

Inside the Closed World: User and Device Profile Analytics for SCADA Security

Xiaojun Zhou^{1,2}, Zhen Xu¹, Liming Wang¹, Kai Chen¹, Cong Chen^{1,2}, Wei Zhang^{1,2}

¹State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, 100195 E-park C1 North, No. 80 Xingshikou Road, Haidian District, Beijing, China

² School of Cyber Security, University of Chinese Academy of Sciences, 100049 No.19(A) Yuquan Road, Shijingshan District, Beijing, P.R.China

Abstract. Attacks that use sophisticated and complex methods increased recently, aiming to infiltrate the Supervisory Control and Data Acquisition (SCADA) system and stay undetected. Therefore, attackers often get access to authorized permissions of SCADA and bring catastrophic damages by sending ‘legitimate’ control commands. Furthermore, insiders may also misuse or abuse their permissions to damage SCADA system, which is difficult to predict and protect against them. Most existing security systems employ standard signature-based or anomaly-based approaches, which are not able to identify this type of malicious activities. In this paper, we use machine learning algorithms based on Singular Values Decomposition (SVD) to create profiles of users and devices. The major contribution of this paper is providing a general process to detect anomalies, independent of specific use-cases. Suspicious actions are altered to analysts with relevant contextual information for further investigation and action. We provide detailed description of algorithms, methodology, processing of profiling and anomaly detection. Having profiles of different users and devices can provide us a baseline of normal behavior to compare against unusual behaviors. To demonstrate the proposed method, attack scenarios have been simulated at a Compressed Natural Gas (CNG) system in our lab. Experimental results illustrate that the proposed method is effective for abnormal behaviors in SCADA system.

1 Introduction

Modern SCADA system require seamless integration between human and machine where human interactions with SCADA system become a large part of operations [1]. Outside attackers use sophisticated and complex methods to infiltrate SCADA network and to stay undetected, often using valid credentials and standard administrative tools to hide between legitimate user actions and to hinder detection. They get access to authorized permissions of SCADA and brings catastrophic damages by sending ‘legitimate’ control commands. While insiders may also misuse or abuse their permissions to damage SCADA system, which is difficult to predict and protect against them, because they do not include any exploit of a software implementation vulnerability (e.g., protocol implementation) and all commands are legal. As a result, insider attacks are one of the most dangerous threats organizations face today [2] [3].

Most existing security systems employ standard signature-based or anomaly-based approaches, which are not able to identify this type of malicious activities.

What’s worse, it is not feasible to analyze user and entity profile manually, due to the complexity of this task and the high amount of different user and entity. Thus, it is of great importance to develop new automated approaches to analyze user and entity profile for SCADA security.

In this paper, we use machine learning algorithms based on Singular Values Decomposition (SVD) to create profiles of users and devices. User and device behaviors are modeled and a baseline is created. Suspicious actions of users and devices are compared against the profile to detect any abnormal scenarios. The major contribution of this paper is providing a general process to detect anomalies, independent of specific use cases. Suspicious actions are altered to analysts with relevant contextual information for further investigation and action. We provide detailed description of algorithms, methodology, processing of profiling and anomaly detection. Having profiles of different users and devices can provide us a baseline of normal behavior to compare against unusual behaviors. To demonstrate the proposed method, attack scenarios have been simulated at a Compressed Natural Gas

* Corresponding author: author@e-mail.org

(CNG) system in our lab. Experimental results illustrate that the proposed method is effective for anomaly actions in SCADA system.

The remaining of this paper is organized as follows. Section 2 reviews related work in anomaly detection in SCADA system. Detailed explanation of our methodology is presented in Section 3, including the SCADA system model, algorithms and generic architecture of profiling. Section 4 illustrates the profiling of user and device by using enhanced algorithms. Experiments of anomaly detection on CNG testbed are displayed in Section 5. Section 6 gives the conclusion and Section 7 offers future research directions. Finally, we make our acknowledgements in Section 8.

