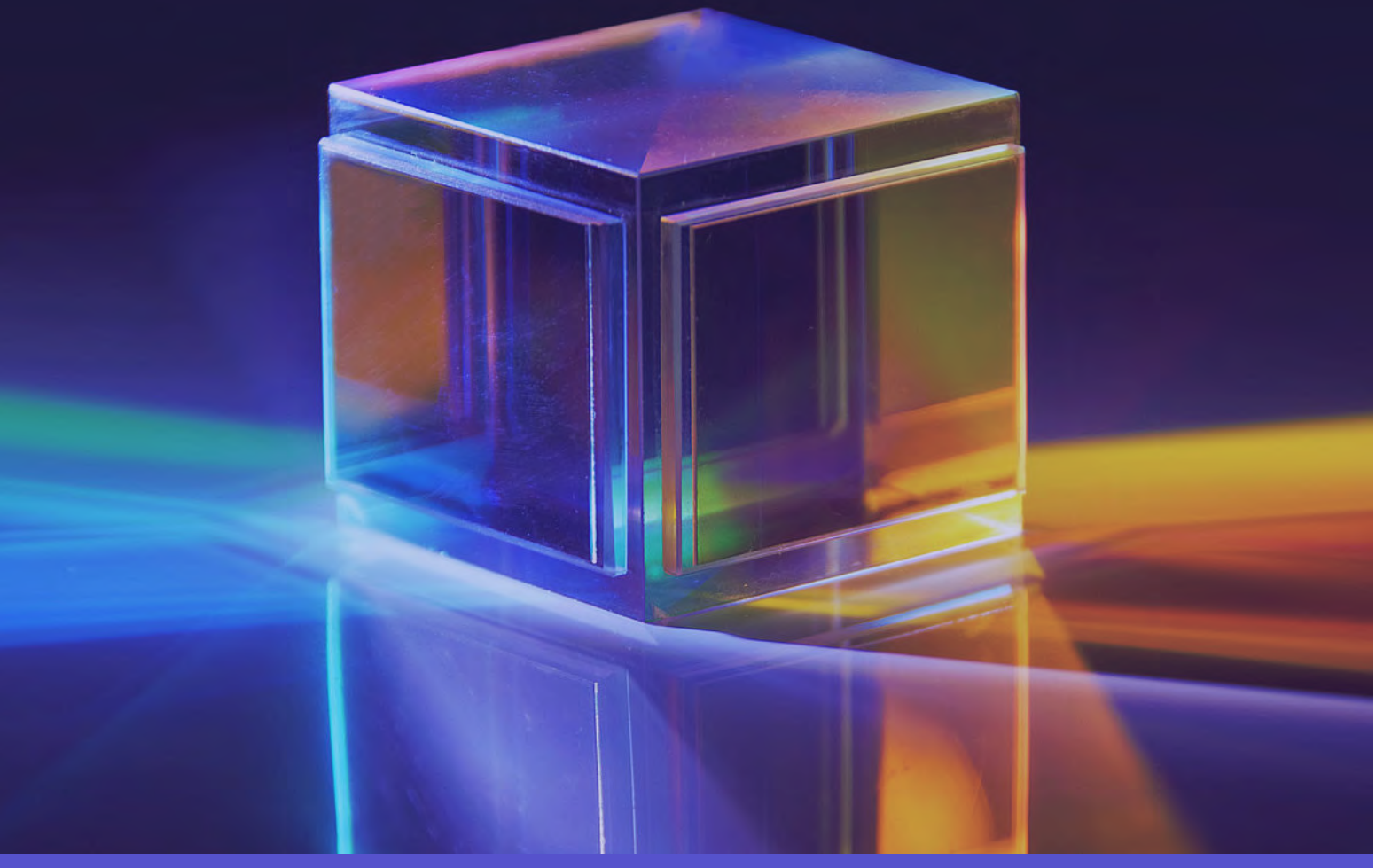
Reveal security

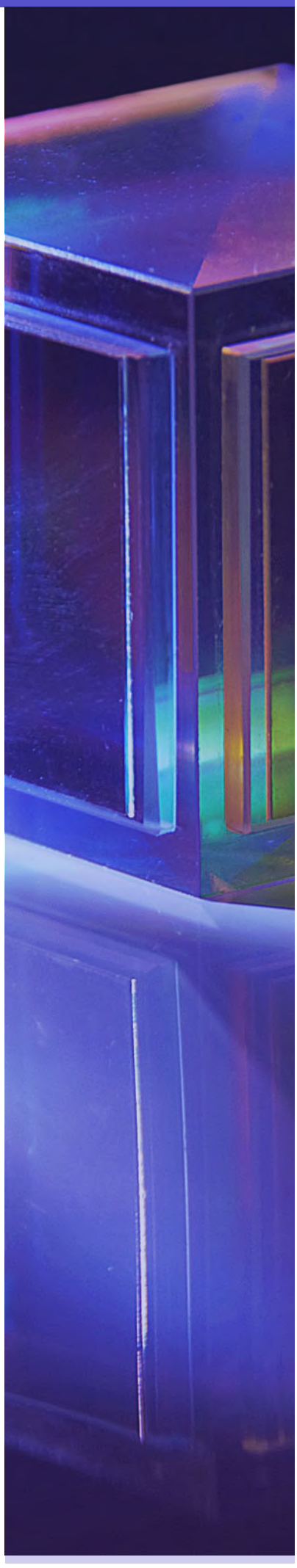
SOLUTION BRIEF

TrackerIQ for Custom-Built Business Applications



Oct 2022

Table of Contents



Why Monitor Application Usage?

Rogue insiders and external attackers have become a growing concern in enterprise business applications.

Rogue insiders include employees and admins who might misuse their access rights in enterprise business applications and engage in malicious activities. These malicious insiders exploit the rare monitoring of business applications, and even when their activities are logged, proactive detection on these logs is scarce.

Consequently, enterprises only discover misuse, abuse and/or malicious activities after complaints from victims. Such examples could include a bank teller skimming cash, or a customer service agent at an insurance company modifying a policy to add themselves as a beneficiary, or a salesperson moving to a competitor who downloads a report of all customers to take to their new employer. Even after the enterprise receives a complaint or is otherwise suspicious, detection of these breaches usually consists of manual sifting through tons of log data from multiple sources.

External attackers leverage the increased attack surface resulting from post-COVID-19 business realities, whereby many business applications are now exposed to the internet so that employees can work remotely. This makes it easier for attackers to use stolen credentials to impersonate an insider and perform malicious activities within the enterprise's business applications.

The combination of rogue insiders and external attackers makes application detection a massive pain point for enterprises, particularly within their core business applications. Current detection solutions, which are mainly based on rules, are application-specific and in most cases ineffective. Thus, a new approach is required.

Rules are Ineffective in Detecting Application Layer Malicious Activities

Detecting malicious activities in business applications is mainly performed today using rules that are defined by a security/business analyst, within a rule engine that is usually part of a SIEM or a log repository (e.g. Splunk, Snowflake, etc.). These rules define illegal activities that should not be performed by a user in the application. They are applied to logs generated by the business application and stored in a central repository or a SIEM solution.

The use of rules to detect malicious activities suffers from the following drawbacks:

Writing rules requires understanding the application's business logic

First, in order to define rules, one must be very familiar with the entire application business logic, and know what should be considered forbidden in any given scenario. This knowledge rarely exists in a typical enterprise, and because each application is different, the security/business analyst must be familiar with each application's business logic. Without such detailed knowledge, rules written will not be accurate, therefore generating many false alerts.

Rules also require continuous maintenance

Application developers may change the application business logic in any version of the application. Knowledge about these changes and their effect on forbidden activities is rarely transferred to the security/business analyst who maintains the rules for application detection. Many false alerts are generated when rules aren't up to date with changes made in the application logic. However, more importantly, new types of malicious activities may not even be alerted (false negatives).

