ManageEngine
Log360

How to improve **risk scoring and threat detection** with UEBA

By Hiranmayi Krishnan

Learn how anomaly detection works

Discover how UEBA determines the risk score

Explore ways to improve risk scoring accuracy

Table of Contents

Introduction

Cyberattacks are a growing menace with a new hacking attempt launched every 39 seconds.[1] The sophistication of attacks is also increasing, with cybercriminals developing new techniques of attack and evolving older ones. This is evidenced by 6.3 trillion intrusion attempts and 5.5 billion malware attacks that the world witnessed in 2022.[2] At the same time, organizations also have to worry about internal threats that are increasing day by day. Globally, more than 34% of businesses face insider threats yearly, and in the years 2021-2022, it increased by 44%.[3]

In such a grave climate, organizations must be properly equipped to detect, prioritize, and respond to threats quickly. But, in 2022, the mean time taken to detect (MTTD) a breach was 207 days, and the mean time to contain (MTTC) the breach was 70 days.[4]

A *security information and event management* (SIEM) solution integrated with *user and entity behavior analytics* (UEBA) capabilities helps bring your MTTD and MTTC down.

A SIEM solution collects and aggregates log data from various sources in your organization's network, and analyzes the data to detect threats. It also alerts you to those threats in real time. A SIEM uses predefined and custom correlation rules, response workflows, and threat intelligence feeds to accomplish this. In addition to these capabilities, SIEMs of today also have the UEBA functionality. With UEBA capabilities, a SIEM, over time, identifies and tracks risky users and assets across your network based on their anomalous behavior. This helps you detect threats before they occur. You'll also be able to prioritize threats based on risk score, and deal with them accordingly.

In this e-book, we break down how the risky behavior of users and entities is unraveled with UEBA, a precise, analytics-based anomaly detection capability.

This e-book addresses:

- How anomaly detection works.
- How techniques, such as peer group analysis, seasonality, and user identity mapping improve risk scoring accuracy.
- How ManageEngine Log360 helps you detect anomalies and mitigate cyberattacks.

What is UEBA?

UEBA is a cybersecurity mechanism within a SIEM that monitors and analyzes the behavior of every user and entity, such as routers, servers, and endpoints in an organization's network, to detect anomalies (see Figure 1). UEBA first creates a baseline of expected activity for every user and entity using machine learning (ML) algorithms. Any deviation from this baseline is considered an anomaly. However, for establishing this baseline accurately, you need to provide your SIEM with at least two weeks of historical data.

Figure 1: UEBA algorithm monitoring user's behavior to detect anomalies

Every current action is compared against the behavioral baseline to identify whether the action is normal. Depending on the extent of the deviation, your SIEM increases the risk score of the user or entity by a suitable amount. It also alerts your security analysts.

What are the benefits of UEBA?

Apart from derailing insider attacks, UEBA helps you:

- Reduce the number of false positive alerts.
- Detect and alert you to malicious lateral movements and privilege escalation attempts of an attacker. In this way, it also helps with tackling advanced persistent threats.
- Thwart zero day attacks for which there are no known signatures.[5]

How does anomaly detection work?

To understand how UEBA creates a behavioral profile for every user, let's go over an example and understand how humans do it first. John is a newly hired marketing intern. On his first day of work, the security guard recognizes him as someone new and pays close attention to ensure that all his credentials check out. The guard also tracks the time John enters and exits the facility. He monitors John's activity for a few days and gets to know his expected time pattern—arrival at 10am and exit at 6pm. Any deviation from this, such as John's arrival at 5am, will raise the guard's suspicion. This is how humans detect an anomaly.

Similarly, the ML algorithm in a UEBA-integrated SIEM solution will monitor the log data to establish patterns in your network. For instance, a user's logon and logoff times and the actions the user performs on particular devices will tell the SIEM solution of the activities that are expected from that user. Once it monitors activities for a few days, the UEBA engine will know the user's expected behavior, including any deviation from it. The user's risk score will increase to indicate the severity of the threat, and the SIEM solution will flag an alert to the security analysts. **"But if a human can already do that, why do you need UEBA?"** Because it is not humanly possible for your security team to constantly observe and analyze the behavior of the thousands of employees who work at your organization; generate reports on anomalous activities at different parts of the network; and take appropriate action immediately.

What type of anomalies can the UEBA solution identify?

UEBA identifies anomalies associated with time, count, and pattern. Let's take a look at what each of these means.

Time anomaly: If a user or entity deviates from the expected baseline, it is termed a time anomaly. You can take John's login at 5am instead of his usual 10am as an example of a time anomaly.

Count anomaly: If a user or entity performs an abnormal number of activities within a short span of time, we call it a count anomaly. An example would be a user accessing a database with customer data 50 times in an hour.

Pattern anomaly: If an unexpected sequence of events results in a user account or entity being accessed in an atypical or unauthorized manner, it is termed a pattern anomaly. For example, a user account having a successful logon after eight consecutive login failures, followed by multiple file deletions, modifications, and data transfers made from that account, is an example of a pattern anomaly.

Now that we know what type of anomalies a UEBA identifies, let's dig deeper and see how it accomplishes this in the first place.

How does UEBA establish the baseline to detect anomalies?

UEBA employs statistical models such as robust principal component analysis (RPCA) and Markov chains to establish the behavioral baseline. These anomaly detection models help detect time, count, and pattern anomalies.

RPCA

This method is a variation of the popular technique, principal component analysis. Here, the ML algorithms look at the historical data to determine the line of best fit as shown in Figure 2. The observed events are considered as the matrix summation of expected and outlier events. The outliers that deviate from the predicted line of best fit are considered to be anomalous.[6] The RPCA method is used for identifying time and count anomalies.

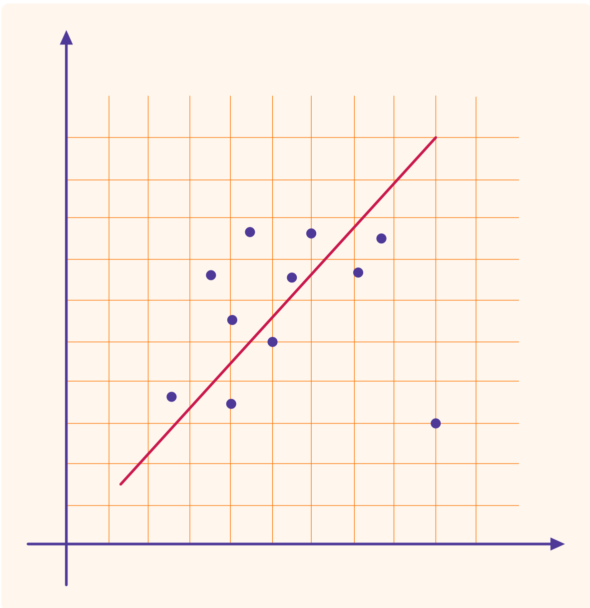

Figure 2: RPCA algorithm determining the line of best fit

Markov chains

A Markov chain is characterized by a sequence of events in which the probability of the next event is entirely dependent on the state of the current event. Here, the algorithm compares each action of the user or the host against a list of possible actions, and over time, identifies the event with low probability as an anomaly. You can determine pattern anomalies using this method.

In Markov chains, the pattern of interest is divided into two consecutive actions, and the algorithm checks to see if the probability of the second action occurring after the first is good. The algorithm relies on historical behavior to determine this probability.