2 Related work

Anomaly detection SCADA has become one of the hottest topics and many researchers and experts have carried out fruitful work in this area. Some employ the specification of protocols to detect anomalies. Reference [4] proposed a vulnerability assessment framework to systematically evaluate the vulnerabilities of SCADA systems at three levels: system, scenarios, and access points. Reference [5] built a model of Modbus / TCP with configuration-level specifications in addition to protocol specifications. Reference [6] considers the specification-based intrusion detection. Reference [7] proposed a way of automatically learn a DFA (deterministic finite automata). Reference [8] extended this approach for Siemens S7 protocol.

Others uses the signature-based anomaly detection in SCADA system. Reference [9] added a module to Snort to develop signatures for ICS (industrial control system) protocols used in electrical utilities, while reference [10] proposed a framework for dynamic rule generation and deep packet inspection. Reference [11] adapted the well-known Bro IDS (intrusion detection system) to support SCADA protocols. Reference [12] [13] proposed a method to model operator behavior of resolving alarms in electric power SCADA by using Petri net. Reference [14] explored the viability of machine learning methods in detecting the new threat scenarios of command and data injection.

There are also some other approaches of academic research on anomaly detection for SCADA system. Reference [15] provided an overview of standard device fingerprinting techniques and an assessment on the application feasibility in ICS infrastructures. Reference [16] proposed different fingerprinting methods designed to augment existing intrusion

detection methods in the ICS environment. In reference [17] an anomaly detection method for SCADA systems based on features including network traffic, link utilization and CPU usage is proposed. All these mentioned work are not applicable to detect anomalies issued by legal users with legitimate control commands to create catastrophic damages [4].

3 Methodology

In this section, we first describe the SCADA system model to illustrate why we employ user and device profile in anomaly detection. Then a detailed explanation of the algorithms used in our method is presented. Finally a generic architecture is showed to illustrate our detection method.

3.1. SCADA system model

We divide SCADA system into three parts: users, network and physical devices. As illustrated in Fig.1, users issue a command through the SCADA network and the actuator take actions accordingly on physical equipment. When the state of physical devices changes, the data will be transmitted back to central site, carrying out any necessary analysis and control and then displaying the information on HMI (human machine interface) for users. There are two kinds of SCADA users [18]: (i) engineers and (ii) operators (or dispatchers). An engineer is responsible for managing object libraries and user interfaces, setting grid topology, normal states and setting parameters of devices, defining process set points, writing automation scripts, etc. While an operator monitors the system status in HMI server, and reacts to alarms and some events to ensure the whole system runs smoothly. The devices include RTUs (remote terminal unit), PLCs (programmable logic controller) and other IEDs (intelligent electronic device). Devices will run in a normal state only if the user issues a command.

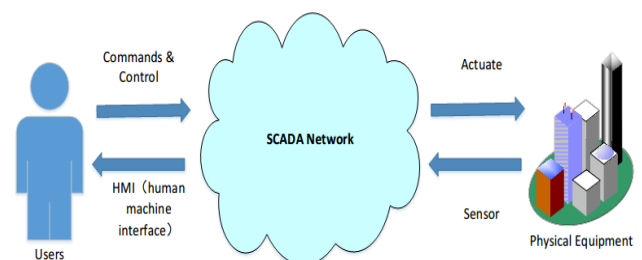


Fig. 1. The abstract structure of SCADA system

SCADA system usually controls critical infrastructure, which affects the daily lives of millions and so the security of SCADA system is of very important. SCADA system is vulnerable to

cyberattacks, which may cause inauspicious degrees of harm to the critical infrastructure. Operators or contractors, with extensive internal knowledge of SCADA architecture and system policies/procedures, can cause insider attack. On the other hand, adversaries, with an insider role, could use valid SCADA control application to perform undesirable actions [12].

