UDC 004.7.056.5

V. TKACH
Blekinge Institute of Technology, Karlskrona, Sweden; National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute," Kyiv, Ukraine,
e-mail: *volodymyr.tkach@bth.se, vntkach@gmail.com.*

A. KUDIN
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute;" National Bank of Ukraine, Kyiv, Ukraine,
e-mail: *pplayshner@gmail.com.*

V. ZADIRAKA
V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine, Kyiv, Ukraine,
e-mail: *zvk140@ukr.net.*

I. SHVIDCHENKO
V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine, Kyiv, Ukraine,
e-mail: *inetsheva@gmail.com.*

SIGNATURELESS ANOMALOUS BEHAVIOR DETECTION IN INFORMATION SYSTEMS

Abstract. The early detection of cyber threats with cyber-attacks adapted to the nature of information systems is a crucial cybersecurity problem. This problem and the task of recognizing normal and abnormal states and behavior of various processes in information systems are closely related. An additional condition is often the absence of templates, signatures, or rules of normal behavior that would allow using existing statistical or other known methods of data analysis. We analyze the existing and propose a new method for detecting abnormal behavior without the use of signatures based on the finite state machine (FSM) model and the Security Information and Events Management (SIEM) system.

Keywords: anomaly detection, finite state machine, SIEM, time-series, cybersecurity.

INTRODUCTION

The increasing impact of cyber threats on critical infrastructure is attributable to the simultaneous rise of information flows and infrastructure complexity. Higher complexity often results in higher functionality, which, in turn, reduces the security level of any system, as has been established [1].

As the number of cyber-attacks continues to rise [2], safeguarding critical infrastructure objects [3], which are mostly state institutions, becomes the top priority for cyber defense. It is widely recognized that modern cyber-attacks are becoming increasingly sophisticated and causing significant damage to their targets. In recent times, several highly sophisticated supply chain attacks have been witnessed, which may have had a multi-step history and could have been detected at early stages if a method for detecting anomalous behavior was available.

A significant issue in cyber defense is the absence of effective mechanisms for detecting and preventing attacks unless specific attack patterns or even signatures are identified. As a result, the development of pre-detection and prevention mechanisms for cyber threats has become crucial, particularly in cases where there is insufficient information about potential threats and their signatures. Such mechanism should be based on identifying anomalies in user behavior, where an anomaly refers to unusual user behavior that deviates from what is expected rather than the "normal" behavior. It is important to note that what is considered normal can be subjective and varies across different systems. Forecasting will be the primary method for identifying deviations from expected behavior.

To sum up the above-mentioned, modern cybersecurity is defined on the one hand by the changeable character of threats, and on the other hand, by adaptive change of the very state of specific information system in cyberspace, which is considered as safe. Traditional information security technologies are built in the

following way. At first, we explore the vulnerabilities, and then we identify the threats on their basis. Such technologies cannot work in our situation in spite of using the best methods for vulnerabilities detection (see example [4]). Moreover, intrusion detection systems built on signature or pattern of cyberattacks (including patterns changeable in time [5]) do not solve this problem completely, too. For that, we need a lot of time for signature or pattern forming (by neural network for example). So, the investigation of practically effective methods for detecting the system's anomaly behavior is becoming urgent. It is especially important for the system where the very meaning of its "abnormal state" is changing quickly enough.

In order to develop such a mechanism, it is imperative to establish an uninterrupted stream of user behavior indicators that serves as the foundation for identifying anomalies.

The paper is divided into four sections according to the paper topic. The anomalies section provides a definition and comprehensive understanding of different types of anomalies, as well as existing methods for detecting them. The section on SIEM systems describes the fundamental process of collecting raw data for detecting anomalies, including data description, examples, formalization of the data model, and identification of features in logs. The section on finite state machines explains the finite state machine model and covers inputs, outputs, control signals, and sets of states that explain the transitions between them. This section also presents a method for detecting anomalous behavior without using signatures, including the likelihood component of a finite state machine model. Finally, the conclusions section summarizes the paper and discusses the method's application, as well as potential future improvements.

ANOMALIES

The process of extracting valuable insights from large datasets is achieved through data mining. As the volume of data continues to grow and the significance of analysis results increases, the need for identifying anomalies becomes more critical. Anomaly detection involves identifying unexpected values or deviations from the normal behavior of a system. The terms "anomaly", "outlier", "error", and "exception" are used interchangeably in this paper to refer to such deviations that can occur in data of varying structures and nature, whether resulting from technical malfunctions, accidents, intentional disruptions, or other factors. Various techniques and algorithms have been developed for identifying anomalies in different types of data. In this section, we aim to review the most well-known approaches to anomaly detection.

In the following text, we will refer to the paper [6], as it is necessary for a complete understanding of the nature of anomalies and their classification. Anomalies in data can generally be classified into one of three main types, namely point anomalies, contextual anomalies, and collective anomalies.

Point anomalies refer to situations in which a single data instance can be considered anomalous compared to the remaining data. Figure 1, *a* shows an example of a point anomaly, where the point 350 is identified as an abnormality in a normal random walk type chart. This type of anomaly is widely recognized, and many existing methods are designed to detect point anomalies.