For example, John has performed a software installation at an unusual time. In this case, the pattern to be analyzed is: **Username > Host name > Time.**

To implement Markov chains, you'll need to split the pattern into two parts:

Part 1: Username > Host name

Part 2: Host name > Time

The algorithm first checks if the probability of the user (John) accessing the host (server) is good, and then checks if it's acceptable for the host to be accessed at that particular time (see Figure 3). If the algorithm identifies either of these actions as unexpected, then this pattern is deemed anomalous.[7]

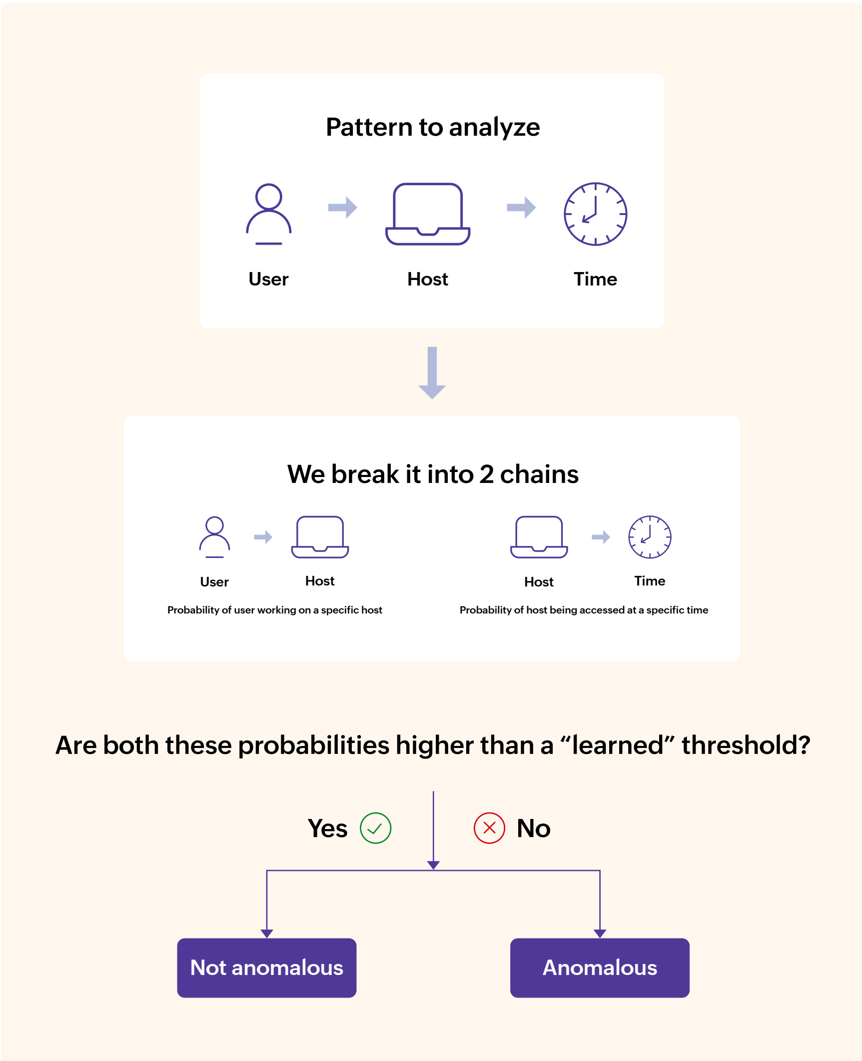

Figure 3: Detecting pattern anomalies using Markov chains

Improving anomaly detection with anomaly modeling

Every organization is different. What's normal activity for an organization maybe anomalous to another. That's why your SIEM solution should provide a custom anomaly modeling feature. Anomaly modeling is the process you can use to build custom anomaly models based on your requirements. Based on the parameters you chose, the algorithm will analyze behavior and establish the baseline. Anomaly modeling can be customized for time, count, and pattern anomalies.

Anomaly modeling for time anomalies

Opting for this enables you to choose the time interval in which the algorithm studies user behavior. For instance, if you want the algorithm to monitor logons at every one-hour interval instead of the default 15-minute interval, then you can do so with anomaly modeling.

Consider John who logs on between 10am and 11am with the default model considering John's logon any time after 10:15am as an anomaly. Now, with anomaly modeling, you can change the interval to 1 hour as per your requirement. Table 1 below shows the difference between what's considered anomalous or normal at a 15-minute interval and an one-hour interval.

Default anomaly model (tracks user behavior in 15-minute intervals by default)			**Custom anomaly model** (tracks user behavior in 1-hour intervals as set by you)		
Date	**John's log on time**	**Anomaly**	**Date**	**John's log on time**	**Anomaly**
June 1	10:02	No	June 1	10:02	No
June 2	10:10	No	June 2	10:10	No
June 3	10:16	Yes	June 3	10:16	No
June 4	10:30	Yes	June 4	10:30	No
June 5	10:12	No	June 5	10:12	No
June 6	11:05	Yes	June 6	11:05	Yes

Table 1: Comparing default and custom anomaly modeling for tracking time anomalies

Anomaly modeling for count anomalies

By creating anomaly models for count anomalies, you can customize the time interval at which you want to track the count anomaly. For instance, you can track file accesses every one-hour instead of the default 30-minute interval. Consider an example of the default model where the expected file accesses made in a specific 30-minute interval is 100, and the actual number of accesses observed is 200. This will be considered an anomaly. However, the analyst would know that it's normal for 200 accesses to be made in that specific half-hour interval. A UEBA solution that provides custom anomaly modeling would enable that analyst to build a model that looks for count anomalies in one-hour interval instead. Table 2 below shows the difference between what's considered anomalous or normal at a 15-minute interval and an one-hour interval.[8]

Default anomaly model (tracks file accesses every 30-minutes by default)			**Custom anomaly model** (tracks file accesses every hour as defined by the analyst)		
Time period	**Number of file accesses**	**Anomaly**	**Time period**	**Number of file accesses**	**Anomaly**
09:00 - 09:30	100	No	09:00 - 10:00	250	No
09:30 - 10:00	150	Yes	10:00 - 11:00	375	No
10:00 - 10:30	200	Yes	11:00 - 12:00	400	No
10:30 - 11:00	175	Yes	12:00 - 13:00	550	Yes

Table 2: Comparing default and custom anomaly modeling for tracking count anomalies

As observed in the table above, in the default model, your UEBA solution will tag any file accesses which exceeds 100 as an anomaly, whereas in the custom model, the algorithm will only consider file accesses exceeding 400 times in an hour as anomalous.

Anomaly modeling for pattern anomalies

Building unique patterns for your algorithm to study and analyze is a useful feature to have, as this will give you the freedom to be independent of the predefined models, or tweak them to suit your requirements.

For example, if the predefined pattern for an attack looks only at username, hostname, and time, but you want the algorithm to look at the domain, and logon type as well, then you can build your own model to accomplish that. Again, the Markov chains technique will be implemented here for analyzing the pattern.

Predefined pattern: Username > Hostname > Time

Custom pattern: Username > Logon type > Domain > Hostname > Time

With custom patterns, you can also change the order of the predefined pattern chain. For example, Username > Time > Hostname. Building custom models can help you accommodate your organization's security situation effectively.

Now that you've mastered how anomaly detection techniques establish baseline and identify anomalies, and how you can customize anomaly model to work in your favor, let's learn how risk scoring works in UEBA.

How does risk scoring in anomaly detection work?

The risk appetite of every organization is different, because they all differ in what they consider to be risky and how much risk they're willing to accept—or consider acceptable—to achieve their goals. Determining the acceptable level of risk also depends on factors such as:

- The size and nature of their business.
- The economy.
- The industry they're in.
- Their competitors.
- Their pricing and revenue model.

Apart from this, organizations also have to consider the risk posed by users and entities to arrive at their risk score. But how do they do it?

As discussed earlier, powered by ML algorithms, UEBA uses historical data to establish a baseline of normal behavior for every user and entity. It then monitors the users and entities in real time to determine if they follow the same pattern of behavior or if they deviate from it, and assigns a risk score. But what is a risk score and how exactly does UEBA arrive at this risk score?

What is a risk score?

A risk score is a value between zero and 100 that is assigned to each user and entity depending on the frequency and severity of deviations from the established baseline. The greater the deviation, the greater the risk. The deviations or anomalies can be a time anomaly, count anomaly, or pattern anomaly.

How does UEBA arrive at a risk score for specific users or hosts?

Apart from looking at the level of deviation, the ML algorithm is usually also programmed to look for the following criteria to assign the risk score:

The role of the employee and the privileges it provides: For example, a C-suite executive would have higher privileges and access to sensitive data compared to entry-level executives that, conceivably, pose a higher risk to the organization. An account compromise or a risky activity carried out from this account would have catastrophic consequences.

