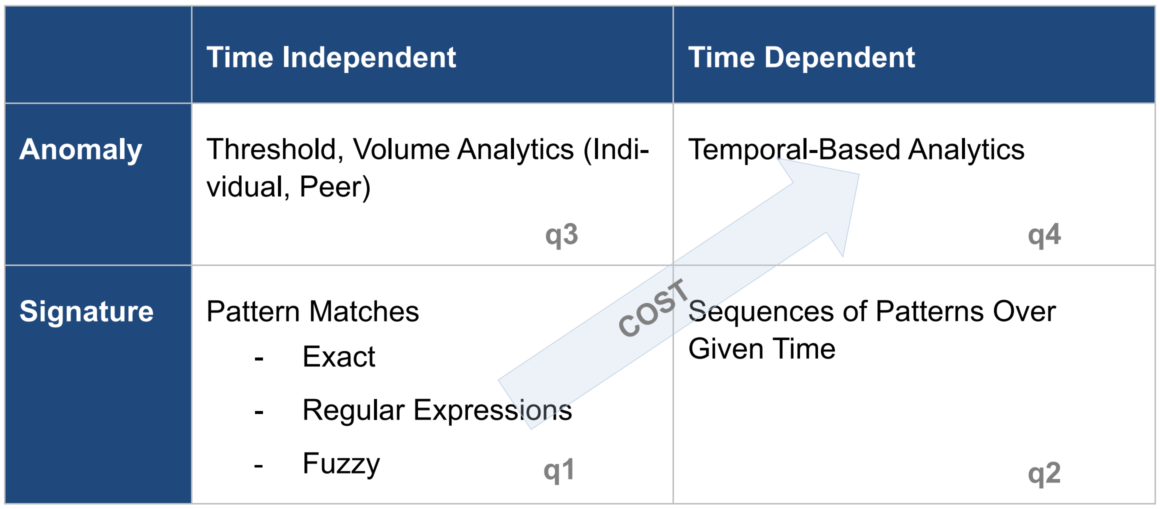
3 Cost Matrix

To operationalize the data classification model, we sought to develop a tool to measure various costs associated with using the different types of analytics (i.e., the cost matrix). The current version of the cost matrix defines direct and indirect operational costs for analytics designed to operate at the observable-to-indictor mapping levels. Future work will build on this matrix and define additional cost factors at all data transformation levels.

Our internal insider threat indicator repository and external, current publications continue to support the historic classification of threat detection approaches as either *Signature* or *Anomaly*.

These threat-detection approaches make up the y-axis (i.e., vertical quadrants) of the cost matrix, which is set against a more novel categorization—time dependency. The x-axis (i.e., horizontal quadrants) denote whether the analytic requires a time-dependent feature. We found evidence to support the hypothesis that time-dependent analytics require significantly more complexity than time-independent ones.

The cost for an analytic in any of the quadrants is defined as the amount of resources (e.g., memory, computation, network traffic, energy) required to execute the analytic. An increase in the need for resources usually corresponds to an increase in the complexity required for the analytic. The cost matrix is built using a quadrant chart with increasing cost going from left to right and bottom to top. Putting it together, the cost matrix is shown in Figure 2.



	Time Independent	Time Dependent
Anomaly	Threshold, Volume Analytics (Individual, Peer) q3	Temporal-Based Analytics q4
Signature	Pattern Matches - Exact - Regular Expressions - Fuzzy q1	Sequences of Patterns Over Given Time q2

Figure 2: Cost Matrix

Analytics in quadrant 1 (q1), in the lower left corner, are a class of the time-independent, signature pattern matches. These analytics can range from simple keyword matches to more sophisticated matches using regular expressions or some other type of fuzzy matching technique (e.g., word stemming). Time independence for this matrix means that the analytic does not explicitly use some form of temporal relationship. For example, the time-independent analytics show “all uses of a given keyword in a data source over the last 24 hours.”

Analytics in quadrant 2 (q2), in the lower right corner, can be used for analytics with an explicit time dependency or that are used to identify a sequence of signatures over a period of time. An example of the first case is designed to identify a user printing after normal business hours. An example of the sequence of signatures could be used to identify when a user downloads a large number of sensitive files from a file share, compresses them on their desktop, and then moves them to a USB. Another example is designed to detect an insider who seek national security information without a need-to-know, where the user repeatedly browses and reads sensitive files outside the scope of their duties, searches for sensitive files, and asks others for access to sensitive documents without a need-to-know.

Analytics in quadrant 3 (q3) describe a class of time-independent anomaly detectors. An anomaly detector is an analytic that expects a particular "normal" behavior and looks for outliers from the behavior. In this quadrant, the anomaly detection is done in an explicitly time-independent manner. An example of this type of analytic is excessive file downloads. Although there is an implicit timeframe associated with this analytic, it is really concerned with what is considered a "normal" volume over any given implicit time frame.

