The SRI IDES Statistical Anomaly Detector

Harold S. Javitz and Alfonso Valdes
SRI International
Menlo Park, CA 94025

Abstract

SRI International's real-time intrusion-detection expert system (IDES) system contains a statistical subsystem that observes behavior on a monitored computer system and adaptively learns what is normal for individual users and groups of users. The statistical subsystem also monitors observed behavior and identifes behavior as a potential intrusion (or misuse by authorized users) if it deviates significantly from expected behavior. The multivariate methods used to profile normal behavior and identify deviations from expected behavior are explained in detail. The statistical test for abnormality contains a number of parameters that must be initialized and the substantive issues relating to setting those parameter values are discussed.

Overview

The SRI IDES[1] system is a real-time intrusion detection expert system that observes behavior on a monitored computer system and adaptively learns what is normal for individual users, groups, remote hosts and the overall system [1]. Observed behavior is flagged as a potential intrusion if it deviates significantly from expected behavior or it triggers a rule in the expert-system rule base. This paper describes the multivariate statistical engine.

The IDES statistical anomaly detector, maintains a statistical subject knowledge base consisting of *profiles.* A profile is a description of a subject's normal (i.e., expected) behavior with respect to a set of intrusion-detection measures. Profiles are designed to require a minimum amount of storage for historical data and yet record sufficient information that can readily be decoded and interpreted during anomaly detection. Rather than storing all historical audit data, the profiles keep only statistics such as frequency tables, means, and covariances.

The deductive process used by IDES in determining whether behavior is anomalous is based on statistics, controlled by dynamically adjustable parameters, many which are specific to each subject. Audited activity is described by a vector of intrusion-detection variables, corresponding to the measures recorded in the profiles. Measures can be turned "on" or "off" (i.e., included in the statistical tests), depending on whether they are deemed to be useful for that target system. As each audit record arrives, the relevant profiles are retrieved from the knowledge base and compared with the vector of intrusion-detection variables. If the point in N-space defined by the vector of intrusion-detection variables is sufficiently far from the point defined by the expected values stored in the profiles, with respect to the historical covariances for the variables stored in the profiles, then the record is considered anomalous. Thus, the statistical procedures pay attention not only to whether an audit variable is too high or too low, but also to whether any audit variable is too high or too low relative to the values of the other audit variables (in other words, the correlation between variables). Thus, IDES evaluates the total usage pattern, not just how the subject behaves with respect to each measure considered singly.

The statistical knowledge base is updated daily using the most recent day's observed behavior of the subjects. Before incorporating the new audit data into the profiles, the frequency tables, means, and covariances stored in each profile are first aged by multiplying them by an exponential decay factor. Although this factor can be set by the user, we believe that a value that reduces the contribution of knowledge by a factor of 2 for every 30 days is appropriate (this is the daily profile aging factor). This method of aging has the effect of creating a moving time window for the profile data, so that the expected behavior is influenced most strongly by the most recently observed behavior. Thus, IDES adaptively learns subjects' behavior patterns; as subjects alter their behavior, their corresponding profiles change.

The details of the implementation of the SRI IDES statistical anomaly detector are contained in the following sections, which are briefly summarized below:

- *The IDES Score Value.* Each time an audit record is generated, a summary test statistic (denoted IS) is generated, reflecting the degree to which recent behavior is similar to the historical profile. Large values are indicative of abnormal behavior. The security officer can track changes in the summary test statistic using a time series and is alerted when appropriate thresholds are exceeded.

- *How* IS *is Formed from Individual Measures.* The IS statistic is formed from many individual constitutent measures (denoted S_i). The formula for computing IS from the S_i is provided.

[1] IDES is supported by the U.S. Navy, SPAWAR, which funded SRI through U.S. Government Contract No. N00039-89-C-0050.

- *Individual Measures.* Each individual measure S_i reflects the extent to which a particular type of recent behavior (such as file accesses or CPU time used) is similar to the historical profile for that type of behavior.

- *Heuristic Description of the Relationship of* S *to* Q. Each S_i statistic is a transformation of a more basic statistic Q_i. For example, if S_i reflects the degree of abnormality of recent CPU time usage, then the corresponding Q_i is a measure of how much CPU time was actually used in the recent past. Si is computed by comparing the current value of Q_i to its historical profile (that is, the historical probability distribution of Q_i). When the most recent value for Q_i has a low probability of occurrence, S_i has a large value, and vice-versa.

- *Algorithm for Computing* S *from* Q. The formula for deriving S_i from Q_i is provided, under the assumption that the historical probability distribution for Q_i is available.

- *Computing the* Q *Statistic for Ordinal Measures.* The procedure for computing a Q_i statistic when the underlying measure is ordinal (e.g., a counting measure such as CPU time or I/O counts) is presented. Q_i is shown to be an exponentially weighted sum of the changes that have occurred in the underlying measure. The half-life of the Q_i statistic is typically on the order of few hours or a few hundred audit records.

- *Computing the Frequency Distribution for* Q. The procedure for computing the historical profile for Q_i is presented. The historical profile is also an exponentially weighted sum with a half-life typically on the order of 30 days. It is updated nightly.

- *Computing the* Q *statistic for Categorical Measures.* The general formula for computing a Q_i statistic for a categorical measure (such as the names of files accessed or the names of terminals used for logging in) is presented. Q_i is an exponentially weighted sum. It tends to attain large values when the categories for the underlying measures that have been recently observed (for example, the names of the particular files accessed) have been only infrequently observed in the past. Depending on the parameters used in the formula for Q_i, this statistic may also be sensitive to the number of recent occurrences of infrequently occurring categories.