Contextual anomalies are identified when a data instance is considered anomalous only in a particular context or condition, also known as conditional anomalies. Detection of such anomalies requires the selection of appropriate contextual and behavioral attributes.

- Contextual attributes refer to the features that define the context or environment of each data instance. For instance, in time series data, the contextual attribute is typically the timestamp that determines the position of the instance in the entire sequence. Alternatively, a contextual attribute can represent a spatial position or a more complex combination of properties that define the context.
- Behavioral attributes refer to non-contextual properties that are specific to a data instance. The abnormality of a data instance is determined by the values

of its behavioral attributes in the specific context. It should be noted that a data instance may be considered a contextual anomaly under certain conditions, but with the same behavioral attributes, it may be considered normal in a different context. For instance, in Fig. 1, b, point 950 displays an anomaly. The ability to distinguish between contextual and behavioral attributes is crucial when detecting contextual anomalies.

Collective anomalies occur when a sequence of related instances of data (such as a time series section) is anomalous with respect to an entire data set or its considerable part. A single instance of data in this sequence may not be a deviation, but the co-occurrence of such instances is a collective anomaly. In Fig. 1, *c*, points between 600 and 750 are a subject to collective anomaly.

In addition, while point or contextual anomalies can be observed in any data set, collective anomalies are observed only in those where the data are interconnected. It is also worth noting that point or collective anomalies can also be contextual.

There are several options for classifying existing methods of finding anomalies [6]. In this paper, we will consider two types of division: the mode of recognition and the method of implementation.

Depending on the algorithm, the result of the anomaly identification system may be either to label the data instance as abnormal or to assess the level of likelihood that the instance is abnormal.

The process of detecting anomalies can be performed for data of different formats:

- data flow (real-time operation);
- data archive.

The task of identifying anomalies often requires a labeled dataset that describes the system. Each instance in the dataset is labeled as either normal or abnormal, with many instances belonging to the same class. However, creating such a labeled dataset can be a manual and expensive process, and in some cases, it may be impossible to obtain instances of the anomalous class due to the lack of data on potential system deviations. Furthermore, labels may be unavailable for both classes. Anomaly detection methods can be performed in one of three modes, depending on which data classes are utilized to implement the algorithm.

Supervised anomaly detection. In this technique, a training dataset containing both normal and anomalous instances that represent the system is required. The algorithm operates in two stages: training and detection. During the training stage, a model is constructed based on the available labeled data, which is later used to compare with unlabeled instances during detection. It is usually assumed that the statistical characteristics of the data do not change over time. Otherwise, the classifier needs to be updated accordingly [7].

The main difficulty of algorithms that work in the mode of recognition with the supervisor is the formation of data for learning. Often an anomalous class is represented by a much smaller number of instances than a normal one, which can lead to inaccuracies in the resulting model. In such cases, artificial generation of anomalies is used.

Semi-supervised anomaly detection. In this approach, the initial data only consists of instances belonging to the normal class. The system is trained on this class and can determine the membership of new data to this class, and therefore identify anomalies by recognizing data that do not belong to the normal class. Algorithms that work in a recognition mode with partial supervision do not require information about the anomalous class of instances, making them more widely used and enabling the detection of anomalies in the absence of pre-determined information about them.

Unsupervised anomaly detection. This technique is employed when there is no prior knowledge about the data. Non-supervised recognition algorithms operate on the assumption that anomalous instances are significantly rarer than normal ones. The data is analyzed, and the most extreme instances are identified as anomalies. In order to apply this method, the entire dataset needs to be available, otherwise real-time analysis is not possible.