The employee's resignation status: If an employee has submitted their resignation and given two week's notice, they pose a greater risk. For instance, an opportunistic employee who's leaving the company to join a rival might be tempted to exfiltrate sensitive client information or other confidential business information.

The data stored on the server or system: A system or a server that stores sensitive data, such as leads and customer information, revenue details, employee PII, product roadmaps, designs, and copyright data, poses a greater risk, so unusual activity on business-critical systems or servers would result in a greater risk score than the rest.

The nature of the account: A service account or a non-human account used by critical applications or services would have higher privileges. This is because it requires these privileges to execute batch files and interact with operating systems and applications hosted across databases, devices, and file servers. These accounts, if compromised, pose a severe risk to the business, and will result in a greater risk score if any anomalous activity is detected.

In addition to these, UEBA also looks at the type of risk to determine the right risk score.

What are the different types of risks that UEBA identifies?

The anomaly detection algorithm looks at four different types of risks to assign a risk score. These are:

1

Insider threats: Any threat to the organization's data posed by an individual inside the organization is known as an insider threat. It can be malicious, where the employee is deliberately trying to steal, modify, or corrupt the data; or it could be unintentional, where the user's account was used to steal sensitive information from the company. Some common indicators of insider threats include a new or unusual system or file accessed at an unusual time, or multiple authentication failures.

2

Account compromise: When a particular user's account is accessed by an unauthorized user, it is termed account compromise. This can occur when a user's password is weak or when an attacker uses sophisticated tools to decipher the user's password. Continuous login failures, followed by unknown software downloads and installation, are a sign of account compromise.

3

Suspicious logons: Any attack, irrespective of its origin as internal or external, will need to have a successful logon at some stage. In the case of an external threat, a successful logon will probably be preceded by multiple logon failures. So, you could say that an anomalous logon is the first sign of an attack. Your SIEM solution should be capable of alerting on anomalous logon successes as well as failures to make sense of the bigger picture. For instance, a successful logon after multiple failed logon attempts could be indicative of a brute-force attack. An abnormal logon on a server or database could signify an impending threat or attack.

4

Data exfiltration: If an individual is making an unauthorized transfer of data to any user or entity outside the organization, it is called data exfiltration. It is a clear sign of an attack, so the risk score of the user rises exponentially. Your UEBA will assign a high risk score and alert the analysts to take immediate action to prevent a data leak. Some signs of data exfiltration are an unusual number of file downloads or data transfers via removable USB devices.

In all the above cases, irrespective of whether the *user* or *employee* attacks the system or network, or whether the *attacker uses that employee's credentials* to attack, that user's risk score will increase. The increase in the risk score is how your UEBA solution will alert the analyst of an anomaly.

How does UEBA calculate the overall risk score?

After the algorithm has computed risks based on the different sub-risks mentioned above, it will calculate the overall risk score of the user (see Figure 4), taking into account the weight (or weightage) and decay factor assigned to each of the risk types. Here, weightage refers to the importance assigned to that specific sub-risk, and decay factor refers to how soon the risk score will return to normal over time in the event of no further anomalies detected.[9]

The overall risk score for users and entities can be calculated using the below formula:

$$\textit{Overall risk score} = w_1 \times DF_1 \times \textit{Risk of insider threats} + w_2 \times DF_2 \times \textit{Risk of account compromise} + w_3 \times DF_3 \times \textit{Risk of suspicious logons} + w_4 \times DF_4 \times \textit{Risk of data exfiltration}$$

Where:

w_1 = Weightage assigned to insider threats

w_2 = Weightage assigned to account compromise

w_3 = Weightage assigned to suspicious logons

w_4 = Weightage assigned to data exfiltration

DF_1 = Time decay factor assigned to insider threats

DF_2 = Time decay factor assigned to account compromise

DF_3 = Time decay factor assigned to suspicious logons

DF_4 = Time decay factor assigned to data exfiltration

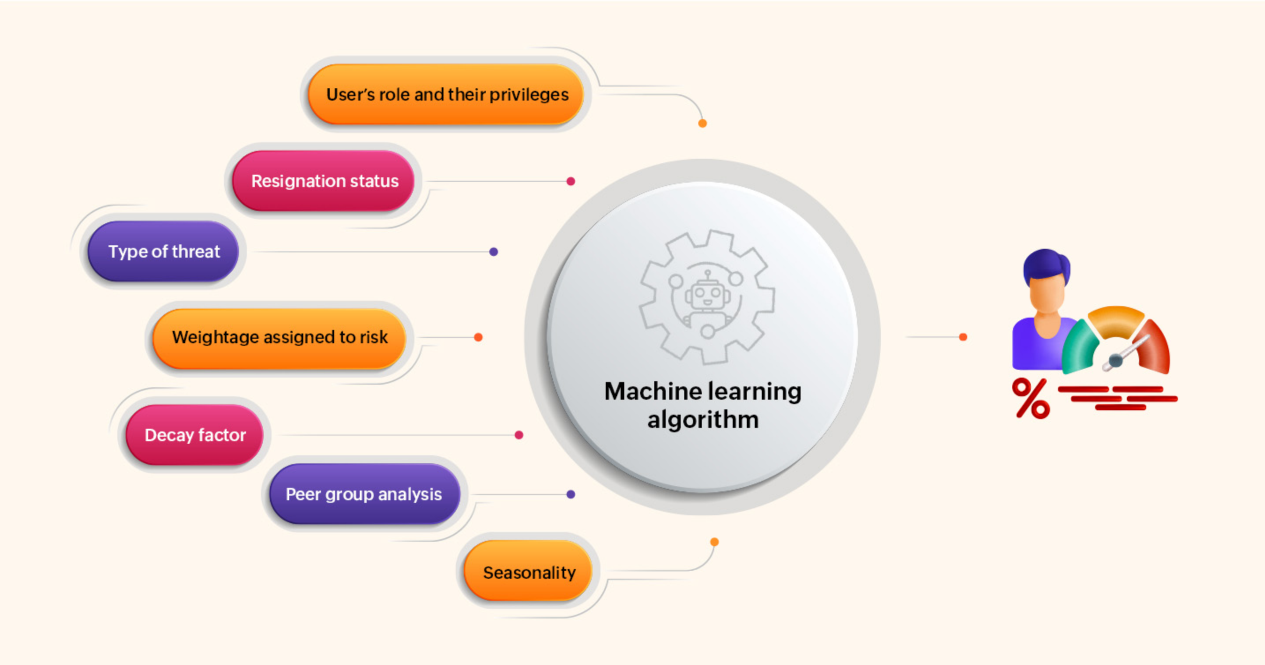

Figure 4: Factors influencing risk score computation

Now you know how UEBA calculates the risk score of users and entities. But is it possible to increase its risk scoring accuracy further? If so, how?

How to increase the risk scoring accuracy in anomaly detection

It's indisputable that the risk scoring aspect of UEBA makes it easier for security analysts to prioritize which threat needs to be mitigated first. This means that the more accurate your risk scoring, the less chance of false positives, and even lesser the chance of a successful attack. The risk scoring accuracy of UEBA can be improved if we factor in peer group analysis, seasonality, and user identity mapping (UIM) while calculating a user's risk score. Let's understand each of these in detail.

Peer group analysis

Peer group analysis is a technique powered by ML algorithms, where statistical models are employed to identify users and hosts that share similar characteristics and categorize them as one group. The idea behind peer grouping is that, by identifying the context behind a user's behavior and comparing it with the behavior of a relevant peer group, the risk scoring efficiency and accuracy will increase. Essentially, if the pattern of your deviation is similar to that of your peer group's, then your risk score will not be negatively affected. However, if your actions don't fit the expected behavior of any relevant peer groups, it'll be considered anomalous and your risk score will increase significantly.

If there is no historical data for a peer group showing anomalous data, then a new group is created and you will be the first member in it. The risk score of the first member in a new group is going to have a much higher score initially when compared to the rest. If this action is performed by other members, then it becomes a trendsetter rather than an outlier, and your risk score normalizes accordingly.