- *The Binary, Linear Binary, and Intermediate Forms for the Categorical* Q *Statistic.* The values of two parameters in the formula for the Q statistic for categorical measures may be varied to achieve different forms for the Q statistic, which are discussed herein.

- *Specification of the Likelihood of Occurrence Values* f_m. For each categorical measure, a variety of procedures are available for quantifying the relative frequency with which categories of the measure have occurred in the past. The following three sections (*The Fixed Chronological Time Period Method*, *The Relative Frequency of Occurrence Method*, and the *Absolute Exponentially Weighted Interarrival Time Method*) discuss three such procedures.

- *Specification of the Function* g(). This section describes another "parameter" of the formula for the Q statistic for categorical measures.

- *Decision Options Affecting the* Q *Statistic.* This section and the following three sections (*How Probabilities of Different Magnitudes Affect the* Q *Statistic*, *How Multiple Occurrences Affect the* Q *statistic*, and *Whether Time Should Be Chronological or Count Related*) discuss factors that should be taken into account when setting the parameters of the Q statistic.

The IDES Score Value

For each audit record generated by a user, the IDES system generates a single test statistic value (the IDES score value, denoted *IS*) that summarizes the degree of abnormality in the user's behavior in the "near" past. (The concept of "near" past is defined later.) Consequently, if the user generates 1000 audit records in a day, there will be 1000 assessments of the abnormality of the user's behavior. Because each assessment is based on the user's behavior in the near past, these assessments are not independent.

Large values for *IS* are indicative of abnormal behavior, and values close to zero are indicative of normal behavior (e.g., behavior consistent with previously observed behavior). For the *IS* statistic, we select one or more "critical" values that are associated with appropriate levels of concern and inform the security officer when these levels are reached or exceeded. For example *IS* values between 0 and 22.0 might be associated with no concern, values between 22.0 and 28.0 might be associated with a "yellow" alert, and values in excess of 28.0 might be associated with "red" alerts. The critical values are selected so that they have a probabilistic interpretation; for example, we might expect false red alerts only once every 100 days. However, the security officer has the freedom to raise or lower the critical values for each system user, in case there is a need to monitor a particular user's behavior more closely or in case the standard critical values result in too many false alerts for a particular user.

Because the *IS* statistic summarizes behavior over the near past, and sequential values of *IS* are dependent, the *IS* values will slowly trend upward or downward. Once the *IS* statistic is in the red alert zone, it will take a number of audit records before it can return to the yellow or green zone. To avoid inundating the security officer with notification of continued red alerts we only notify the security office when a change occurs in the alert status, or when the user has remained in a yellow or red zone for a specific time. In addition, the security officer is able to generate a time plot of the IS values for a user and thus assess whether or not the user's IS statistic indicates a return to more normal behavior.

How IS is Formed from Individual Measures

The IS statistic is itself a summary judgement of the abnormality of many measures. Suppose that there are n such constituent measures, and let us denote these individual measures by $S_i, 1 \leq i \leq n$. Let the correlation between S_i and S_k be denoted by C_{ik}, where $C_{ii} = 1.0$. Then the IS statistic can be written

$$IS \quad = \quad (S_1, S_2, \cdots, S_n)C^{-1}(S_1, S_2, \cdots, S_n)^t$$

where C^{-1} is the inverse of the correlation matrix of the vector $(S_1, S_2, \cdots, S_n)$, and $(S_1, S_2, \cdots, S_n)^t$ is the transpose of that vector. Each of the S_i measures is constructed in such a manner that it can take only positive values, and for the most part the correlations also tend to be positive or zero. (For technical reasons, the correlations are not allowed to exceed 0.90 in absolute magnitude). IS tends to accumulate evidence from the separate measures in an additive fashion. For example, if all the measures were independent, the correlation matrix would be the identity matrix, and IS would simplify to $S_1^2 + S_2^2 + \cdots + S_n^2$, the sum of the squares of the measures. When two measures are highly correlated, then because of the way the S_i are defined, the effect of the inverse correlation matrix is to usually give each measure approximately half of the weight that it would otherwise receive. The IS statistic doesn't tell the security officer which constituent measures are contributing the most to the decision that behavior is abnormal, only the summary judgement that behavior is abnormal. However, when the IS statistic is large, the security officer interface indicates which individual measures have substantially contributed to the IS value.

Individual Measures

The individual S measures each represent some aspect of behavior. For example, an S measure might represent file accesses, CPU time used, or terminals used to log on. Two S measures might also represent only slightly different ways of examining the same aspect of behavior. For example, both S_i and S_j might represent slightly different ways of examining file access, where the differences manifest themselves in different selections of parameters used to construct these measures. In many such cases, we would expect their correlations to be high. Fortunately, the IS statistic will adjust automatically (via the correlation matrix) for the diminishing usefulness in examining the same aspect of behavior from more and more viewpoints that do not represent fundamentally different aspects of behavior.

Heuristic Description of the Relationship of S to Q

Each S measure is derived from a corresponding statistic that we will call Q. In fact, each S measure is a transform of the Q statistic that indicates whether the Q value associated with the current audit record and its near past is unlikely or not. For example, consider an S measure that represents CPU time used. The corresponding Q statistic would also measure CPU time used in the near past, and might be expressed in units of milliseconds. By observing the values of Q over many audit records, and by selecting appropriate intervals for categorizing Q values, we could build a frequency distribution for Q. For example, we might find the following:

- 0.5% of the Q values are in the interval 0 to 1 millisecond
- 7% are in the interval 1 to 2 milliseconds
- 15% are in the interval 2 to 4 milliseconds
- 42% are in the interval 4 to 8 milliseconds
- 12% are in the interval 8 to 16 milliseconds

The S statistic would be a large positive value whenever Q was in the interval 0 to 1 millisecond (because this is a relatively unusual value for Q) and would be close to zero whenever Q was in the interval 4 to 8 milliseconds (because this is a relatively frequently seen interval). We do not require that the frequency distribution of Q be unimodal. For example, if a particular user does CPU-nonintensive tasks on some days and CPU-intensive tasks on other days, we might expect that the CPU Q measure would have a bimodal distribution. The selection of appropriate intervals for categorizing Q is important, and it is better to err on the side of too many intervals than too few. We are currently using 16 intervals for each Q measure, with interval spacing determined dynamically for each user. The last interval does not have an upper bound, so that all values of Q belong to some interval.