3.2. Algorithms

Intuitively, finding anomalies with low probability is equal to finding outlier- s that are far away from the common observations. Our approach is based on the concept of Mahalanobis distance [19], which is a multi-dimensional generalization of the z-score [20]. When each observation has multiple variables, Mahalanobis distance shows how many standard deviations away from an observation is from the mean value of all observations[21]. Mahalanobis distance is selected because it is unit-less and scale-invariant. Given an observation vector of n variables $x=\{x_1, \dots, x_n\}$ and a set of observations $X=\{x_1, \dots, x_m\}$ with mean vector $\mu=\{\mu_1, \dots, \mu_n\}$ and covariance matrix Σ , the Mahalanobis distance is defined as the following E.q.1.

$$M_d = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)} \quad (1)$$

However, the values of mean vector μ and the covariance matrix Σ are not always available because we cannot get the overall distribution. As a result, we have to use the empirical estimates from the observations. What we cannot ignore is that the covariance matrix Σ must be reversible in order to get Σ^{-1} . Reference [22] gives a detailed discussion on the calculations of Mahalanobis distance.

To compute the distance, we take an alternative approach using the observations. We use the sample to represent the whole as an approximation. Then we get mean vector μ . We first normalize all variables to have zero-mean (E.q.2).

$$Y = X - \mu \quad (2)$$

Where $Y=\{y_1, \dots, y_m\}=\{x_1 - \mu, \dots, x_m - \mu\}$

Then, the calculation of the covariance matrix Σ has been simplified as E.q.3.

$$\Sigma = Y^T Y \quad (3)$$

But there is another problem, the matrix Y has m rows and n columns, which is a large sparse matrix. So we use Singular Values Decomposition (SVD) [23] to simplify the computational process (E.q.4).

$$Y_{m \times n} = U_{m \times m} S_{m \times n} V_{n \times n}^T \quad (4)$$

Where U and V are orthodox matrices called left singular matrix and right singular matrix respectively, S is diagonal matrix containing singular values as its diagonal elements. The columns of U indicate orthogonal direction in decreasing order of variance corresponding to decreasing magnitudes of singular values in S. Then the covariance matrix Σ can be computed as

$$\Sigma = V S^2 V^T \quad (5)$$

Finally, we get the Mahalanobis distance as E.q.6.

$$M_d = \sqrt[3]{Y^{-1} V S^{-2} V^T Y} \quad (6)$$

To summarize, all the steps in the algorithm are shown in Algorithm 1.

Algorithm 1 The computation of Mahalanobis distance

Step1. Input the original baseline matrix X;

Step2. Compute mean vector μ and zero-mean X to get Y;

Step3. SVD on Y, then $Y_{m \times n} = U_{m \times m} S_{m \times n} V_{n \times n}^T$;

Step4. Get Σ , then $\Sigma = V S^2 V^T$;

Step5. Compute $M_d = \sqrt[3]{Y^{-1} V S^{-2} V^T Y}$;

We use Mahalanobis distance to cluster the normal behaviors and has made some improvements on the algorithm in several ways to adjust it to meet the actual computing needs in Section 4.

3.3. Generic architecture

The process of user and device profiling is illustrated in Fig.3. Our method is independent of use-cases and can be applied to protect specific components. The entire work flow can be divided into four distinct phases. The detailed description of each phase is given in Figure 2..

1) Data Collecting. We collect all the relevant data about user and device from all data sources: IP address, MAC address, the manufacture, operating system, the list of installed software, the list of port and services, logs of a specific user account, login/out logs, registry information, change-log for a specific file/directory, resource usage log, process operation log, USB usage log, traffic log, access log of resources..... Some information can be obtained directly from the log of the host or server, while other information only can be get from the agent installed on the host/server. Experiment results showed that our agent just increases less than 0.5% usage of CPU so as to maximize the effect. We do not give a detailed information about the agent for it is only our little trick to collect information and it is out of the scope of this paper.