There are two different types of peer groups: *static and dynamic.*

The static method

Using the static method, data about users is obtained from databases such as Active Directory to create a peer group. Essentially, the grouping is based on attributes such as a user's department, designation, location, or their reporting manager. For example, all the employees who work in the finance department or all the employees who report to the same manager could constitute one peer group.

You can create multiple peer groups for a user this way. This is essential because if a user is a part of only one group, then the risk assessment and scoring might not be accurate. Each user may fall into more than one group: For instance, a designer in a marketing team will fall under the "Marketing" group as a whole, and also under a smaller, specific group called "Visual designers." They might also be grouped based on location, say "California." So, in this case, to accurately calculate the employee's risk, you'd have to look into the context (pattern of behavior) of all three groups. Also, if changes such as an employee changing roles or teams are not updated in Active Directory, the risk scoring accuracy decreases.

To ease risk assessment and enhance scoring accuracy, UEBA should also be capable of performing dynamic peer grouping.

The dynamic method

Using the dynamic method, UEBA builds peer groups based on behavioral data collected over time. With the dynamic mode of analysis, it's easier to compare the behavior of a user with that of their peers. It does this by checking if the behavior exhibited by a user for the first time is the expected behavior of that of their peers, or if it's an aberration. If it's found to be anomalous, the risk score increases accordingly. Unlike the static method, dynamic peer groups are created and analyzed based on patterns of similar behavior rather than grouping based on broad categories such as location.

However, for risk calculation based on dynamic grouping, care should be given in considering the size of the peer group as well as the frequency of the activity or behavior, which can be observed from historical data. This is because the smaller the peer group, the more alerts you can expect. With a larger group, it becomes easier to understand the context; accordingly, there are fewer alerts, so the results are more accurate. Similarly, if an action performed by a user is akin to that performed by their peers, then the risk scoring will not be negatively impacted and vice versa.

While it may seem like the dynamic method of peer grouping is better than the static method, a UEBA-integrated SIEM solution that is capable of building peer groups based on both methods is the most effective option for precise risk assessment and scoring.[10]

Now that you know the different methods of performing peer group analysis, let's go a step further and see how exactly peer grouping works.

Digging deeper into peer group analysis

As discussed earlier, the static method of peer grouping works based on common attributes (such as location, designation, and reporting manager) shared by users, and is rather straightforward. The dynamic peer grouping that works based on the behavioral similarity between users, however, raises several questions: How exactly is peer grouping performed? How does the algorithm determine which group a user should be part of? How does this impact risk scoring?

How is peer grouping performed?

The way peer grouping is performed depends on the algorithm employed in your SIEM solution. Different SIEM vendors have different algorithms, but an effective way of grouping users into peer groups is by using a clustering algorithm such as ConStream. Clustering algorithms work on the principle of clustering users into groups based on a similarity index. The ML algorithm learns the behavioral pattern of users over time and identifies which users have a similar pattern of behavior so it can assign them a similarity score (SS). Whether a user belongs to a cluster or not will be decided based on whether their SS is greater or lesser than the predefined threshold value. To understand this better, let's start from the ground up.

For performing peer group analysis, your anomaly detection engine will first look at the security analytics generated from ingesting the logs of your network devices. Each set of analytics generated by your SIEM solution is unique and will be associated with a different model. For example, analytics based on user logon times will constitute one model, and analytics based on SQL queries will constitute another model. A model is like a warehouse that stores the clusters associated with a particular type of activity. The number of clusters in a model will differ depending on the algorithm used.

Consider a model, M_1, that's associated with the logon activities of users. Assume that M_1 can host 1,000 clusters, named C_1, C_2,..., C_{1000}. Even though the clusters such as C_1 and C_2 fall under the broad category of "user logons," they are still different in terms of the events they consider for analysis. For example, C_1 will group events of users who exhibit certain types of logon behavior and C_2 will group events of users who exhibit another set of logon behavior, that is markedly different from the overall behavior of users in C_1. You can further classify user events into clusters based on other logon behaviors (C_3, C_4 and so on). Essentially, M_1 could have a maximum of 1,000 clusters and every cluster can have an "n" number of events added to it.

How does the algorithm determine which group a user should be a part of?

With so many clusters available, determining which cluster an event should be a part of depends on the similarity score. Whenever a logon event is recorded, the SS will be calculated between the event and the already available clusters, and this SS will then be compared to the threshold value. The threshold value is the definitive value for determining which cluster a user event should be a part of. For instance, let's say that for an event, E_1, to be added to cluster C_1, its SS should be greater than or equal to the threshold value, say 0.6. If it's not, then E_1 will become part of a new cluster called C_2 instead. Figure 5 below gives you an idea of how ML algorithms calculate SS and add events to clusters.

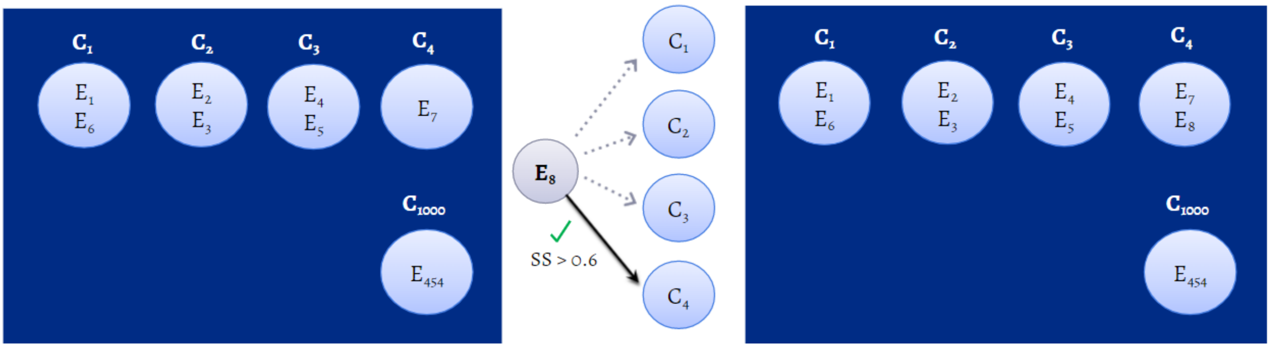

Figure 5: Machine learning algorithm calculating SS in the background to identify which cluster an event should be a part of

How does peer grouping impact risk scoring?

Once a user becomes a part of a group or cluster, their behavior will be compared with that of their peers, to identify if their activity was actually anomalous, or if it is the norm for peers in that group. Consider an organization called Anthem, where they have on-premises biometric-based attendance for its employees. You have to scan your fingerprint every time you enter or exit the organization, and that's how your working hours are recorded. However, to accommodate the remote users, they have a separate portal where you can use your credentials to mark your attendance. A marketing associate, Mark, who usually works from the office everyday has started to logon remotely every other day, and access the marketing database. In the general course of events, his action would be considered risky and his risk score will increase. However, if this action is compared with that of his peers, who have also been following the hybrid work model, the calculated confidence level will be readjusted and his risk score will immediately decrease, and normalize over time. The confidence level is a value calculated by your ML algorithm to give you an idea of how risky an event is.

With peer grouping, your risk assessment and risk scoring accuracy increase, and a great deal of false alerts can be reduced. However, it should be noted that while peer grouping positively impacts your risk scoring accuracy, it is not the sole deciding factor when it comes to a user's overall risk score. That will depend on various factors such as the weight assigned to the threat type, time decay factor, and seasonality.

Delving further

By now, you would have a fair understanding of how peer grouping works. But, you probably still have some questions lingering in your mind, such as:

- Can a user be in more than one group?
- What happens when there's an event which cannot be categorized into any of the existing clusters, and the model is already hosting 1,000 clusters?

To address the first question, let's consider two users, Mike and Harvey. Mike logs on remotely at 8am (E_1) whereas Harvey logs in from the office at 11am (E_2). Now, if you consider these events, you'll notice that both Mike and Harvey fall into two clusters each, one based on their mode of logon (Mike under C_1 and Harvey under C_2), and the other based on the time of their logon (Mike will be a part of C_3 and Harvey, C_4) as shown in Figure 6 below. So, yes, a user can belong to more than one cluster.