Algorithm for Computing S from Q

Assume for the moment that we have defined a method for updating the Q value each time a new audit record is received, and that we have defined intervals that we have used to develop a historical frequency disttribution for Q. The algorithm for converting individual Q values to S values is as follows:

- Let P_m denote the relative frequency with which Q belongs to the mth interval. In our example the first interval is 0 to 1 millisecond and P_1 equals 0.5%.
- Let $i1$ denote the interval with the smallest P value, $i2$ denote the interval with the second smallest P value and so forth. For example, we might find that the first interval has the smallest P value, the 10th interval has the second smallest P value, and so on, in which case $i1 = 1$, $i2 = 10$, and so forth.

- Let $TPROB_1 = P_{i1}$, $TPROB_2 = P_{i1} + P_{i2}$, $TPROB_3 = P_{i1} + P_{i2} + P_{i3}$, and so forth. The $TPROB_i$ values increase as i increases and the final $TPROB$ value is equal to 1.0.
- For each $TPROB_i$ value, find the value s_i such that the probability that a normally distributed variable with mean 0 and variance 1 is larger than s_i in absolute value equals $TPROB_i$. The value of s_i satisfies the equation $Prob(|N(0,1)| \geq s_i) = TPROB_i$, or $s_i = \Phi^{-1}(1 - \frac{TPROB}{2})$ where Φ is the cumulative distribution function of an $N(0,1)$ variable. For example, if $TPROB_i$ is 5%, then s_i is equal to 1.96. If $TPROB_i$ is zero, then we set s_i equal to 3.0. The s_1 value corresponding to $TPROB_1$ is the largest s value, and the s value corresponding to the largest $TPROB$ value is equal to 0.0. The following graph in Figure 1 shows the relationship between $TPROB_i$ values and the corresponding S_i values.

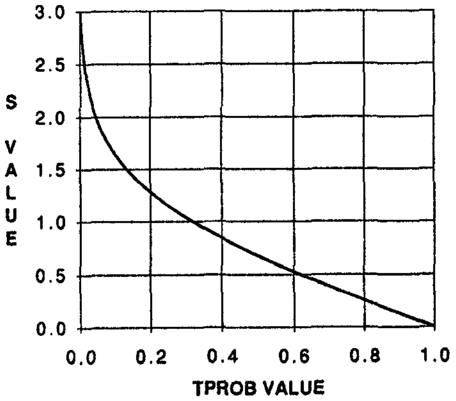

Figure 1: Relationship of S to $TPROB$

- Suppose that after processing an audit record we find that the Q value is in the mth interval. Furthermore, the mth interval is the interval with the ith largest P value. Then S is set equal to s_i, the s value corresponding to $TPROB_i$.

In practice this algorithm is easy to implement, and the calculations of the s_i values are done only once at update time (usually close to midnight). Each interval for Q is associated with a single s value, and when Q is in that interval, S takes the corresponding s value.

Computing the Q Statistic for Ordinal Measures

The simplest version of the Q statistic occurs when the underlying measure is ordinal (i.e., a counting measure). For example, the ordinal measure might be CPU time, the number of files accessed (without regard to which files are accessed), the number of logons from locations outside the facility (without regard to where outside the facility the logon occurs), and so forth. This section examines how the Q statistic is defined for such ordinal measures.

When a user is first audited, that user has no history. Consequently, we chose some convenient value to begin the Q statistic history. For example, we might let each Q statistic be zero, or some value close to the mean value for other (we hope, similar) users.

The Q statistic for each measure is updated each time a new "appropriate" audit record is generated. There are two possible definitions for appropriate. The first is to consider any audit record to be appropriate, whether or not it contains any additional information about the measure being examined by Q. The second definition is to consider only those audit records that contain information about the measure being examined by Q to be appropriate. For example, if the measure being examined is I/O activity, then under the first definition of appropriate, Q would be updated each time an audit record arrived, even if that audit record contained only information about file accesses, and none about I/O activity. In this case, the updating procedure would effectively assume that I/O activity is unchanged from its last value. (As explained later, the value for Q might change even if the I/O activity is unchanged.) Under the second definition for appropriate, we would not update Q until an audit record appeared that contained additional information about I/O activity. The first definition has the advantage of uniformity across all Q statistics. In addition, as discussed later, it is easier to define the "recent" past in terms of a known time interval or number of audit records generated. The second definition has the advantage that for measures of rare activity (for example, network usage) Q need not be updated for the many irrelevant intervening audit records, and these irrelevant records will not alter the value for Q. (Note: these update procedures for Q should not be confused with the daily profile update operation discussed earlier.)

Let us now consider how to update Q. Let Q_n be the value for Q after the nth appropriate audit record, and let Q_{n+1} be the value for Q after the $(n+1)$st appropriate audit record. The formula for updating Q is as follows:

$$Q_{n+1} = D(A_n, A_{n+1}) + 2^{-rt} \times Q_n$$

The symbols in the formula for Q_{n+1} denote the following:

- A_n represents the nth appropriate audit record and A_{n+1} represents the $(n+1)$st appropriate audit record.
- $D(A_n, A_{n+1})$ denotes the change that has occurred in the measure being examined from A_n to A_{n+1}. For example, if Q is I/O activity, then $D(A_n, A_{n+1})$ represents the increment in the I/O count that has occurred between the nth and $(n+1)$st appropriate audit record. If all audit records are considered to be appropriate, and A_{n+1} contains no information about I/O activity, then the I/O count is assumed not to have changed, and $D(A_n, A_{n+1})$ is 0.