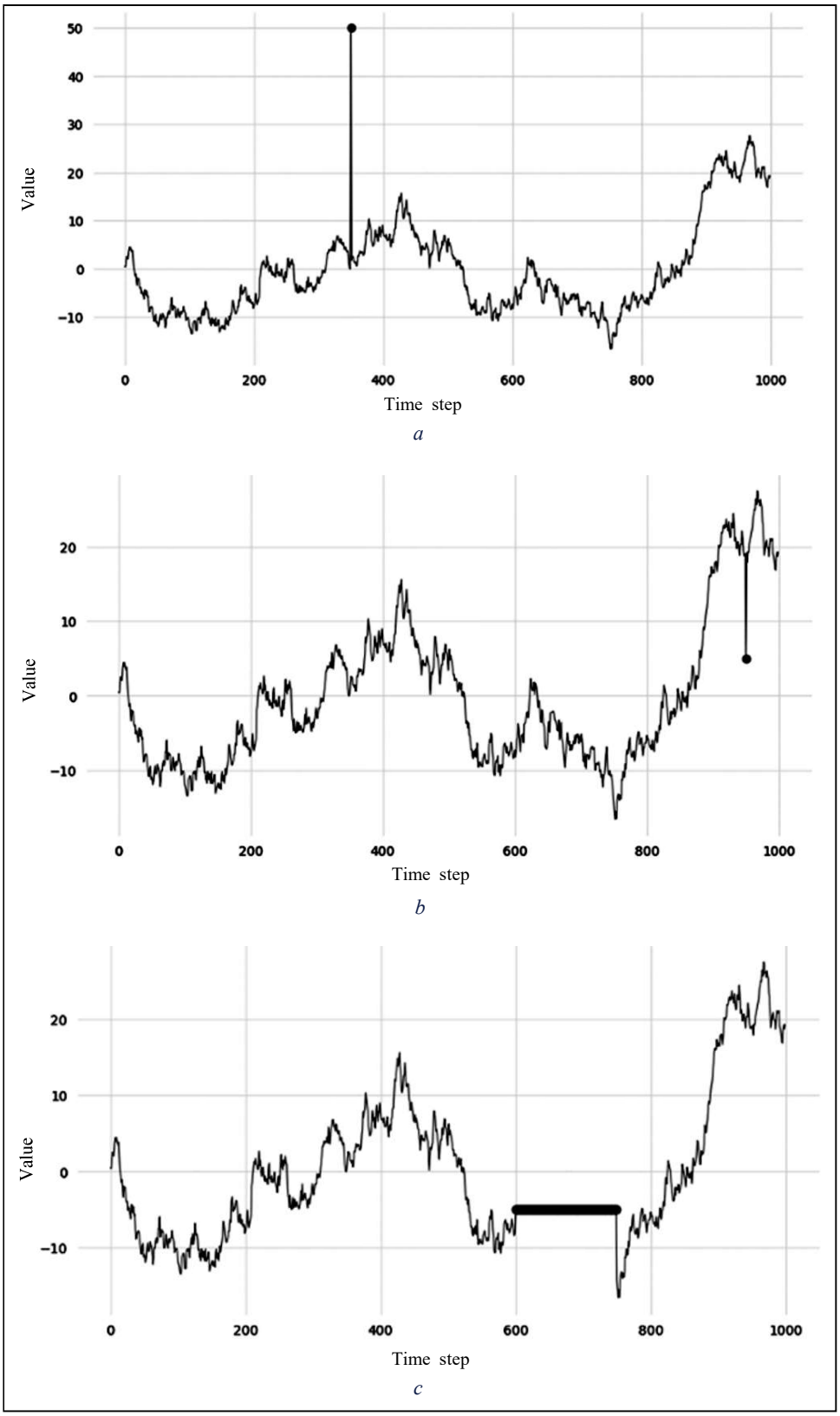

Fig. 1. Main types of anomalies recognized in timer-series analysis: point anomaly (*a*), contextual anomaly (*b*), collective anomaly (*c*). The units of measurement in this chart are arbitrary

Further, an overview of the existing methods for anomaly detection will be provided.

Classification. The implementation of this method is based on the assumption that the normal behavior of the system can be determined by one or more classes. Thus, an instance that does not belong to any of the classes is a deviation. The search for anomalies takes place in two stages: learning and recognition. The classifier is studied on an array of marked data, then the affiliation to one of the known classes is determined. Otherwise, the instance is referred to as an anomaly.

Anomaly detection can be implemented through classification using several mechanisms, including neural networks, Bayesian networks, reference vectors, and rule-based methods. Among these, neural networks are the most commonly used. The process of detecting anomalies through neural networks involves two stages: first, the network learns to recognize normal behavior classes using a training set, and second, each instance is fed into the network as input to identify anomalous behavior. Such systems can detect one or multiple classes of normal behavior.

Replicative neural networks are used to find anomalies by recognizing only one class [8]. The technology of neural networks of deep learning (Deep Learning), which has become widespread, is also successfully used to solve this problem [9].

Bayesian network is a graphical model that reflects the probability dependencies of many variables and allows you to draw a probability conclusion with these variables. It consists of two main parts: a graphical structure that defines a set of dependencies and independencies in a set of random variables representing the subjects of the subject area, and a set of probability distributions that determine the strength of dependency relations encoded in the graphical structure. Thus, the use of the Bayesian network in the identification of anomalies is to estimate the probability of observing one of the normal or anomalous classes. The simplest implementation of this approach is the Naive Bayes Approach [10].

The Support Vector Machine (SVM) method is specifically designed for detecting anomalies in systems where normal behavior is represented by a single class. The method constructs a boundary around the region where instances of normal data are located. Subsequently, each data point is evaluated to determine if it lies within the boundary. If a point is found to be outside the boundary, it is classified as an anomaly. This method for detecting anomalies is based on generating rules that correspond to normal behavior of the system. An instance that does not comply with these rules is recognized as an anomaly. This algorithm involves two steps: step 1 involves learning the rules using a specific algorithm, such as RIPPER, Decision Trees, etc. Each rule is assigned a value that is proportional to the ratio of training instances classified by the rule to the total number of training instances covered by the rule. Step 2 involves searching for the best fitting rule for each test instance. The system can recognize both one and multiple classes of behavior.

One subtype of rule-based systems is fuzzy logic systems. They are used when the line between normal and abnormal system behavior is not strictly determined. Each instance is an anomaly to some extent away from the center of mass of the normal interval.