In business applications, what is forbidden for one user may be legitimate for another. Thus, rules must be defined per user role, which makes rule creation and maintenance even more difficult, and in most cases renders rules-based detection impractical.

Plus, rules suffer from significant false negatives…

Rule-based detection detects only known malicious activities with well-defined patterns. But in the actual day to day, a security/business analyst (even one fully familiar with the application) still cannot foresee all possible attack patterns (i.e. the combination of legitimate activities that can cause damage). Attackers, on the other hand, are constantly identifying and leveraging loopholes in application business logic to achieve malicious goals, thus leading to many false negatives.

And finally, rules don't scale!

Last but not least, even if one succeeds in creating perfect rule-based detection for an application, one would still have to start from scratch to apply rule-based detection to another business application, as each application is different, with its own business logic.

Consequently, rule-based detection solutions are notoriously problematic. They generate numerous false positives and false negatives, and don't scale across the ever-increasing number of business applications used by the enterprise (typically ranging from tens to hundreds).

UEBA is Ineffective for Application Detection

Over a decade ago, the security market adopted statistical analysis to augment rule-based solutions in an attempt to provide more accurate detection for the infrastructure and access layers. Statistical analysis has therefore been in use to detect suspicious access to applications by imposters. However, User and Entity Behavior Analytics (UEBA) has rarely been implemented for detecting malicious activities in business applications (especially by malicious insiders).

First, to implement UEBA, we must define the quantities on which we will perform the volumetric / statistical analysis. Defining these measures requires familiarity with the application's business logic, and as explained above most security/business analysts don't have this knowledge.

In addition, because application business logic changes, the UEBA configuration should change accordingly, to match the business logic. A UEBA system that isn't constantly updating configuration to reflect respective changing business logic, will very quickly become less relevant and generate many false positives and negatives.

But the most important reason that UEBA has not been implemented for detecting malicious activities at the application layer is that it has failed to deliver a promised dramatic increase in accuracy and reduced false positive alerts. This is due to a fundamentally mistaken assumption that UEBA is based on - that user behavior can be characterized by statistical quantities, such as the amount of money transferred in a transaction, or the number of stock buy/sell transactions performed daily, etc. In reality though, people don't exhibit "average behaviors," and it is thus futile to try and characterize human behavior with quantities such as "average," "standard deviation," or "median" of a single activity.

As an example of non-average behavior:

Meet David, a personal banking account manager at a major bank. As part of his normal daily activities, David has a variety of different professional working profiles:

a. He may be called by a customer to perform a bank transfer on their behalf, either externally, between branches, or between accounts at the same branch.

b. At other times, he may assist a customer with the buying and selling of various stocks.

c. On a monthly basis, David generates a status report of all customers he is responsible for and email it to his manager.

Computing an average of the daily activities in David's workday would be meaningless. We should focus instead on learning David's multiple typical activity profiles.

AI/ML Models are Application Specific and Hard to Tune

A third approach for detecting malicious activities in business applications has been applied for specific types of transactions, like credit card transactions. In this approach, an AI/ML model is built and tuned for a specific type of transaction. Such models can provide accurate detection, but they must be built and tuned for every type of transaction. As a result, these models have been applied (with limited success) to a very small set of transactions, like credit card transactions and money transfer transactions between banks.

Note as well that to tune these models, labeled training data is usually a mandatory requirement. However, such labeled data does not exist in most cases of business activities/transactions, and a model that cannot be tuned is unusable.

Consequently, although AI/ML models have been applied with limited success (mainly for credit card and money transfer transactions which have a standard data structure and training data), they are not a practical solution for most business applications.

TrackerIQ

Application Layer **User Journey Analytics**

To accurately detect malicious activities performed by authenticated users in a business application, TrackerIQ has adopted the concept of User Journey Analytics. According to this concept (based on Cisco's NetFlow for accurate detection in the network layer), one must analyze Sequences of Activities to contextualize each activity and achieve accurate detection.

At the application layer, the sequence of activities performed by a user describes the user journey in an application session. TrackerIQ's detection is based on analyzing these journeys and identifying anomalous journeys that can indicate suspicious/malicious journeys in the application.