- The variable t represents the "time" that has elapsed between A_n and A_{n+1}. There are two possible units for "time". The first is clock time. In this case, if A_n is generated at 2:08:32 and A_{n+1} is generated at 2:09:45, then t is equal to 73 seconds. If time is measured chronologically, it is probably preferable to define all audit records as appropriate, although it is not necessary to do so. The second possible unit of time is appropriate audit record counts. In this case, t is always equal to 1.0, because A_{n+1} is always one appropriate audit record after A_n (regardless of how many actual audit records have occurred between A_n and A_{n+1}).
- The decay rate r determines the "half-life" of the measure Q. Large values of r imply that the value of Q will be primarily influenced by the most recent few appropriate audit records. Small values of the decay rate r mean that Q will be influenced by audit records in the more distant past. For example, if time is measured chronologically in hours, then a half-life of 4.0 hours corresponds to an r value of $0.25 = -\log_2(0.5)/4.0$. If time is measured in appropriate audit records, then a half-life of 100 appropriate audit records corresponds to an r value of 0.01.

In our example, Q is the sum of I/O activity over the entire past usage, exponentially weighted so that the more current usage has a greater impact on the sum. Thus Q is more a measure of "near" past behavior than of distant past behavior, even though all past behavior has some influence on Q. The Q statistic has the important property that it is not necessary to keep extensive information about the past to update Q. For example, if Q had been based on a moving window of 200 appropriate audit records, then it would have been necessary to keep information on the most recent 200 appropriate audit records in memory, and this would probably preclude real-time processing.

We note that we can write Q in a closed formula as follows:

$$Q = \sum_{k \geq 1} D_k \times 2^{-rt_k}$$

where

- k is an index of appropriate audit records with $k = 1$ denoting the most recent appropriate audit record and the sum extending over all appropriate audit records.
- $D_k = D(A_k, A_{k+1})$ is the change that occurred between the $(k+1)$st and kth appropriate audit records.
- t_k is the time that has elapsed between the kth and most recent appropriate audit record.

The three important decisions involved in defining Q were as follows:

- Decision #1: Determine whether Q should be updated after any audit record or only after audit records that contain new information about the behavior being examined by Q (i.e., what is the definition of an "appropriate" record for updating).
- Decision #2: Determine whether time should be measured chronologically or using appropriate audit record counts. We recommend that if Q is updated after any audit record, then time should be measured chronologically, and if Q is updated only after audit records that contain new information about the behavior being examined, then time should be measured using appropriate record counts.
- Decision #3: Determine the half-life of the Q statistic. This half-life should be sufficiently short that Q can respond rapidly to changes in behavior, but also sufficiently long that it is based on a reasonable amount of data on which to judge behavior.

Some of the issues that should be taken into account in these decisions are described later. It should be noted that these three decisions are also necessary when computing Q statistics for categorical measures.

Computing the Frequency Distribution for Q

We have previously alluded to the frequency distribution for Q, without specifying how that frequency distribution should be calculated. As before, let P_m denote the relative frequency with which Q is in the mth interval. The formula for calculating P_m is as follows:

$$P_m = \frac{\sum_{k \geq 1} W_{mk} 2^{-bt_k}}{\sum_{k \geq 1} 2^{-bt_k}}$$

where

- k is an index of audit records with $k = 1$ denoting the most recent audit record and the sum extending over all audit records for the user being monitored (not just audit records for the measure being examined by Q). It is important that the sum extend over all audit records because the IS statistic, which is ultimately derived from the Q statistics, is assessed after each audit record.
- W_{mk} is an indicator function that attains the value of 1 if, after the processing of the kth audit record, Q is in the mth interval.
- t_k is the time that has elapsed between the kth and most recent audit record. Time may be measured either chronologically, or by a counting variable that is incremented by 1 at the arrival of any

new audit record. We recommend that time be measured in the same way (e.g., either chronologically or as counts) as determined in decision #2.

- b is the decay rate of the data being used to compute P_m. For example, if t were measured in days, a half-life of 60 days would correspond to a "b" value of 0.0116.

To calculate the values of P_m we must make an additional decision:

- Decision #4: Determine the half-life of the frequency distribution of the Q statistic. This half-life should be sufficiently long that the normal behavior of the Q statistic can be quantified, but not so long that behavior in the very distant past significantly influences our perception of the normality of current behavior.

Although the formula for P_m indicates that it should be updated after every audit record, it is sufficient to calculate P_m once at update time as long as the interval between updates is short relative to the half-life of the data used to calculate P_m. This greatly reduces the computational burden on the IDES system. For example, if t is measured chronologically and updating occurs once per day, the procedures for updating the frequency distribution for Q are as follows:

- We maintain a historical count (also called effective n) for Q as well as a vector of historical probabilities (i.e., P_m). These summarize behavior up to and including the last update.

- Between update intervals, we accumulate counts of the number of times that Q occurs in each interval in a daily accumulation vector. Note that regardless of the definition of "appropriate", each time an audit record arrives, one of the values in the daily accumulation vector is incremented by one.

- At update time, the following operations are performed:

  1. The effective n value is aged by the daily aging factor.
  2. The probabilities in the historical profile are converted to counts by multiplication by the newly aged effective n value.
  3. The counts in the daily accumulation vector (accumulated since the last update) are added, interval by interval, to the historical counts.
  4. Today's total count (the sum of the counts in the daily accumulation vector which equals the number of audit record processed) is added to the old effective n to yield a new effective n.
  5. The updated counts in the newly computed historical profile are divided by the new effective n to give the new historical profile (i.e., the new P_m).
  6. The daily accumulation vector is reset to zero.