Clustering. This technique involves grouping similar instances into clusters and does not require knowledge of the properties of possible deviations. Detection of anomalies can be based on the following assumption:

- Normal instances of data belong to a data cluster, while anomalies do not belong to any of the clusters. However, this wording may raise the problem of defining clear cluster boundaries.

Thus, we have the following assumptions:

- Normal data is closer to the center of the cluster, and abnormal — much further.

In the case where anomalous instances are not single, they can also form clusters. Thus, their detection is based on the following assumption:

- Normal data form large dense clusters, while abnormal — small and scattered.

Statistical analysis. In this approach, the process is analyzed to create a profile or model that represents normal behavior. The model is then compared to the actual behavior of the system. If the deviation between the two, as determined by a designated anomaly function, exceeds a pre-set threshold, the system is flagged as anomalous. The underlying assumption is that the normal behavior of the system will have a high probability of occurrence, while anomalies will have a low probability.

This approach is advantageous because it does not require prior knowledge about the type of anomaly. However, it can be challenging to accurately determine the statistical distribution and threshold value for the system [11].

Methods of statistical analysis are divided in two main groups:

- Parametric methods. It is assumed that the normal data is generated by a parametric distribution with the parameters θ and the probability density function $P(x, \theta)$, where x is an observation. The anomaly is an inverse distribution function. These methods are often based on the Gaussian or regression model, as well as their combinations.
- Non-parametric methods. It is assumed that the structure of the model is not determined a priori; instead, it is defined from the data provided and includes methods based on histograms or kernel functions.

The basic algorithm for finding anomalies using histograms includes two stages. At the first stage, there is a construction of the histogram based on various values of the chosen characteristic for copies of training data. In the second stage, each of the studied specimens is determined to belong to one of the columns of the histogram. Instances that do not belong to any of the columns are marked as abnormal.

Anomaly recognition based on the kernel function is similar to parametric methods except for the method of estimating the probability density.

Algorithm of the nearest neighbor. To use this technique, it is necessary to define the concept of distance (degree of similarity) between objects. An example is the Euclidean distance.

The two main approaches are based on the following assumptions:

• Distance to the k^{th} nearest neighbor. To implement this approach, the distance to the nearest object is determined for each tested instance of the class. The specimen that is the outlier is furthest from the nearest neighbor.

• The use of relative density is based on the estimation of the neighborhood density of each data instance. An instance that is in a low-density environment is evaluated as abnormal, while an instance in a high-density neighborhood is evaluated as normal. For this instance, the distance to its k^{th} nearest neighbor is equivalent to the radius of the hypersphere centered in this instance and contains k other instances.

Spectral methods. Spectral methods aim to approximate the data using a combination of attributes that capture the most important variability in the data. The underlying assumption is that the data can be represented in a lower-dimensional subspace, where normal and anomalous behavior can be distinguished more effectively. These methods are often used in conjunction with other data preprocessing algorithms to improve the accuracy of anomaly detection.

Hybrid methods. Hybrid anomaly recognition techniques allow you to combine the advantages of different approaches. In this case, different techniques can be used both sequentially and in parallel to achieve average results.

Behavior profiles, metrics and statistical approaches. The activity profile characterizes the behavior of a certain subject (or set of subjects) in relation to a certain object (or its set), thereby serving as a signature or description of normal activity for the respective subject (subjects) and object (about objects). The observed behavior is characterized in terms of statistical metrics and models. Metric is a random variable x, which is a quantitative measure accumulated over time.

The period can be a fixed time interval (minute, hour, day, week, etc.) or the time between two audit-related events (i.e., between entry and exit, program initiation and

program termination, file opening and closing, etc.). Observations (sample scores) x_i and x obtained from audit records are used in conjunction with a statistical model to determine whether a new observation is abnormal. The statistical model does not assume assumptions about the main distribution x; all knowledge about x is obtained from observations. Before describing the structure, generation, and application of profiles, we will first discuss statistics and models.

We can define three types of indicators:

- Event counter: x is the number of audit records that satisfy some property that occurs during the period (each audit record corresponds to an event). Examples are the number of logins per hour, the number of times a command is executed during a login session, and the number of password failures per minute.
- Interval timer: x is a duration of time between two connected events; that is, the difference between the timestamps in the relevant audit records. An example is a length of time between successive logins to an account.
- Resource measurement: x is a measure of resources consumed by some actions during the period specified in the field "Resource use" of audit records. Examples are the total number of pages printed by a user per day and the total amount of CPU time consumed by a program during a single run. Note that the resource measurement in our intrusion detection model is implemented as an event counter or interval timer in the target system. For example, the number of pages printed during an input session is implemented in the target system as an event counter that counts the number of print events between input and output; CPU time consumed by the program as an interval timer that runs between the beginning and end of the program. Thus, although event counters and interval timers measure events at the audit-record level, resource measures obtain data from target system events that occur below the audit audits. The field of use of audit record resources thus provides a means of reducing data so that fewer events need to be clearly recorded in audit records.