M_1

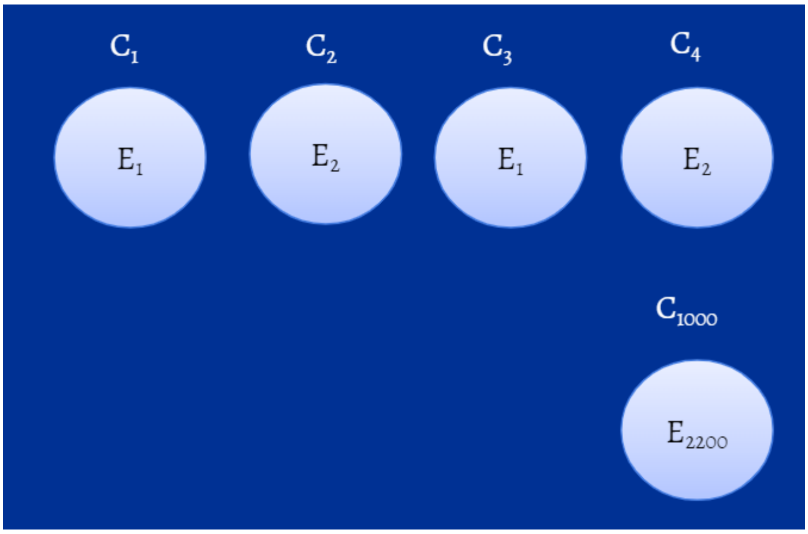

Figure 6: Representation of clusters of model, M_1

Now, coming to the second question, if such a situation were to arise, a cluster death occurs. Cluster death refers to the removal of an outdated cluster from the model to accommodate the addition of a new cluster. An outdated cluster is a cluster in which no new event has been added to in a very long time. Since there's no point in having outdated events, cluster death will occur. A new cluster is born in which the new event will be added, and this cycle continues.[11]

Benefits of peer grouping

Peer grouping helps make risk assessment and scoring more accurate. To better understand how, let's take a look at a few examples.

- Robert is a new associate who works in the finance department of an organization doing payroll. He usually works from 10am to 5pm most days. However, a couple of days before the month end, he logs in at 6am and logs off at 9:30pm. This is a deviation from his expected behavior and a SIEM solution would normally flag it as an anomaly and increase his risk score. However, if the SIEM solution was capable of performing peer group analysis, it would compare this behavioral pattern with that of his peers and understand that this behavior is normal for people working in the finance department towards the end of the month to prepare for payday. Thus, Robert's behavior wouldn't be considered anomalous, and his risk score wouldn't increase.
- Jane, an HR executive who's serving her notice, attempts to remotely access the marketing database to download the client list at 9pm on a Sunday. This is a red flag that a SIEM solution can identify if it has the capability to perform peer group analysis because the solution would check Jane's behavior against that of her fellow peers, and identify that HR executives don't normally access the marketing database, and certainly not at 9pm on a Sunday. So, the SIEM solution increases Jane's risk score drastically and alerts the security analyst. However, if the organization didn't have a SIEM solution with UEBA capabilities, then this abnormal behavior might not have been identified as risky immediately, and Jane might have successfully exfiltrated confidential information.

Seasonality

An activity is considered seasonal if it occurs with a specific degree of regularity, such as hourly, daily, weekly, or monthly. Your SIEM solution should be able to tag seasonal activities as non-anomalous. If an activity occurs out of its seasonal routine, it should be considered an anomaly, and your SIEM should be able to detect it.[12]

For example, this could be a user working on alternate Saturdays or a database that's queried heavily on the last day of every month. While these occurrences might be rare, they are not anomalous.

Scrutinizing seasonality in depth

It's crucial that we factor in seasonality for determining risk scores. If not, you might miss vital clues that could have detected and stopped an attack, or your security analysts might be inundated with numerous false alerts resulting in alert fatigue. Taking seasonality into account in your SIEM solution will enhance your risk scoring accuracy and reduce false positives. This will give your analysts time to prioritize and respond to genuine threats. Now that you know why seasonality is important, let's take a look at how it works.

How does seasonality work?

Seasonality is fueled by ML algorithms that study the past behavior of users and entities to determine if an activity is anomalous. The following four time-based parameters are looked at to determine if activities follow a seasonal pattern:

- Time of the day (ToD)
- Date of the month (DoM)
- Day of the week (DoW)
- Week of the month (WoM)

To fully understand what seasonality means and how it improves your risk scoring accuracy, let's take a look at the following example and make a comparison between how it would play out if a SIEM solution offered seasonality and if it didn't.

Let's consider the organization Anthem again. On September 30, Anthem's payroll server is accessed multiple times in one-hour time intervals between 9am and 2pm (see Table 3).

Time period	Number of accesses in the payroll server
09:00 - 10:00	100
10:00 - 11:00	125
11:00 - 12:00	150
12:00 - 13:00	250
13:00 - 14:00	750

Table 3: Payroll server access data

In this example, the ToD is the time interval in which the server was accessed (for example, 09:00-10:00), the DoM is 30, the DoW is 6 (assuming that the algorithm indexes the first day of the week, Sunday, as "1"), and the WoM is 5 (as September 30 is the fifth Friday of the month).

Let's understand how these accesses will be treated under two scenarios: 1) with a SIEM solution that does not have seasonality, and 2) with a SIEM solution that does have seasonality.

Scenario 1: SIEM solution without seasonality

When a SIEM solution doesn't have seasonality, it sets the time interval as one hour (this might differ depending on the algorithm being used) and compares the number of accesses in each one-hour time period to a dynamically calculated threshold value. This threshold value is calculated after considering the number of accesses in all previous one-hour intervals. This means that the algorithm does not look at the history of the number of accesses made between a particular time period, say, 9 and 10am. It looks upon it as just another one-hour interval no different from the one before it or the one after.

Simply put, a SIEM solution with no seasonality does not consider the time interval in which the access happens to trigger an anomaly. That's why the difference in the number of accesses made between 12pm-1pm and 1pm-2pm are considered anomalous (see Table 4). The number of accesses during these time periods (250 and 750) must have far exceeded the threshold value.

Time period	Number of accesses in a server	Counts vs. time interval	Result
09:00 - 10:00	100		Normal
10:00 - 11:00	125		Normal
11:00 - 12:00	150		Normal
12:00 - 13:00	250		Anomaly
13:00 - 14:00	750		Anomaly

Table 4: UEBA without seasonality

Scenario 2: SIEM solution with seasonality

In this case, the algorithm will first compare the number of accesses on the payroll server with the time of the day to decide if it's normal or not. For example, if we consider the data in Table 1, then the first check the algorithm will perform is to see if 100 accesses between 9am and 10am is normal or not. The threshold will be a function of the number of accesses between 9am and 10am across all previous days for which data is available.

If an anomaly is not triggered, the algorithm proceeds with the second check, where it determines if the number of accesses (100) is normal for a particular date of the month—in this case, the 30th—for the same time period, i.e., 9am to 10am. This means that the algorithm will check the number of accesses made between 9am and 10am on the 30th of every month for as long as the data exists. If this check also yields normal results, it proceeds with the third check. If not, it's identified as an anomaly.

Assuming that the access count is normal so far, the third check is performed, which involves taking into account the day of the week. In this case, it's the sixth day of the week, Friday. Again, this means that the algorithm compares the number of accesses between 9am and 10am on the Friday of interest to that of all the previous Fridays to arrive at a conclusion.

If this again does not yield an anomaly, the final check is done where the count is considered along with the week of the month. So, the algorithm checks if 100 accesses between 9am and 10am on the Fridays of the fifth week of a month are normal or anomalous. The same steps are followed for the other time periods as well, as shown in Table 5.

Time period	Number of accesses in a server	ToD	DoM	DoW	WoM	Result
09:00 - 10:00	100	✓	✓	✓	✓	Normal
10:00 - 11:00	125	✓	✓	✓	✓	Normal
11:00 - 12:00	150	✓	✓	✓	✓	Normal
12:00 - 13:00	250	✓	✓	✓	✓	Normal
13:00 - 14:00	750	✓	✓	✓	✓	Normal

Table 5: Seasonality in action

Now, you might wonder: Why is the algorithm considering 750 as normal when it clearly looks like a case of heightened activity on the payroll server?