Computing the Q Statistic for Categorical Measures

Categorical measures are those that involve the names of particular resources being used, for example, the names of files being accessed, terminals being used, and locations from which logons are attempted. The Q statistic for categorical measures is more complex than that for ordinal variables, and involves the specification of additional parameters.

The General Form for the Q Statistic

The general form of the Q statistic for categorical variables is as follows:

$$Q = \sum_{m=1}^{M} \left\{ g(f_m) \times \left[\textstyle\sum_{i \geq 1} W_{m_i} 2^{-rt_i}\right]^{vk} \times \left[2^{-rT_m}\right]^{1-k} \right\}$$

where

- M denotes the total number of categories of the resource that have been used. For example, M unique files may have been accessed. Therefore the first sum extends over the categories of the resource.

- f_m is a measure of the likelihood of the occurrence of the mth category. f_m is not necessarily a probability, although it will be between 0 and 1 in magnitude.

- $g()$ is a functional transform of its argument. This function transforms small values of f_m (which indicate the occurrence of rarely observed categories of the measure being examined) into large values of $g(f_m)$ so that the Q statistic will be large.

- i is an index over all audit records that indicate use of some category of the measure being examined, with $i = 1$ indicating the most recent audit record indicating such use.

- W_{m_i} is an indicator variable that attains the value of 1.0 if the ith audit record indicates the use of the mth category of the measure being examined.

- t_i is the time since the ith audit record. Time may be measured chronologically or as counts of audit records that indicate use of some category of the measure being examined.

- r is decay rate of the statistic Q. For example, the decay rate might be set so that the half-life of Q is on the order of 4 hours or 100 appropriate audit records.

- T_m is the time since the most recent audit record in which the mth category of the resource was used. If the current audit record indicates the use of the mth category of the resource, then $T_m = 0$.

- k and v are parameters, such that k is 0 or 1, and $0 < v \leq 1$. Criteria for selecting values for k and v will be discussed below.

We find it convenient to write Q as follows:

$$Q = \sum_{m=1}^{M} \left\{ g(f_m) \times [N_m]^{vk} \times \left[2^{-rT_m}\right]^{1-k} \right\}$$

where N_m is the exponentially decayed count of the number of audit records that indicate usage of the mth category of the resource. That is,

$$N_m = \sum_{i \geq 1} W_{m_k} 2^{-rt_i}.$$

The Binary, Linear, and Intermediate Forms for the Categorical Q Statistic

If k is set equal to 0, the formula for Q simplifies to

$$Q = \sum_{m=1}^{M} g(f_m) 2^{-rT_m}.$$

We call this the "binary categorical" form for Q. It is affected only by the time interval since the most recent occurrence of the mth category of the resource being measured, and not by how frequently that category has been used in the past. Note that it is relatively easy to update the value of Q if we keep separate data elements containing the most recent value of 2^{-rT_m}. To reduce the number of data elements that the IDES system must store, we specify that a category can be dropped from the summation whenever 2^{-rT_m} becomes less than 0.001, or some other suitably small number.

If k and v are each set equal to 1.0, the formula for Q simplifies to

$$Q = \sum_{m=1}^{M} \{ g(f_m) \times N_m \}$$

In this event, Q has attained what we have called its "linear categorical" form. In its linear categorical variable form, Q is dependent upon the number of times that the mth category of the measure being examined has been used. This contrasts with the binary categorical form for Q, which only depends on the time of occurrence of the most recent usage of the mth category. For example, if a category has an effective count of 9.0, it will add approximately nine times as much to the linear categorical form for Q as it does to the binary categorical form for Q. It is particularly easy to update the value of Q when it is in the linear categorical form. If we let Q_{n+1} denote the value of Q after the $(n+1)$st audit record indicating use of some category of the measure being examined, assume that the $(n+1)$st audit record indicates usage of the mth category of the measure being examined, and let t be the time since the nth audit record, then

$$Q_{n+1} = g(f_m) + 2^{-rt} Q_n$$

This is reminiscent of the updating formula for ordinal measures.

If k is set equal to 1.0, and v is set to a value between 0 and 1, then we observe "partial" counting of the number of records that use the mth category of the resource. The result is a Q value that lies between the binary categorical and linear categorical forms for Q. We call this an intermediate categorical form. For example, consider $v = \frac{1}{2}$. In this case, if a category has an effective count of 9.0, it will add about one third as much to the intermediate categorical form for Q as to the linear categorical form for Q, and about three times as much to intermediate categorical form for Q as to the binary categorical form for Q.

The specification of values for k and v is the fifth decision that must be made in implementing the general statistical framework:

- Decision #5. Determine appropriate values for k and v for any Q statistics based on categorical measures. A value of $k = 0$ corresponds to the "binary categorical" form for Q. Values of $k = 1$ and $v = 1$ correspond to the "linear categorical" form for Q. Values of $k = 1$ and v values between 0 and 1 correspond to intermediate forms for Q.

Specification of the Likelihood of Occurrence Values f_m

The formula for Q requires the specification of the pseudoprobability values f_m. We are currently considering four methods for calculating f_m: (1) probability of occurrence during a fixed chronological time period, (2) relative frequency of occurrence, (3) absolute exponentially weighted interarrival times, and (4) relative exponentially weighted interarrival times. These methods are explained below.