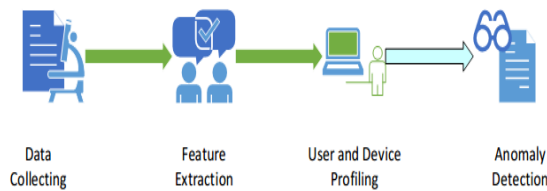


Fig. 2. The generic architecture

2) Feature Extraction. All the gathered information is classified according to its attributes to build five tuples: (U, M) , (M, F) , (U, A) , (A, F) , (M, M) , where U means users vector, M means devices vector, F means the functions vector of a specific device and A means the authorities vector of a specific user. Then the grouped features are computed and stored.

3) User and Device Profiling. Five matrix are created to profile users and devices. Mahalanobis distance and SVD are used in the profiling process. After this step, we create a correlation between users, authorities, devices, functions and actual behaviors.

4) Anomaly Detection. The aim of our method is to detect any anomalies of users and devices. The test vector are scored against the profiles with a description. If it is consistent with profiles created, the behavior is considered benign, or it will be determined as anomaly with a confidence score. As mentioned above, an event often evolves many features, so the probability of an event is the multiply outcome of multiple probabilities. For example, if the n features are involved, the probability of the event is $P(X) = \prod_{i=1}^n P(x_i)$.

4 User and device profiling

In this section, we give the details of user and device profiling process. First, we make some improvements to adjust the algorithms be more appropriate for application. Then the method are described in detail.

4.1. Enhanced algorithms

We extend the algorithms in five aspects to make it more flexible and friendly.

1) Simplification of Computation

We make dimensionality reduction on matrix Y . Based on empirical knowledge, one or more of the singular values have extremely low magnitudes. We can pick a threshold - say 95% or 99% - and choose only as many components from U such that the sum of squares of corresponding singular values (as a percentage of the total sum) is greater than the threshold. If the threshold is given by t , $0 < t < 1$, then

$$r = \arg \min_i, \text{ where } \left(\sum_{j=1}^i s_j^2 / \sum_{\text{all } j} s_j^2 \right) \geq t. \quad (7)$$

We take the first r columns of U to simplify matrix Y and the computation process. Then the E.q.4 can be simplified as E.q.8

$$Y_{m \times n} = U_{m \times r} S_{r \times r} V_{r \times n}^T \quad (8)$$

That is to say, we only care about the key first r features of users and devices in the profiling process. What's more, the left-singular vectors, which are r columns of U , are orthogonal and represent a semantic dimension respectively. And so does right-singular vectors, the r rows of V . Then we can classify the rows of U and the columns of V according to the singular values of S , which means we investigate users and devices in group based on different features. In other words, we don't compare the behaviors of user against all the other users but only a small proportion of the staff based on the singular vectors interested.

2) Different Variable Weights

In practical, different variable weights to different variable is required. The security level of different devices vary, and different operations have different effects on the whole system. Furthermore, the focus of analysts also differ greatly. For instance, if we want to monitor users for writing to the SCADA system but do not care much about their reading activity, then the writing variables contribute more to the Mahalanobis distance when compared to the reading variables. We define a new vector of weights w where w_i is the weight for the i_{th} variable. After that, all the variables have been attached with a corresponding factor w_i .

3) Unidirectional Detection

For a specific behavior, usually the deviations are one-sided. For example, Reference [13] proposes an alarm based statistical anomaly detection method and only detected the alarm handling behaviors lowered than his normal level. While in other scenarios, one might not really care if somebody downloads less than what is normal but would want to know if the download magnitude is much larger than his peers. So we make adjustment on the variables to specify whether to detect deviations in a positive or negative direction against the mean. In detail, we introduce a vector v , where each v_i means the direction of interest.