After determining the metric for the random variable x and n of observations $x_1 \ldots x_n$, the purpose of the statistical model x is to determine whether the new observation x_{n+1} is abnormal with respect to previous observations. The following models may be used in intrusion detection systems:

Operational model. This model is based on the operational assumption that the anomaly can be solved by comparing the new observation x with fixed limits. Although the previous sampling points for x are not used, it is likely that the boundaries are determined from previous observations of one type of variable. The operating model is most applicable to metrics, where experience shows that certain values are often associated with penetrations. An example is an event counter for the number of password failures over a short period of time, when more than 10, say, involve an intrusion attempt.

Standard and mean deviation model. This model is based on the assumption that all we know about $x_1 \ldots x_n$, is the mean and standard deviation determined from its first two points:

$$sum = x_1 + \ldots + x_n$$
$$sumsquares = x_1^2 + \ldots + x_n^2$$
$$mean = sum / n$$
$$stdev = \sqrt{\left[sumsquares \Big/ (n+1) - mean^2 \right]}.$$

A new observation x_{n+1} is defined as abnormal if it goes beyond the confidence interval, which are d standard deviations from the mean for some parameter d: $mean \pm d \cdot stdev$.

In a case of Chebyshev's inequality, the probability that the value falls outside this interval is not more than $1 / d^2$; for $d = 4$, for example, this is a maximum of 0.0625. It should be noted that zero-events must be included so as not to shift data.

This model is commonly used for monitoring event counters, interval timers, and resource events accumulated over a specific time interval or between two related events. It offers two key benefits over the traditional model. Firstly, it does not require prior knowledge of normal activities to establish restrictions; instead, it learns what is normal activity from its observations, and the confidence intervals automatically reflect this increased knowledge. Secondly, since the confidence intervals depend on the observed data, what is considered normal for one user may differ significantly from another.

A slight variation of this model involves assigning higher weights to recent observations, resulting in the heavier observations being placed at the end of the calculation.

Multivariate model. This model is similar to the model of mean and standard deviations, except that it is based on correlations between two or more indicators. This model would be useful if experimental data shows that better discriminant power can be obtained through a combination of related measures, rather than individually, such as CPU time and 1/0 units used by the program, input frequency, and elapsed session time (which can be inversely related).

Markov process model. This model, which applies only to event counters, considers each individual event type (audit record) as a state variable and uses the state transition matrix to characterize the transition frequencies between states (not just the frequencies of individual states, i.e., audit records taken separately). A new observation is defined as abnormal if its probability is determined by the previous state and the transition matrix is too low. This model can be useful for viewing transitions between specific commands where command sequences are important.

Time series model. This model, which uses an interval timer together with an event counter or resource measurement, takes into account the order and time of interaction of observations $x_1, \dots, x_n$, as well as their values. A new observation is abnormal if the probability of its occurring at that time is too low. The time series has the advantage of measuring trends in behavior over time and identifying gradual but significant changes in behaviour. However, the disadvantage is that it is more time-consuming comparing to the mean and standard deviation calculations.

Other statistical models can be considered, for example, models that use more than the first two points, but less than the full set of values.

Profile components. The activity profile contains information that identifies the statistical model and metrics of the random variable, as well as a set of classroom events measured by the variable. The structure of the profile contains 10 components, the first 7 of which do not depend on the specific objects and objects being measured:

<Variable Name, Action Template, Exception Template, Resource Usage Template, Period, Variable Type, Threshold, Subject Template, Object Template, Value>.

Subject and object independent components are as follows:

- Variable name: Variable name;
- Action template: A template that corresponds to zero or more actions in audit records, such as “login,” “read,” and “execute”;
- Exception Template: A template that corresponds to the Audit Record Exception-Condition field;
- Resource Usage Template: A pattern that matches in the Resource Usage field of the audit record;
- Period: time interval for measurement, for example, day, hour, minute (expressed in tens of clock units). This component is invalid if there is no fixed time interval; the period is the duration of the activity;
- Variable type: The name of an abstract data type that defines a specific type of metric and statistical model, such as an event counter with a mean and standard deviation model;

- Threshold: parameter defining the boundary used in the statistical test to determine the anomaly. This field and its interpretation are determined by a statistical model (Variable type). For the operating model, this is the upper (and possibly lower) limit of the observation value; for the mean and standard deviation model, this is the number of standard deviations from the mean.

The subject and the object dependent components are as follows:

- Subject Template: A template that corresponds to the Subject field in the audit records;
- Object Template: A template that corresponds to the Object field in the audit records;
- Values: the value of the current (most recent) observation and the parameters used by the statistical model to represent the distribution of previous values. For the model of mean and standard deviations, these parameters are the calculation, the sum, and the sum of squares (the first two points). The operating model does not require parameters.

A profile is uniquely identified by a variable name, an object template, and a template object. All profile components are invariant except the value.

SIEM SYSTEMS

One of the most popular approaches to the real-time security perimeter data analysis is ongoing analysis of events within any system or security perimeter. There is one common name for the systems allowing to track and analyze security events, so-called Security Information and Events Management (SIEM) system. In the next section we will explore why SIEM is a must-have tool for any system that requires real-time data analysis for security purposes.