The Fixed Chronological Time Period Method

In the probability of occurrence method for a fixed chronological time period, f_m is an estimate of the probability that the mth category of the measure being examined will occur in a randomly selected time period of a fixed length. The formula for calculating f_m is as follows:

$$f_m = \frac{\sum\limits_{k \geq 1} W_{m_k} 2^{-bk}}{\sum\limits_{k \geq 1} 2^{-bk}}$$

where

- k is an index of time units. For example, if the time unit is a day, then k indexes days. Let $k = 1$ denote the most recent time period.
- W_{m_k} is an indicator function that attains the value of 1 if the mth category of the measure was used in the kth time period, and is 0 otherwise.
- b is the half-life of the data being used to compute f_m. For example, if the fixed time period is 1 day, then a half-life of 60 days would correspond to a "b" value of 0.0116. We recommend that b be set so that the half-life of f_m is the same as the half-life of the frequency distribution of the Q statistic.

We do not currently recommend the use of this method, because it imposes a fixed time constraint (e.g., 1 day) where there is no natural time constraint. We note that the values for f_m are absolute in the sense that they do not depend on how frequently or infrequently other categories of the measure occur. That is, all f_m values could be equal to 1.0, or equal to 0.0.

The Relative Frequency of Occurrence Method

In the relative frequency of occurrence method, f_m is an estimate of the relative number of times that the mth category of the measure occurs. The formula for calculating P_m is as follows:

$$f_m = \frac{\sum\limits_{k \geq 1} W_{m_k} 2^{-bt_k}}{\sum\limits_{k \geq 1} 2^{-bt_k}}$$

where

- k is an index of audit records that indicate the use of some category of the measure, with $k = 1$ denoting the most recent such audit record.
- W_{m_k} is an indicator function that attains the value of 1 if the mth category of the measure was used on the kth audit record, and is 0 otherwise.
- t_k is the time that has elapsed between the kth and most recent audit record indicating the use of some category of the measure. Time may be measured either chronologically, or by a counting variable that is incremented by 1 at the arrival of a new audit record indicating the use of some category of the measure being examined.
- b is the decay rate of the data being used to compute f_m. For example, if t were measured in days, a half-life of 60 days would correspond to a "b" value of 0.0116.

We are currently using the relative frequency of occurrence method when Q is specified to be in its linear categorical form.

To reduce the computational burden on the IDES system, we simplify the procedure for calculating f_m using the relative frequency of occurrence method as follows:

- A set of weighted counts is maintained of the number of audit records that have been processed in the past that indicated that category m of the measure occurred. Let R_m be the weighted number of times that the mth category of the measure has been observed on previous days.
- A set of unweighted counts is maintained of the number of audit records in the current day that indicate the usage of category m of the measure. Let U_m be the number of audit records on the current day indicating the usage of category m.
- At the end of the day (or other appropriate update time), the values of R_m are updated as follows:

  $$NewR_m = U_m + 2^{-bt} \times (OldR_m)$$

  where b is the decay rate of the R_m counts (usually on the order of 30 to 60 days) and t is the time that has elapsed since the last update (e.g., usually 1 day).
- The values for f_m are calculated at update time as

  $$f_m = \frac{NewR_m}{N}$$

  where N is the sum of the R_m over all categories. This value of f_m is used until the next update time.

This updating procedure is essentially identical to that used to maintain the frequency distribution for Q.

The Absolute Exponentially Weighted Interarrival Time Method

In the absolute variant of the exponentially weighted interarrival time method, f_m is given by the following formula:

$$f_m = 2^{-cA_m}$$

where

- A_m is the exponentially weighted interarrival time for audit records indicating the use of the mth category of the measure being examined. For example, if time is measured in hours, then an A_m value of 3.4 would denote that an audit record indicating the use of the mth category of the measure being examined arrives on average once every 3.4 hours.

- c is a scaling factor allowing the IDES analyst to vary the relationship between interarrival time and the magnitude of f_m. For example, if t is measured in days and $c = 1$ then an interarrival time of 1.0 days would correspond to an f_m value of 0.5, whereas if $c = 0.1$ then an interarrival time of 10 days would correspond to an f_m value of 0.5. By setting c properly, the value of f_m can be made to approximate the values obtained from the daily probability of occurrence method. For example, consider six categories of the measure being examined that arrive on average 0.1, 1, 2, 3, 4, and 10 days apart. Under the daily probability of occurrence method their f_m values would be approximately 1.0, 1.0, 0.5, 0.33, 0.25, and 0.10. If we set $c = 0.5$, then the corresponding f_m values under the first variant of the weighted interarrival time method would be 0.97, 0.71, 0.50, 0.35, 0.25, and 0.03. By selecting different values for c, we can also approximate hourly or weekly probabilities of occurrence.

The average interarrival time A_m is calculated as follows:

$$A_m = \frac{\sum_{k\geq 1} (t_{m,k+1} - t_{m,k})\, 2^{-bt_{m,k}}}{\sum_{k\geq 1} 2^{-bt_{m,k}}}$$

where

- k is an index denoting all audit records that indicate usage of the mth category of the measure being examined. The most recent such audit record is denoted $k = 1$.

- $t_{m,k}$ is the time that elapsed between the kth audit record and the most recent audit record in the summation. Time may be measured chronologically or as counts of appropriate audit records.

- b is the decay rate of the data used in computing A_m. We recommend a half-life equal for A_m equal to the half-life used in computing the frequency distribution of Q.

We note that if audit records indicating the use of the mth category of the measure arrive infrequently, then A_m will be large and f_m will be small. Furthermore f_m is always between 0 and 1, so it is scaled the same as a probability, even though it is not a probability.