4) Unified Scoring Mechanism

Mahalanobis distance is not bounded, so the calculation results can be arbitrarily big and we will have to build many evaluation mechanism of different features for their computations may vary largely. We map the Mahalanobis distance into a confidence score in an interval of $[0,1]$ by using an sigmoid function [24]. Let assume md the distance, then the score can be calculated in the following E.q.9

$$score = \frac{1}{1 + \exp^{-kmd}} \quad (9)$$

Where k is the steepness of the S-curve and m_d is the Mahalanobis distance.

5) Meaningful Results

What cannot be ignored is that we not only need to identify the anomaly but also need to provide information on why some behaviors are determined as anomalous to justify our detection. We will display the

context information and demonstrate how much each variable contributes to the anomaly score with its weights. The contribution of j -th variable is given by E.q.10.

$$c_j = (w_j x_j)^2 / \sum_{i=1}^n (w_i x_i)^2 \tag{10}$$

4.2. Profiling process

The profiling process is to build five matrices: (U, M), (M, F), (U, A), (A, F), (M, M), where U means users vector, M means devices vector, F means the functions vector of a specific device and A means the authorities vector of a specific user. Firstly, relationship matrices are created by automatically analyzing the source data we have collected. The following matrices A, B, C, D, E represent the relation of (U, M), (M, F), (U, A), (A, F), (M, M) respectively.

$$A = \begin{matrix} & \mathbf{m}_1 & \mathbf{m}_2 & \cdots & \mathbf{m}_n \\ \mathbf{u}_1 & a_{11} & a_{12} & \cdots & a_{1n} \\ \mathbf{u}_2 & a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \mathbf{u}_m & a_{m1} & a_{m2} & \cdots & a_{mn} \end{matrix} \tag{11}$$

Where u means users and m means devices, a_{ij} means whether the i -th user can operate the j -th device. The value of a_{ij} is defined in E.q.12

$$a_{ij} = \begin{cases} 1, & \text{if the } i\text{-th user can operate the } j\text{-th device} \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

$$B = \begin{matrix} & \mathbf{f}_1 & \mathbf{f}_2 & \cdots & \mathbf{f}_n \\ \mathbf{m}_1 & b_{11} & b_{12} & \cdots & b_{1n} \\ \mathbf{m}_2 & b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \mathbf{m}_m & b_{m1} & b_{m2} & \cdots & b_{mn} \end{matrix} \tag{13}$$

Where m means devices and f means functions, b_{ij} means whether the i -th device has the j -th function. The value of b_{ij} is defined in E.q.14.

$$b_{ij} = \begin{cases} 1, & \text{if the } i\text{-th device has the } j\text{-th function} \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

$$C = \begin{matrix} & \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \\ \mathbf{u}_1 & c_{11} & c_{12} & \cdots & c_{1n} \\ \mathbf{u}_2 & c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \mathbf{u}_m & c_{m1} & c_{m2} & \cdots & c_{mn} \end{matrix} \tag{15}$$

Where u means users and a means authorities, c_{ij} means whether the i -th user has the j -th authority. The value of c_{ij} is defined in E.q.16.

$$c_{ij} = \begin{cases} 1, & \text{if the } i\text{-th user has the } j\text{-th authority} \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

$$D = \begin{matrix} & \mathbf{f}_1 & \mathbf{f}_2 & \cdots & \mathbf{f}_n \\ \mathbf{a}_1 & d_{11} & d_{12} & \cdots & d_{1n} \\ \mathbf{a}_2 & d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \mathbf{a}_m & d_{m1} & d_{m2} & \cdots & d_{mn} \end{matrix} \tag{17}$$

Where u means users and f means functions, c_{ij} means whether the i -th authority can operate the j -th function. The value of d_{ij} is defined in E.q.18.