The SIEM systems provide real-time analysis of security events from sensors of information and communication systems. They are represented by applications, devices, or services. It is used to accomplish the following tasks:

- collecting, processing and analyzing security events that come into the system from many sources;
- real-time detection of attacks and violations of security criteria and policies;
- prompt assessment of the security of information, telecommunications and other critical resources;
- security risk analysis and management;
- conducting investigations into incidents;
- making effective decisions to protect information;
- reporting documents.

As one of the most enhanced SIEM systems SPLUNK was selected as the system to consider. The type of SIEM system is irrelevant for this specific paper, however, this fact should be noted to explain the conditions for developing the method.

The basis of the information collected in the SPLUNK system is an index. It is a data repository, which inherently is a file or a set of files that store data.

There is no specific type of data that can be stored in SPLUNK because it can process any data (completely unstructured and poorly structured data is automatically identified and processed by SPLUNK).

That is why in SIEM-system SPLUNK (using so-called “forwarders”) any type of data flow can be obtained:

- configuration files;
- notification of systems and applications;
- alerts;
- metrics;
- scripts;
- log files of changes to databases;
- network data, etc.

The structure of such data streams may be different, but the common feature is the presence of the same data rows, which is defined by a certain structure. For example, for web logs received from a webserver (e.g., Apache server). Following is typically the structure of log data:

- IP address of the request;
- IP address of the request source;
- date and time;
- GET / POST method;
- request URL;
- HTTP status code (response);
- browser information.

The structure may be different depending on the settings; however, each element can be considered a feature of the input. Their set forms a characteristic space of input data. Certain features have numerical meanings, some are symbolic, and some are categorical. The full feature space for further study can be modified and supplemented with processed or calculated data (e.g., URL length, number of bytes of information, number of parameters in a URL request, etc.).

One of the main features of these logs of any type is a temporal feature that forms a discrete space of values and makes it necessary to consider the definition of anomalies in discrete time. One of the approaches allowing the stochastic processes simulation, and/or determination of the system states and transitions between them is a finite state machine.

THE FINITE STATE MACHINES

Within the finite state machine model, we define the following concepts:

- Input alphabet — the set of all possible system inputs defined by the *n*-dimensional feature space. In particular, in our case, the measurement will be determined by the number of features in the SIEM logs.
- Output alphabet — a set of all values that characterize the output streams of information, or actions (system response to an input signal, system status change messages, etc.).

We will consider subset of the flow of data from SIEM systems as a set of states. The input information can be represented as a vector of values — the main features

$$x = (a_1, a_2, a_3, \dots, a_k), \tag{1}$$

where the value of the coordinates of the vector a_j, $j = \overline{1 \dots k}$, is the value of log entry parameters (numeric or categorical, which however should be transformed to a numeric representation). Without loss of generality, configuration files, system messages, and applications can be considered as such. In this way, we establish a connection between the input data and the system states.

To use the numerical values of the input vectors to obtain a finite number of states, it is desirable to reduce them to categorical ones by dividing them into intervals and assigning them to certain categories.

For example, if one of the attributes has definitions on the set of rational numbers, then its minimum and maximum values (thresholds) and the intervals between them that can correspond to the categorical values “low,” “medium,” and “high” should be determined. Values outside the maximum and minimum thresholds will correspond to the categorical values “high” and “low”.

Thus, the input alphabet can be defined as a finite set of all possible states of the vector

$$X = (x_1, x_2, x_3, \dots, x_n), \tag{2}$$

whereby the total number of these states *n* can be determined by the formula (3):

$$n = \prod_{i=1}^{k} |a_i|, \tag{3}$$

where $\prod |\bullet|$ is the operator of determining the power of the set of values, which takes a certain coordinate of the vector of input information.

The output alphabet can be defined within the signals coming from the system

$$Y = (y_1, y_2, y_3, \ldots, y_m). \tag{4}$$

These output signals will allow responding to deviations from the normal values under certain conditions according to the transition matrix, which will be defined below. Specific definitions for multiple outputs are suggested as follows:

- do nothing;
- increase the likelihood of anomaly;
- reduce the likelihood of anomaly;
- signal an anomaly.

Now we define the set of states of the system, which in the general form are as follows:

$$S = (s_1, s_2, s_3, \ldots, s_d), \tag{5}$$

where d is the number of all possible states.

Without reducing the generality of the work and relying on simple logic, we can define two basic states — "normal" and "anomalous".

It should be noted that in this case the type of anomaly will not be determined, but we will determine the likelihood of its occurrence.

We can construct a finite state machine model for certain sets. Its general appearance is as follows:

$$A = \langle X, Y, S, f, h \rangle, \tag{6}$$

Where f, h are the state-to-state transition and the output signal determination functions, respectively.