The Relative Exponentially Weighted Interarrival Time Method

In the relative exponentially weighted interarrival time method, f_m is given by the following formula:

$$f_m = \frac{min_k(A_k)}{A_m}$$

where the minimum function extends over all categories of the measure being examined. If m is the category that arrives the most frequently, then $f_m = 1.0$. If the interarrival time for the mth category is five times as long as the interarrival time for the category that arrives most frequently, then $f_m = 0.2$. Note that this variant of the exponentially weighted interarrival time method yields results that are comparable to the relative frequency of occurrence method.

As discussed above, in calculating the values for f_m the IDES analyst must make the following decision:

- Decision #6. Determine whether the values for f_m shall be calculated using the probability of occurrence method for fixed time periods, the relative frequency method, the absolute exponentially weighted interarrival time method, or the relative exponentially weighted interarrival time method. If the first method is chosen, determine the appropriate time period. If the third method is chosen determine the appropriate value for the scaling parameter used to relate interarrival times to values for f_m.

Specification of the Function $g()$

As seen earlier, the formula for calculating a Q statistic based on a categorical measure involves the function $g()$, which is applied to the likelihood value f_m. This function allows the IDES/STAT system to transform small values of f_m (which indicate the occurrence of an infrequently seen category) into large values of $g(f_m)$, and consequently into large values for Q.

The general functional form for $g()$ that we are examining is as follows:

$$g(x) = [-\log_2(x)]^y$$

where $\log_2(x)$ is the log-based 2 of x. The parameter y is a number greater than 1.0. We are currently examining the effect of choosing $y = 1$ or $y = 2$. The next section describes the rationale for chosing between these two values of y. We record the necessary decision:

- Decision #7. Complete the functional specification of $g()$ by selecting a value for the exponent y.

Decision Options Affecting the Q Statistic

The seven decisions listed above determine the specific form for the Q statistic. In this section we discuss some of the factors that should be considered in making those decisions.

How Probabilities of Different Magnitudes Affect the Q Statistic

In making decision #7 (concerning the exponent y in the functional form for the $g()$ function) we need to consider how probabilities of different magnitudes for various categories affect the Q statistic.

Consider a categorical measure for which we have calculated 11 f_m values using the relative frequency of occurrence method. Let the f_m values for the first nine categories be denoted f_1 through f_9, and suppose that each of these values is equal to 0.10. Further suppose that $f_{10} = 0.01$ and that $f_{11} = 0.001$.

If the exponent of the $g()$ function is 1.0, then $g(f) = -\log_2(f)$. In this case $g(f_i) = 3.3$ for $1 \leq i \leq 9$, $g(f_{10}) = 6.6$, and $g(f_{11}) = 9.9$. Thus every time that category 1 through 9 occurs, Q is incremented by 3.3, whereas if category 10 or 11 occurs, Q is incremented by 6.6 and 9.9 respectively. If categories 1 and 2 each occur once, Q is incremented by 6.6, the same as if category 10 occurs by itself. That is, accessing two different files that each occur one-tenth of the time counts equally to accessing one file that occurs one-hundredth of the time. If categories 1, 2, and 3 each occur once, Q is incremented by 9.9, the same as if category 11 occurs once by itself. That is, accessing three different files that each occur one-tenth of the time counts equally to accessing one file that occurs one-thousandth of the time.

If the exponent of the $g()$ function is 2.0, then $g(f) = [-\log_2(f)]^2$. In this case $g(f_i) = 11.0$ for $1 \leq i \leq 9$, $g(f_{10}) = 44.1$, and $g(f_{11}) = 99.3$. Thus every time that category 1 through 9 occurs, Q is incremented by 11.0, whereas if category 10 or 11 occurs, Q is incremented by 44.1 and 99.3 respectively. If categories 1, 2, 3, and 4 each occur once, Q is incremented by 44.1, the same as if category 10 occurs by itself. That is, accessing four different files that each occur one-tenth of the time counts equally to accessing one file that occurs one-hundredth of the time. If each of categories 1 through 9 is accessed once, Q will be incremented by 99.3, the same as if category 11 occurs by itself.

In the near future we will be conducting experiments to determine the appropriate magnitude for the exponent of the $g()$ function. We currently recommend that this exponent be larger than 1.0. It is our subjective judgement that the level of concern that should be raised by using a file that is only accessed one time per 1000 file accesses should be considerably higher than the level of concern that should be raised by accessing three files, each of which is accessed one time in 10 file accesses. We believe that the former occurrence is more indicative of an intrusion attempt than the latter occurrence. On this basis, we currently recommend an exponent of 2 or even 3.

How Multiple Occurrences Affect the Q Statistic

In making decision #5 (concerning the exponents k and v in the Q statistic for categorical measures) we need to consider how multiple occurrences of a category affect the Q statistic.

Continuing with the example from the previous section, consider a categorical measure for which we have calculated 11 f_m values using the relative frequency of occurrence method. Let the f_m values for the first nine categories be denoted f_1 through f_9, and suppose that each of these values is equal to 0.10. Further suppose that $f_{10} = 0.01$ and that $f_{11} = 0.001$.

If the exponent k for the Q statistic has been set equal to 0.0, then Q counts only the most recent occurrence of each category. That is, Q is incremented by approximately the same amount whether the category occurs once or multiple times. For example, Q would be incremented by $g(.01)$ if the last nine audit records were all occurrences of category 1, but would be incremented by $9 \times g(.01)$ if the last nine audit records were single occurrences of categories 1 through 9.

If the exponent k for the Q statistic has been set equal to 1.0, and v has been set equal to 1.0, then Q counts number of occurrences of each category. That is, Q is incremented by the same amount whether the category 1 occurs nine times in the last nine audit records, or categories 1 through 9 occur once each in those same audit records.

If the exponent k for the Q statistic has been set equal to 1.0, and v has been set equal to 0.5, then Q counts the square root of the number of occurrences of each category. For example, if it has been a while since any of categories 1 through 9 have been used, then Q is incremented by approximately the same amount whether category 1 occurs nine times in the last nine audit records, or categories 1, 2, and 3 occur once each in the last three audit records.