$$d_{ij} = \begin{cases} 1, & \text{if the } i\text{-th authority can operate the } j\text{-th function} \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

$$E = \begin{matrix} & \mathbf{m}_1 & \mathbf{m}_2 & \cdots & \mathbf{m}_n \\ \mathbf{m}_1 & e_{11} & e_{12} & \cdots & e_{1n} \\ \mathbf{m}_2 & e_{21} & e_{22} & \cdots & e_{2n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \mathbf{m}_n & e_{n1} & e_{n2} & \cdots & e_{nn} \end{matrix} \tag{19}$$

Where m means devices, a_{ij} means whether the i -th device has communicates with the j -th device. The value of e_{ij} is defined in E.q.20.

$$e_{ij} = \begin{cases} 1, & \text{if the } i\text{-th device has communicates with the } j\text{-th device} \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

The elements of these matrices are all can be attached with a corresponding factor w to show each element's importance. After the relationship matrices have been created, SVD is used to realize dimensionality reduction to focus on the main components. Let's take matrix A for example. We use SVD to classify the users and devices and build a relationship between users and devices.

$$A = \begin{matrix} \mathbf{u}_1 & \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1r} \\ u_{21} & u_{22} & \cdots & u_{2r} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{u}_m & u_{m1} & \cdots & u_{mr} \end{pmatrix} & \begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1r} \\ s_{21} & s_{22} & \cdots & s_{2r} \\ \vdots & \vdots & \cdots & \vdots \\ s_{r1} & \cdots & \cdots & s_{rr} \end{pmatrix} & \begin{matrix} \mathbf{m}_1 & \mathbf{m}_2 & \cdots & \mathbf{m}_n \\ \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ v_{r1} & \cdots & \cdots & v_{rn} \end{pmatrix} \end{matrix} \end{matrix} \tag{21}$$

where $s_{ij} = 0$, if $i \neq j$, and $r \ll \min(m, n)$.

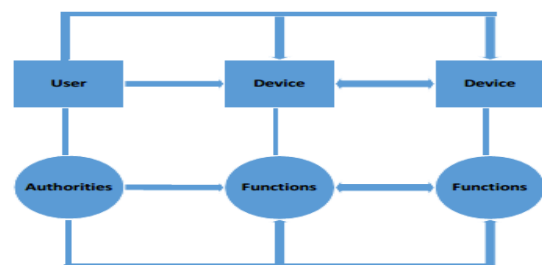


Fig. 3. The relationship of Users and Devices

The matrix multiplication form of matrix A can be illustrated in a simplified form: . In E.q.21, the left-singular vectors, which are r columns of U, are orthogonal and represent a semantic dimension respectively. And so does right-singular vectors, the r

rows of V . While singular values of diagonal matrix S demonstrated the significance of each semantic dimension. Through this manipulation, we obtain the relationship of users and devices. The left matrices B , C , D , E are handled in the same way. Then we will have the profiles of both users and devices, and also the relationship between them. These profiles will serve as the baseline of normal behaviors (See in Fig.3).

5 Anomaly detection

In section 4, we get the profiles of users and devices and the relationship between them. In this section, we will show how to use the profiles to detect anomalies. What is an anomaly? An outlier is an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism^[25]. We can infer that an anomaly is an outlier that appears differently from the majority and occupies a small proportion. The profiles of users can inform us what authorities a user has, which device he usually uses, what operations he usually do. The profiles of devices can tell us what functions a device has, usually operated by which user, and the specific authority needed to perform the function. And the profiles of devices can also tell us whether an illegal connection has been built and whether the communication has something abnormal. We will use these profiles to confirm each other. For example, a grudge staff may commit a damaging operation by installing a virus on different computers. As a sophisticated employee, he may choose the easiest point to launch the attack. Then, the profile of his own can alert us that he has login on a computer which is not his daily using device and the profile of device can alert us that he has installed a software, which behavior is out of his authorities. If the probability of this behavior is lower than 0.3% (which is out of the range of $\mu \pm 3\sigma$), then an alarm is generated. Note that the threshold can be adjusted to adapt to different circumstances of different security level. For those critical components, or suspicious employees, the value can be lowered to detect any potential anomalies, while for common devices and general staff, the value should be set high to reduce the burden of analysts.