Method for signatureless anomalous behavior detection. On the one hand, to determine anomalous behavior, we must have a clearly defined behavior that may be considered as normal. On the other hand, sometimes it is very difficult to clearly define the boundaries of the norm, especially when it comes to a person's behavior. For this reason, the authors proposed new methods for detecting some types of anomalies without patterns based on theory of Finite State Machine (FSM), where we map each FSM state with the generic security data collection system, i.e., SIEM [12].

Therefore, based on the definition of a finite state machine, having inputs, outputs, and a description of states with the specified initial state, as well as the function of determining the state and output signals, we can build a general model for determining anomalous behavior.

In the context of this work, we define anomalous behavior as different from what is expected, that is, in this context, "normal".

In this paper, we assume that each value of the input vector is determined by the distribution function. That is, the values of each of the components of the vector are within the space of values (1), forming some specific vector.

Since it is suggested to evaluate the anomalies of behavior probabilistically, it is necessary to introduce a parameter that will determine the value of the likelihood of anomalous state at a certain point in time. A likelihood means a probability with no defined distribution function. In general, you can define state-to-state transition functions and output signal detection functions as follows:

$$s_{t+1} = f(s_t, p_{t+1}), \tag{7}$$

where s_{t+1} and s_t are states values at appropriate times, p_{t+1} is the current value of the anomaly likelihood.

Because the finite state machine is a discrete-time model, it is necessary to define a transition function as a threshold function, which may be, for example, a transition to the opposite state in the presence of a likelihood value above/below a certain threshold. Depending on the current state and the likelihood value, it may change to the opposite state or again to the current state.

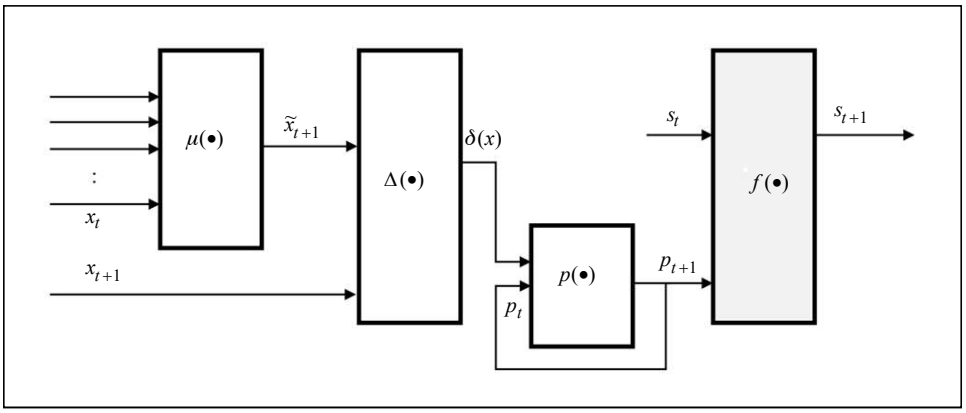

Fig. 2. General graphical diagram of the method of determining anomalous behavior

Now, with a certain sequential ordering of the input vectors, we can construct the predictive value of the next input vector.

Note that in the context of this paper, it does not matter how the predictive value will be determined. We will assume that one way or another, this value has already been obtained. Most often, artificial neural networks are the most commonly used tool for complex data types.

Hereafter, we assume that the predicted value is obtained as a function of $\mu(\bullet)$ from the input dataset (for the time interval x_{t-m} to x_t):

$$\widetilde{x}_{t+1} = \mu(x_{t-m}, \dots, x_{t+1}), \tag{8}$$

Further, to compare this value with the actual value of the input vector, it is necessary to determine a certain measure of the distance of the vectors in the feature space

$$\delta x = \Delta(\widetilde{x}_{t+1}, x_{t+1}), \tag{9}$$

where $\widetilde{x}_{t+1}$ is a predicted value of vector at the time $t+1$, and x_{t+1} is its actual value.

Since we now have a measure of distance, we can determine the function of the dependence of the anomaly likelihood of the state on that distance.

The likelihood determination function ρ can be defined as a function of the current likelihood and a measure of the distance (9) of the predicted vector from the actual one:

$$p_{t+1} = \rho(p_t, \delta x). \tag{10}$$

The task of tuning such a function to real data can be put to machine learning to achieve the accuracy and adequacy of a method for a particular data set. A general scheme of the method for determining anomalous behavior is proposed as depicted in Fig. 2.

According to this method, the last step of the calculation is to determine the state of the system depending on the likelihood and the current state (7). According to (6), $h(\bullet)$ remains the only undefined function. We define it according to the conditions of the adjusted finite state machine, in which the change of states depends on the current state and the calculated likelihood of anomalies:

$$y = h(s_t, p_{t+1}). \tag{11}$$

For anomalies to be determined, the last step of the method should be to determine the thresholds for (7) and (11) in order to obtain a signal of a possible anomaly in the system.

The calculation of the dynamics of equation (6), as well as the prediction of values, can allow to determine the likelihood of anomaly in the data collecting SIEM systems without considering the signatures of certain threats, based only on increasing the likelihood of anomalous behavior due to deviation from the predicted one.