The answer to that lies in the historical data. In this case, the historical data would have shown that employees typically access the payroll server to download their pay stub during their lunch hours—between 12 and 2pm—on the last working day of the month (usually the 30th). So, when the algorithm does its usual check, it's able to identify the 750 accesses as normal.

Digging deeper

By now, you probably have a good grasp of how seasonality works. Yet, you might still have a few questions, such as:

- Is seasonality applicable to users as well?
- Will the threshold calculated by the algorithm differ for different time intervals?

Let's address these questions one at a time.

Is seasonality applicable to users?

Absolutely! Every user could perform a task that's seasonal in nature and specific only to that user. The algorithm will identify this behavioral pattern and alert you if there's any deviation from it. For example, Stacey, a senior marketing associate, updates the new leads generated by her team on the consolidated marketing database on the last Friday of every month. So, the algorithm expects this behavior from her. However, if Stacey were to not only access the database, but access it multiple times on a Tuesday, then the algorithm would identify this as an anomaly. Stacey's risk score increases, and an alert is triggered to notify the security analyst.

Does the algorithm calculate a different threshold for different time intervals?

Yes. The number of activities performed in a particular hour, day, and even month are different. As we saw in the Anthem example above, it's normal for the payroll server to be accessed 100 times between 9am-10am and 750 times between 1pm-2pm on the same day. The number of times a particular activity is performed in a particular time period, or any given time, is going to differ, and your algorithm must be capable of calibrating this dynamically. Only if it does, will your risk scoring be accurate.[13]

User identity mapping

The UEBA engine in SIEM solutions also performs UIM to track anomalies more effectively. A user at any given time could be accessing different applications and devices across your network, and not always using the same username or credentials.

For example, John has the username, *John Watson*, on his Windows device whereas his username on Azure is *john.watson@xyz.com*, and on SQL server, it is *John*.

This raises the question: Does a SIEM solution know that the different activities performed using different usernames are actually the actions of a single user (John)? If so, how does it do that? That's where UIM comes in.

The UIM feature of UEBA helps link all distinct user registries with a base registry like Active Directory to determine the activity of a single user across multiple domains and correlate these activities to identify anomalies. When all of the user's identities are mapped, they will have just one representation and risk score. By mapping all the different user identities to the single user, UEBA improves the risk scoring accuracy of users, and helps us discern the bigger picture of the incident.[14]

How ManageEngine Log360 can help you detect anomalies

ManageEngine Log360 is a unified SIEM solution with integrated UEBA, DLP, CASB, and SOAR capabilities that can help you detect and thwart insider threats and external attacks.

- Log360 provides a comprehensive dashboard that tells you about the anomaly trends, recent trends, recent anomalies, top 10 anomalous activities, anomalies based on categories, risk levels, user risk score distribution, top 10 users by risk score, and so much more, as shown in Figures 7 and 8.

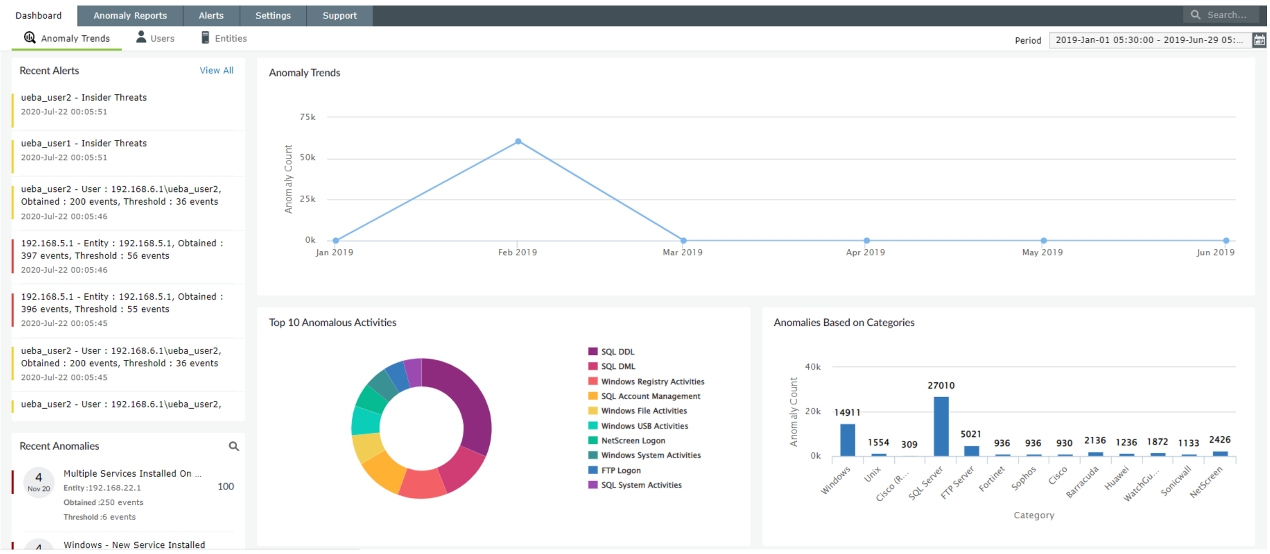

Figure 7: Log360 dashboard showing anomaly trends

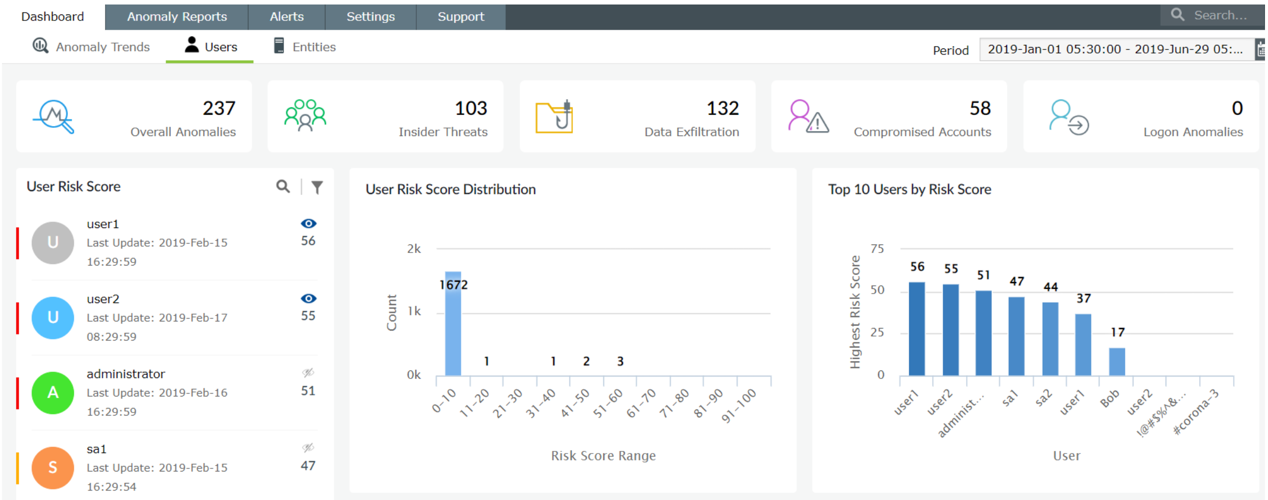

Figure 8: Log360 dashboard showing user risk score distribution

- With its numerous predefined and custom reports, Log360 helps you gain a better understanding of the events (both normal and anomalous) happening across your network, as shown in Figure 9.