Analysis of user journeys accurately detects imposters, as it is very difficult to imitate a user's normal journey in an application. It also accurately detects insiders looking to misuse or abuse an application, as they would then deviate from their normal user journey profiles.

The accurate detection of malicious behavior via analysis of user journeys is based on the underlying assumption that an abnormal session is characterized by a journey which isn't similar to the user's typical journeys in an application. Thus, we can accurately detect abnormal journeys, which are highly correlated to malicious activities by learning typical journeys and creating normative journey profiles.

It is important to emphasize that while User Behavior Analytics (UEBA) is about a single baseline for each activity and an analysis of each activity on its own, User Journey Analytics looks at sequences of activities and learns for each user the complete set of typical user journeys in an application.

TrackerIQ's Ubiquitous Application Detection Model

Each application has a bespoke set of activities and log formats. Thus, to be able to apply user journey analytics to any application, our detection model must be agnostic to the meaning of the application's activities and to the application's log record formats.

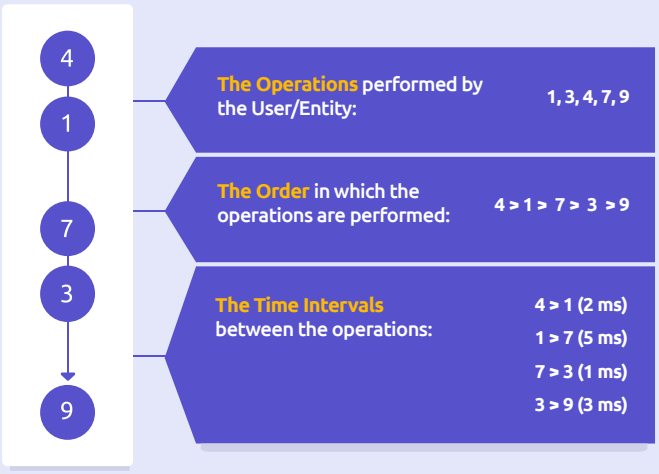
To accomplish this, as TrackerIQ analyzes the user journey in an application session, it extracts the following sequence characteristics which are available in any sequence of log events, regardless of the actual meaning of the application's activities:

(a) The set of activities performed during a session

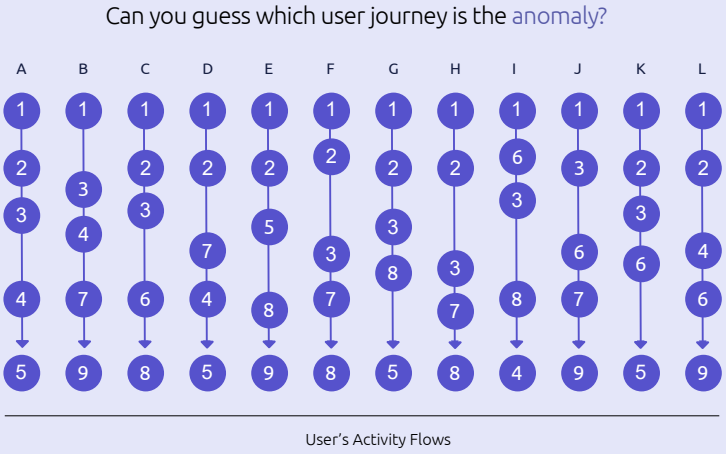
(b) The order in which these activities were performed in the session

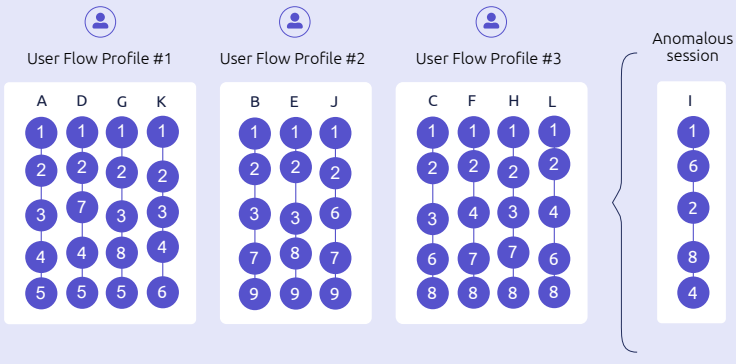
(c) The time Intervals between activities during a session