CONCLUSIONS

The paper proposes a novel method for detecting anomalous user behavior based on the distance from expected data. The system is defined by a finite state machine

model using a set of equations (1)–(11), which allows for potential anomalous behavior signals without the need for specific attack signatures. The likelihood of anomalous user behavior can be determined by considering variations of equations (12)–(14) in the analysis of logs in information management and security systems. However, the prediction function used in this approach is a major limitation and a subject for future research. The accuracy of prediction may vary depending on the function used, resulting in different anomaly likelihood values.

REFERENCES

1. Akinlade E., Adeleye E. Designing a secure interactive system: balancing the conflict between security, usability, and functionality. 2022. URL: https://www.researchgate.net/publication/366252638_Designing_a_secure_interactive_system_balancing_the_conflict_between_security_usability_and_functionality (Last accessed: 03 Jun 2023).
2. Rainie L., Anderson J., Connolly J. Cyber attacks likely to increase. 2014. URL: https://www.pewresearch.org/internet/2014/10/29/cyber-attacks-likely-to-increase/ (Last accessed: 03 Jun 2023).
3. On Basic Principles of Cyber Security in Ukraine: the Law of Ukraine of October 5, 2017 № 2163-VIII. The official gazette of Ukraine, 2017. Issue № 91. P. 2765. URL: https://zakon.rada.gov.ua/laws/show/2163-19.
4. Letychevskyi O., Hryniuk Y., Yakovlev V., Peschanenko V., Radchenko V. Algebraic matching of vulnerabilities in a low-level code. *The ISC International Journal of Information Security*. 2019. Vol. 11, Iss. 3. P. 1–7. https://doi.org/10.22042/isecure.2019.11.0.1.
5. Letychevskyi O., Polhul T. Detection of fraudulent behavior using the combined algebraic and machine learning approach. *Proc. 2019 IEEE International Conference on Big Data (Big Data)* (09-12 December 2019, Los Angeles, CA, USA). Los Angeles, 2019. P. 4289–4293. https://doi.org/10.1109/BigData47090.2019.9006546.
6. Chandola V., Banerjee A., Kumar V. Anomaly detection: A survey. *ACM Computing Surveys*. 2009. Vol. 41, Iss. 3. P. 1–58. https://doi.org/10.1145/1541880.1541882.
7. Huang H. Rank based anomaly detection algorithms.: PhD Thesis, Syracuse University, 2013. 182 p. URL: https://surface.syr.edu/eecs_etd/331/.
8. Hawkins S., He H., Williams G., Baxter R. Outlier detection using replicator neural networks. In: Data Warehousing and Knowledge Discovery. DaWaK 2002. Kambayashi Y., Winiwarter W., Arikawa M. (Eds). *Lecture Notes in Computer Science.* 2002. Vol. 2454. P. 170-180. https://doi.org/10.1007/3-540-46145-0_17.
9. Yan W., Yu L. On accurate and reliable anomaly detection for gas turbine combustors: a deep learning approach. *Proc. Annual Conference of the Prognostics and Health Management Society 2015* (18–24 October 2015, Coronado, CA, USA). Coronado, 2015. URL: https://arxiv.org/ pdf/1908.09238.pdf.
10. Dewa Z., Maglaras L.A. Data mining and intrusion detection systems. *International Journal of Advanced Computer Science and Applications*. 2016. Vol. 7, Iss. 1. P. 62–71. https://dx.doi.org/10.14569/IJACSA.2016.070109.
11. Amer M., Goldstein M., Abdennadher S. Enhancing one-class support vector machines for unsupervised anomaly detection. *Proc. ACM SIGKDD Workshop on Outlier Detection and Description* (11 August 2013, Chicago, Illinois, USA). Chicago, 2013. P. 8–15. https://doi.org/10.1145/2500853.2500857.
12. Tkach V., Kudin A., Kebande V.R., Baranovskyi O., Kudin I. Non-pattern-based anomaly detection in time-series. *Electronics*. 2023. Vol. 12, Iss. 3. 721. https://doi.org/10.3390/electronics12030721.

В.М. Ткач, А.М. Кудін, В.К. Задірака, І.В. Швідченко

БЕЗСИГНАТУРНЕ ВИЗНАЧЕННЯ АНОМАЛЬНОЇ ПОВЕДІНКИ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

Анотація. Однією з найактуальніших задач кібербезпеки є своєчасне виявлення кіберзагроз в умовах адаптивного до системи характеру кібератак. Ця задача тісно пов'язана з визначенням нормального та аномального станів, а також поведінки різних процесів в інформаційних системах. Часто додатковою умовою є відсутність шаблонів, сигнатур або правил нормальної поведінки, які б дали змогу застосувати сучасні статистичні або інші відомі методи аналізу даних. У статті наведено аналіз наявних методів виявлення аномальної поведінки. Запропоновано новий метод її виявлення без використання сигнатур на основі моделі кінцевого автомата й системи управління інформацією та подіями інформаційної безпеки.

Ключові слова: виявлення аномалій, кінцевий автомат, SIEM, часовий ряд, кібербезпека.

Надійшла до редакції 14.03.2023