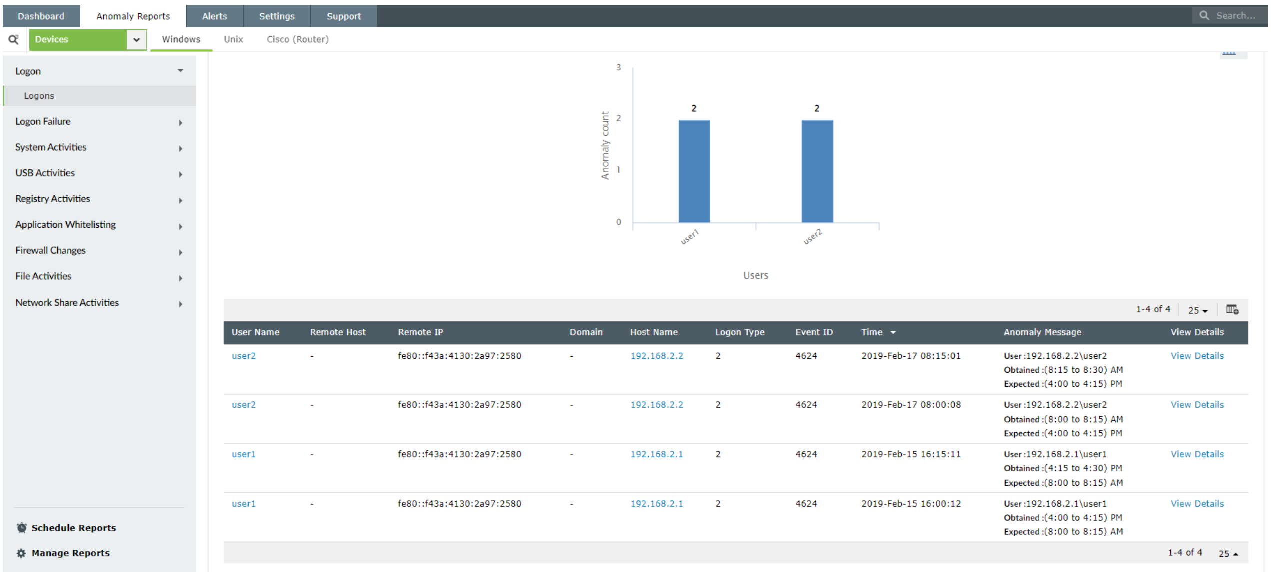

Figure 9: Logon anomaly reports as observed in Log360

- Log360 improves risk scoring accuracy by factoring in peer group analysis, seasonality, UIM, anomaly modeling, and enables you to customize your risk score as shown in the figures below.