Analytics in quadrant 4 (q4) are the most advanced because they look at the system over time, try to learn normal behavior, and then alert on anomalous behaviors. An example analytic could be designed to detect normal and abnormal changes in a user's activity patterns over time, potentially flagging events that are abnormal for a user or user group at a given time of day or set of proceeding events.

To ground this discussion about cost increases, let's use an example of an insider threat program concerned about the cost of storage associated with various proposed analytics. If the program goes from a time-independent to a time-dependent analytic (q1->q2 or q3->q4), additional storage might be required, holding onto the data until an alert is raised. If this was a streaming analytic, the program should consider viewing the cost associated with storage.[4] In the streaming case for the time-independent signature quadrant, if a pattern is not found, then the unnecessary data can be discarded, thereby saving on storage cost.

In quadrant 2, looking for a sequence of patterns in a stream requires keeping state on what has occurred until you can discard events in the stream once you can determine an event cannot occur (e.g., a timeout or log out has occurred). For example, if you are performing an analytic on the activities within a user session looking for a user who remotely connects to a critical server, elevates privileges, and then changes or deletes a critical file, then you would only need to maintain the state and log of those activities until the timeout (or a log out, in this case) occurs. The insider threat program must determine which costs it is concerned about related to the analytics under consideration. A more interesting example of a cost that an insider threat program might consider is the effort required to get legal approval. For instance, a legal team would likely require less interaction to approve something simple, such as a list of keywords in quadrant 1, versus a less concrete request of maintaining various, yet to be determined, user information for an unspecified amount of time to achieve a quadrant 4 analytic.

[4] Streaming analytics query a continuous flow of data and in near-real time to detect a given condition and raise an alert. As a final part of the stream, the data could be moved to longer term storage for later batch processing.

Since the matrix is divided into four quadrants that represent increasing cost, an insider threat program can label each potential indicator analytic under consideration and use this information to help determine if the analytic is worth pursuing. For example, if a program is relatively new, it may not have the maturity to implement anything greater than a q1 analytic. However, as a program matures and becomes more comfortable with baselining various activities, it can consider implementing more advanced analytics in the higher quadrants.

This cost matrix is used to move from the observation stage to the indicator stage. Once the program clarifies a proposed indicator, it does not exist at this mapping transform stage; instead it should consider using another approach to understand the costs associated with that stage.

4 Future Research

This report focuses on the stages of data transformation and the associated analytic cost matrix for the observation-to-indicator stage. As discussed, there are costs associated with subsequent transforms in the process, especially the indicator-to-behavior stage. Future research is needed to understand the features required to categorize the transformation from indicators to behaviors.

Another area of possible future research is taking a known list of potential risk indicators that major organizations use and categorizing them by applying the approach in this report and, perhaps more importantly, remove obvious non-indicators from the list.

Finally, we briefly discussed the epsilon (ϵ) term as part of the mapping transform functions and how it can be related to the uncertainty inherent in the functions. Future research can quantify this value so that insider risk programs can use it to measure their overall effectiveness; this value could also help organizations understand their risk posture and overall resilience.

5 Bibliography

URLs are valid as of the publication date of this document.

[1] M. C. Theis, R. F. Trzeciak, D. L. Costa, A. P. Moore, S. Miller, T. Cassidy and W. R. Claycomb, "Common Sense Guide to Mitigating Insider Threats, Sixth Edition," February 2019. [Online]. Available: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=540644.

[2] INSA'S INSIDER THREAT SUBCOMMITTEE, "An Assessment of Data Analytics Techniques for Insider Threat Programs," July 2018. [Online]. Available: https://www.insaonline.org/an-assessment-of-data-analytics-techniques-for-insider-threat-programs/.

[3] Software Engineering Institute, "Analytic Approaches to Detect Insider Threats," December 2015. [Online]. Available: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=451065.

[4] Wikipedia, "Analytics," [Online]. Available: https://en.wikipedia.org/wiki/Analytics. [Accessed 17 December 2020].

[5] F. L. Greitzer, "Predictive Adaptive Classification Model for Analysis and Notification: Internal Threat," July 2011. [Online]. Available: https://i4.pnnl.gov/focusareas/class_model_desc.stm.

[6] R. Ditmore, "High-Level Technique for Insider Threat Program's Data Source Selection," Software Engineering Institute, 30 May 2019. [Online]. Available: https://insights.sei.cmu.edu/insider-threat/2019/05/high-level-technique-for-insider-threat-programs-data-source-selection.html.

Legal Markings

Contact Us

Software Engineering Institute
4500 Fifth Avenue, Pittsburgh, PA 15213-2612

Phone: 412/268.5800 | 888.201.4479
Web: www.sei.cmu.edu
Email: info@sei.cmu.edu