These three characteristics are agnostic to the meaning of activities and can therefore be applied to any application session, and even to sessions across applications. Thus, TrackerIQ can be used for detection in all business applications (custom-built and SaaS) used by the enterprise, eliminating the need to use bespoke detection solutions for different applications.



These three parameters: the set of activities, their order, and the time intervals between them can be applied to any application, ensuring TrackerIQ is ubiquitous.





TrackerIQ detects anomalous user journeys which deviate from the user's profiles

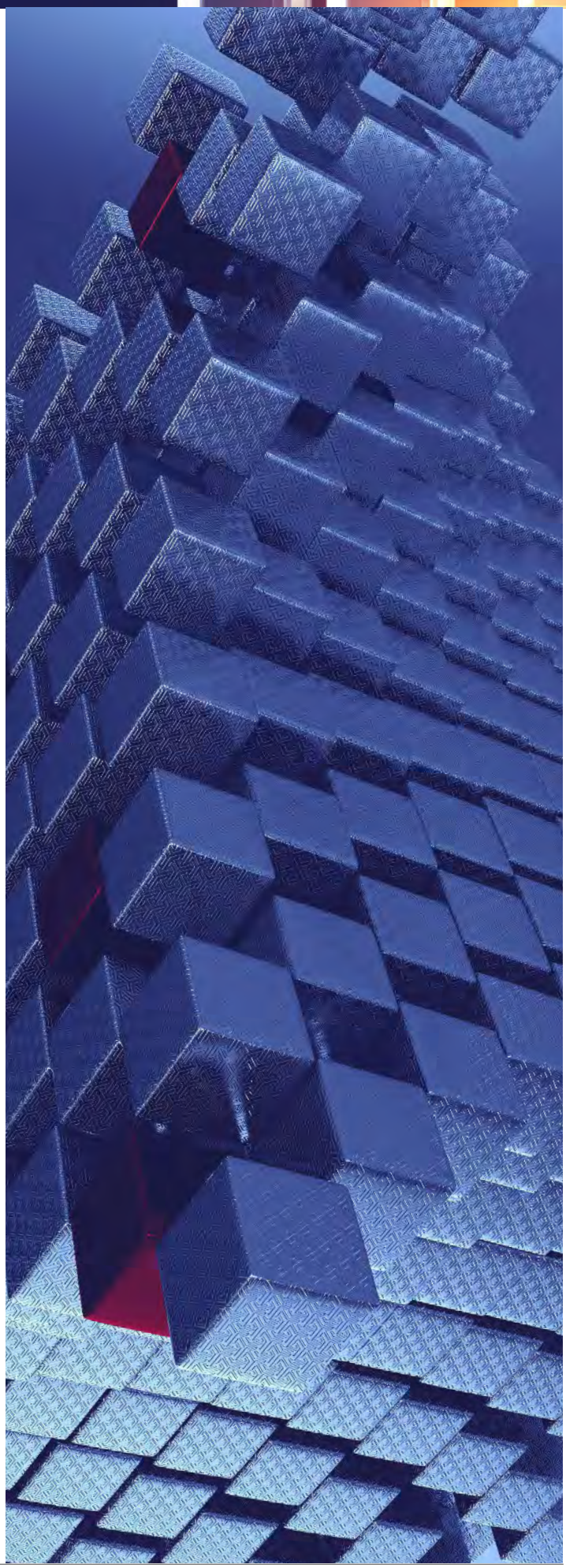
TrackerIQ ML Learns Multiple Typical User Journeys

As explained above, users don't have a single journey, or an "average" journey. Each user has many typical activity journeys in each application, in addition to multiple journeys across applications. Thus, to find anomalies, an accurate detection solution must be able to automatically learn all these multiple typical journeys. TrackerIQ learns all common user journeys for accurate detection. It often learns many user journeys, especially when TrackerIQ is analyzing the journeys of a cohort of users.