In the near future we will be conducting experiments to determine the appropriate values for k and v. We currently believe that if a single category occurs n times, it should count more than a single occurrence of that category, but should not count as much as n occurrences of equally likely, but different categories. This suggests that k should equal 1.0 and v should be an intermediate value such as 0.5.

Whether Time Should Be Chronological or Count Related

In making decision #2 (whether time should be measured chronologically or should be incremented by 1 each time an appropriate audit record is processed) we need to consider both the stability (e.g., coefficient of variation) of the Q statistic and how rapidly it can respond to intrusions. For the purposes of this section, let us assume that Q either is based on an ordinal measure, or is based on a categorical measure with $k = 1$ and $v = 1$.

If time is measured chronologically, the principal strength of Q is that Q tends to concentrate on activity in the past few hours (assuming a half-life in the range of 1 to 4 hours). That is, behavior more than a half-day in the past will have very little effect on Q. A related strength is that Q will tend to be sensitive to large amounts of audit record activity that occur in a period of time that is short relative to the half-life. For example, if a user typically generates only 100 or so audit records an hour, and the half-life is 4 hours, then it might be possible to detect increases to 200 audit records an hour for an hour or two, or to larger increases for shorter periods of time.

If time is measured chronologically, a principal weakness of Q is instability because Q depends on a variable number of audit records. The value for Q will tend to be low at the beginning of the work day, because its value has been decaying overnight. The value for Q will tend to be high at the end of the workday (or during very busy times during the middle of the workday) because Q will then be based on many audit records whose effect has not had an opportunity to decay. That is, the effective number of audit records on which Q is based will tend to be low in the morning and high in the afternoon. This variability in the effective number of audit records spreads out the frequency distribution of Q, and results in a wider range of Q values being classified as normal. It also concentrates Q values early in the morning in the lower part of the distribution, so that even a reasonable amount of intrusive behavior may not raise Q to a sufficiently high level to be declared abnormal. Thus, it will be relatively easy to conduct an undetected intrusion if it is scheduled for, say, 5 a.m. (or any time during the day when the user is on vacation or on business travel). A related shortcoming is that Q will tend to be insensitive to intrusions that do not generate substantial numbers of audit records. When an intrusion is detected it may take many hours for the Q statistic to return to normal through the usual decay process.

If time is measured in terms of appropriate audit record counts, a principal strength of Q is stability. Q becomes based on what is essentially a fixed number of audit records. This stabilizes the distribution of Q and minimizes its coefficient of variation, which should make it easier to detect intrusions. Also, time is effectively suspended overnight (or in any periods of low audit activity) and is resumed when audit activity resumes, so that the Q statistic will not be smaller than usual in the morning. Another strength is that Q will tend to be more sensitive to short intrusions. For example, consider a short intrusion that generates only, say, 50 audit records, and suppose that the half-life of Q were 100 appropriate audit records. Fifty audit records is a respectable percentage (approximately 34%) of the total effective number of audit records (e.g., 145) in the Q statistic. This might be enough to force Q toward large s-values. A related advantage is that the Q statistic based on appropriate record counts may recover more rapidly from an intrusion attempt than one based on chronological time. For example, after 150 or so normal audit records are generated, the intrusion-generated audit records will have been sufficiently decayed so that the IDES/STAT system ceases to issue intrusion warnings.

If time is measured in terms of appropriate audit record counts, a principal weakness of Q is the potentially extended or compressed time frame of past activity that affects the value for Q. For example, if Q is based on a half-life of 100 audit records of network activity, and the user generates only a few network access audit records each week, then the Q statistic may be summarizing behavior over the past few months. On the other hand, if the user is rapidly generating audit records, then the Q statistic may be summarizing activity over only the past few minutes. Furthermore, different Q statistics, based on different measures, may have dramatically different half-lives as measured in clock time, with some summarizing behavior over many weeks and others over a few minutes. This problem might be ameliorated by having measures of specific decay factors based on the number of audit records generated per day by a user, so that the half-life is the number of audit records generated in 1 or 2 hours of user activity in the middle of the day. A second disadvantage of measuring time in terms of appropriate audit records is that Q will not be sensitive to intrusions that increase the number of audit records generated but where the individual audit records appear to be normal. For example, suppose that a user normally generates an average of 100 I/O audit records per day with an average of 12 read/writes per I/O audit record. The user's account is broken into, and the perpetrator generates 1000 I/O audit records in one-half hour, with an average of 12 read/writes per audit record. In this case the Q statistic (with time measured in audit record counts) will not vary from its historical mean value and the intrusion would not be detected. For this reason, whenever time is measured in audit record counts, there should be at least one additional measure, i.e., the number of audit records, measured in chronological time.

We will be conducting experiments to determine whether time should be measured chronologically or in terms of appropriate audit record counts. Currently we believe that measurement in terms of appropriate audit record counts is preferable, but to a large extent this is still an open issue.

Acknowledgements

The authors wish to acknowledge the contributions of the three anonymous reviewers, who pointed out the need for a high-level roadmap to the sections of this paper, recommended the inclusion of the graph, and located grammatical and typographical errors. Special acknowledgement is given to Teresa F. Lunt for the IDES project leadership and support, and to Ann Tamaru and Fred Gilham for the system implementation.

References

[1] T.F.Lunt, A.Tamaru, F.Gilham, R.Jagganathan, C.Jalali, H.S.Javitz, A.Valdes, and P.G.Neumann, *A Real-Time Intrusion Detection Expert System*, SRI Computer Science Laboratory Technical Report, SRI-CSL-90-05, June 1990.