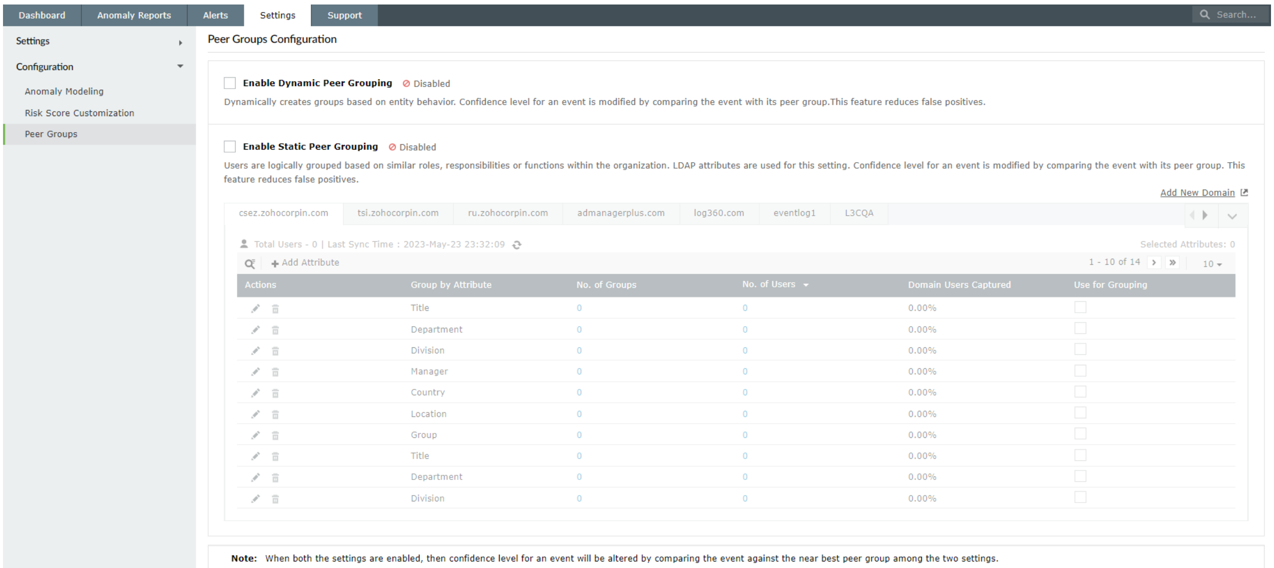

Figure 10: Configuring peer grouping in Log360

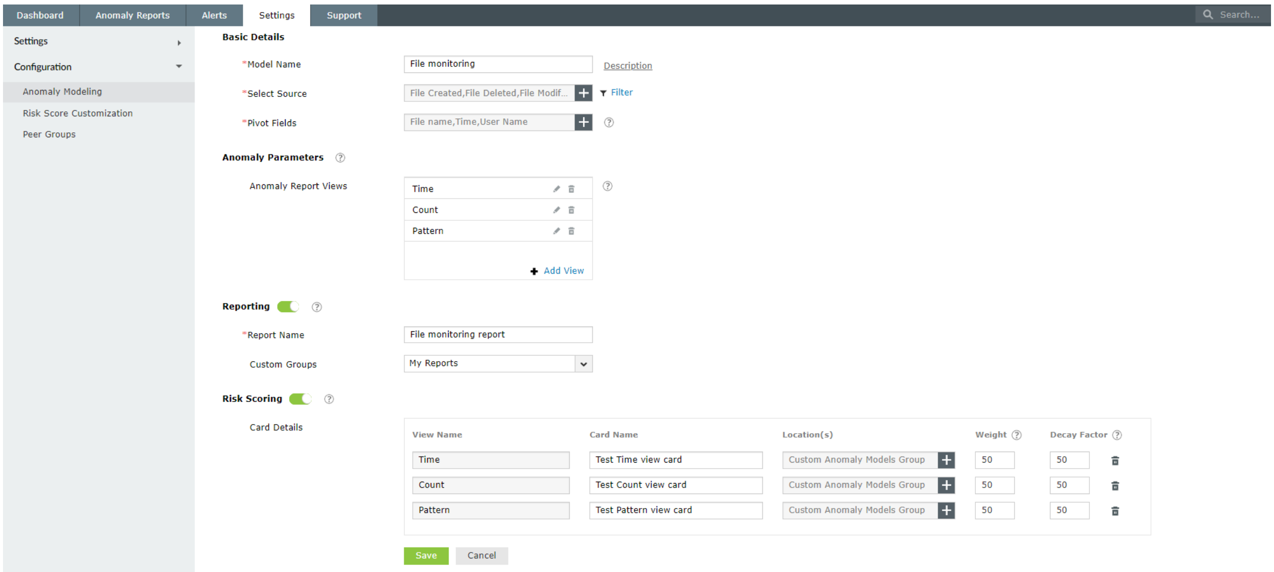

Figure 11: Building custom anomaly models

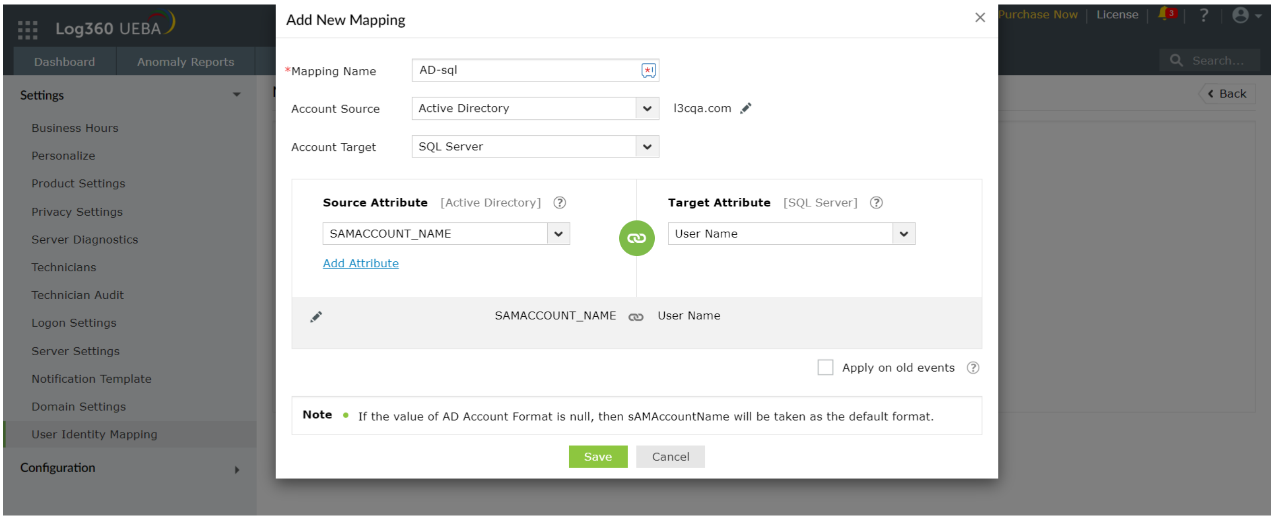

Figure 12: User identity mapping in Log360

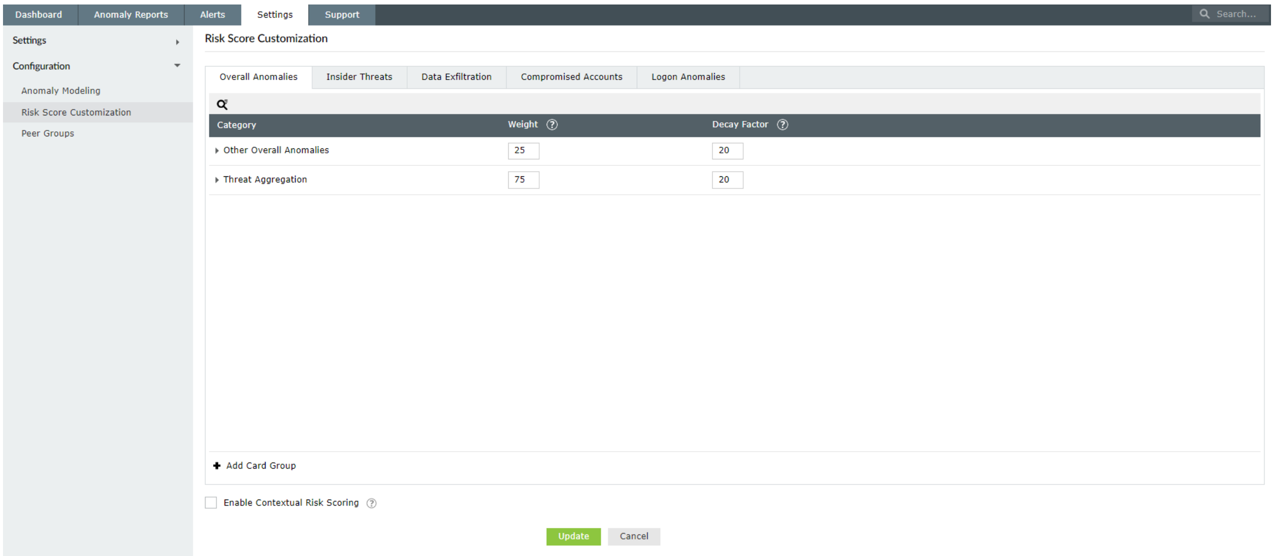

Figure 13: Risk score customization provided by Log360

To learn more, schedule a personalized demo of Log360 and talk to our solution experts.

About the author

Hiranmayi Krishnan

Hiranmayi is a cybersecurity enthusiast who works at ManageEngine, the IT management division of Zoho Corporation. She is an avid blogger of the latest trends in cybersecurity, and her main areas of interest are user and entity behavior analytics, and incident management through security orchestration, automation, and response.

About ManageEngine Log360

ManageEngine Log360, a unified SIEM solution with integrated DLP and CASB capabilities, helps enterprises thwart attacks, monitor security events, and comply with regulatory mandates. The solution comes bundled with a log management component that provides better visibility into network activity, an incident management module that helps quickly detect, analyze, prioritize, and resolve security incidents, an ML-driven user and entity behavior analytics add-on that baselines normal user behaviors and spots anomalous user activities, and a threat intelligence platform that leverages dynamic threat feeds for security monitoring and helps enterprises stay on top of attacks.

For more information about Log360, visit manageengine.com/log-management

To explore the security benefits of Log360, schedule a personalized demo

To explore the security benefits of Log360

Personalized demo   Download

References

1. Krishnan, Hiranmayi. "Unraveling the behavioral blueprint of users and entities with UEBA." *ManageEngine Expert Talks* (blog). March 2, 2022. Accessed August, 2023. https://www.manageengine.com/log-management/cyber-security/unraveling-the-behavioral-blueprint-of-users-and-entities-with-ueba.html

2. SonicWall, "2023 SonicWall Cyber Threat Report," 4-5. Accessed August, 2023. https://www.sonicwall.com/medialibrary/en/white-paper/2023-cyber-threat-report.pdf

3. Chekalov, Maxym. "22 Insider Threat Statistics to Look Out For in 2023." *Techjury* (blog). August 15, 2023. Accessed August, 2023. https://techjury.net/blog/insider-threat-statistics/

4. IBM. "Cost of a Data Breach Report 2022", 14-15. Accessed August, 2023. https://www.ibm.com/downloads/cas/3R8N1DZJ

5. ManageEngine. *Understanding User and Entity Behavior Analytics (UEBA): How machine learning can help secure your business.* 1-16. Accessed August, 2023. https://download.manageengine.com/log-management/user-and-entity-behavior-analytics-guide.pdf

6. Vaidyanathan, Ram. *Anomaly Detection In Cybersecurity For Dummies.* 2022. Accessed August, 2023. https://download.manageengine.com/log-management/ueba/user-and-entity-behavior-analytics-software/anomaly-detection-in-cybersecurity-for-dummies-book.pdf

7. Ibid.

8. A, Anupama. "The importance of anomaly modeling when discerning user behavior." *ManageEngine Expert Talks* (blog). December 28, 2022. Accessed August, 2023. https://www.manageengine.com/log-management/cyber-security/anomaly-modeling-understanding-user-behavior.html

9. Krishnan, Hiranmayi. "How does risk scoring in anomaly detection work?" *ManageEngine Expert Talks* (blog). April 24, 2023. Accessed August, 2023. https://www.manageengine.com/log-management/cyber-security/how-ueba-risk-scoring-works.html

10. Krishnan, Hiranmayi. "Enhance risk scoring accuracy and anomaly detection with peer group analysis." *ManageEngine Expert Talks* (blog). May 30, 2022. Accessed August, 2023. https://www.manageengine.com/log-management/cyber-security/anomaly-detection-with-peer-group-analysis.html

11. Krishnan, Hiranmayi. "Digging deeper into peer group analysis." *ManageEngine Expert Talks (blog)*. September 12, 2022. Accessed August, 2023. https://www.manageengine.com/log-management/cyber-security/digging-deeper-into-peer-group-analysis.html

12. Vaidyanathan, Ram. "Why seasonality factors are important to anomaly detection in cybersecurity." *ManageEngine Expert Talks* (blog). February 10, 2022. Accessed August, 2023. https://www.manageengine.com/log-management/cyber-security/importance-seasonality-factors-anomaly-detection-cybersecurity.html

13. Krishnan, Hiranmayi. "Scrutinizing seasonality in depth." *ManageEngine Expert Talks* (blog). November 28, 2022. Accessed August, 2023. https://www.manageengine.com/log-management/cyber-security/scrutinizing-seasonality-in-depth.html

14. K, Varun. "User identity mapping for improved anomaly detection." *ManageEngine Blog* (blog). May 22, 2023. Accessed August, 2023. https://blogs.manageengine.com/active-directory/log360/2023/05/22/user-identity-mapping-for-improved-anomaly-detection.html