To learn these user journeys, TrackerIQ applies its patent pending clustering technology to group similar sessions together and then build a typical user journey from each such cohort of sessions. This is a process that runs continuously as new log data is available.

Once typical journey profiles have been learned for a user, TrackerIQ compares every new session to see if it is similar to one of the typical user journeys learned for this user. An anomaly is detected when the current user journey is not similar to any user journey profiles learned for the user.

To detect scenarios in which users behave differently than their peer group, TrackerIQ also compares a user journey against typical user journeys learned for the cohort of users to which the user belongs.



Ranking Anomalies

Information security professionals look for anomalies which impact their business, so not every anomaly is "interesting." Anomalies that contain sensitive activities from a business perspective are of course more interesting. Ranking anomalies based on the sensitivity of their activities enables TrackerIQ to alert security/business analysts only about anomalies they care about.

TrackerIQ enables the enterprise to provide a set of sensitive activities for its applications (with sensitivity expressed on a scale from 0 to 10). This information is used to compute a sensitivity score for each user journey (i.e. session). The final risk score calculated for an anomalous user journey combines the user journey anomaly score with its sensitivity score. Thus, anomalies with high sensitivity are ranked higher than anomalies with low sensitivity. This enables analysts to focus only on anomalies that are meaningful from a business perspective.

Anomaly Investigation

Application logs usually consist of codes which are rarely "human readable." To help the business analyst - who is usually not familiar with these codes - analyze an anomalous journey, TrackerIQ "translates" these codes, presenting information in a human readable format and enabling quick and proactive investigation. This translation is done according to a simple text translation file uploaded by the enterprise (as part of TrackerIQ's initial configuration).

Furthermore, TrackerIQ provides analysts with a set of tools and detailed information for the investigation of user journeys. This helps the analyst quickly decide whether or not a further deep dive is required for each of the anomalies detected.

Supported Log Repositories

TrackerIQ detection is based on analyzing log data generated by the business application. Log data records generated by the application require that each log record include at least the following 3 fields: the time of the activity, the user (or entity) which performs the activity, and the activity code/description. When additional data is included in the log record, TrackerIQ may use it to improve the action definition and enhance accuracy.

TrackerIQ can read application log records from any log repository such as Splunk, SQL-database, Kafka Bus, etc. TrackerIQ doesn't replicate log records, it only reads and analyzes them to learn the user journeys, and then generates alerts when sensitive anomalies are detected.

Value Proposition

RevealSecurity detects abuse, misuse and malice at the application layer, from insiders as well as imposters. TrackerIQ's unique differentiators provide a strong value proposition:

High Accuracy

Highly accurate with a low signal-to-noise ratio

User Journey Analytics

Analyzes user journeys, as opposed to UEBA which is based on analyzing individual activities.

Analysis Per Application

Detects anomalies by learning every user's multiple typical user journey profiles per application, as opposed to UEBA which characterizes normal behavior by a single average per activity.

Ubiquitous Modul

Based on a ubiquitous user journey model which can be applied to any business application as well as across applications.

No Rules

No need to define rules!

For more information reach out to us at www.reveal.security

About Reveal Security

RevealSecurity monitors privileged users, malicious insiders and impostors to detect anomalies in applications and platforms. Time and again, reputable research has found that the longer it takes to detect a breach, the greater its damage, yet most detection of breaches within applications is still rule-based, thereby costly and ineffective due to a debilitating high rate of false alerts. Meticulous authentication is never enough, as users who have legitimate application access are still involved in misuse, abuse and malice. RevealSecurity champions ubiquity and accuracy in the application detection market.

TAG CYBER 2022 DISTINGUISHED VENDOR

Visit us at
www.reveal.security

General inquiries
info@reveal.security

Media
media@reveal.security