Table 1. The size of dataset.

Process	total number	size(Bytes)
Raw Data	10,112,563,217	980M
After Preprocess	1,002,136,791	350M

To demonstrate the proposed method, we use the data collected on simulated Compressed Natural

Gas(CNG) system in our lab, which consists of five operator workstations, two engineer workstations, a historian workstation, a router, a real-time database server and two PLCs(programmable logic controller). All the source data is collected of a duration of 6 months from April 17, 2016 to October 17, 2016. The total logs are 0.98T and the number of logs of different kinds is 10, 112, 563, 217(10 billion). We do some pretreatment on the logs and finally get 0.35T, 1, 002, 136, 791(about 1 billion) comprehensive records, where each new record is a 'snapshot' of the whole system (as showed in Table 1). During this time period, we keep the system going and perform regular operations. All the behaviors, including regular operations and abnormal actions have been logged. Some can be directly obtained from the log of devices, while others are gathered by our agent. We use 10-fold cross-validation in our experiments to ensure the accuracy.

5.1. Anomaly detection using user profile

The user profiles can inform us a user initiates what action on which device and the time of occurrence, or not do what he should do. Usually one or more features are more crucial than others, as illustrated in the following three examples.

(1) Abnormal Loading Activity

In our system, the logs of the simulated system need to be copied to a mobile hard disk on the end of day to release the space of the computer. In order to find illegal copy action, we set the anomaly detection of positive direction, which means we don't care about the user downloads less than what is normal but would want to know it he would copy something else. The vector is defined as $\mathbf{x} = \{\text{type, bytes, attributes, encryption, number}\}$, where type means the file type.log (different file type is set to different values), bytes means the total size of the file, attributes means read(1), write(2), or execute(4)(when there are more than one attributes, we use the highest value), encryption means whether the file is encrypted, number means the total number of the files. For the variable type is continuous, we use normal distribution as approximation, while for variable type is discrete and only has two values, we set $p(x = 1) = 99.99\%$. For other discrete distribution, we use its real distribution. The normal distribution is as Fig.4.

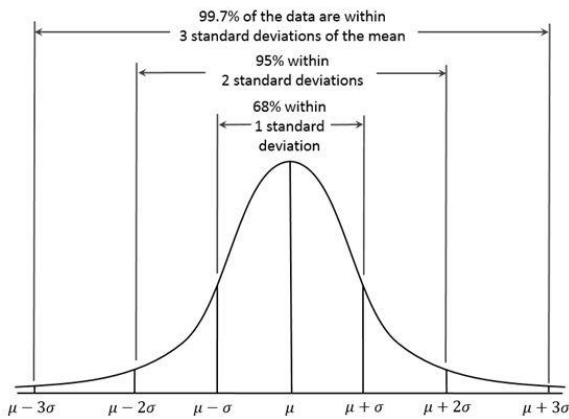


Fig. 4. The normal distribution with probability range

We normalize all the variable $\left(\frac{x_i - \mu_i}{\sigma_i}\right)$, to get a standard normal distribution to have 0 mean and unit variance. For example, the normal behavior is $x = \{1, 1000, 1, 0, 160000\}$, which means the file type is .log (), the total bytes is 1000M (we use MB for the sake of accuracy), file attribute is read only (attributes are), not encrypted (if encrypted, the value is 1), and the mean value of total number is 160000. If we get a new vector $v = \{2, 1005, 4, 0, 160002\}$. Then we can calculate the probability by

$$P(v) = \prod_{i=1}^5 P(x_i) = 0.01 * 0.99 * 0.01 * 0.99 * 0.99 = 10^{-4}$$

which means the new vector v is abnormal and an alarm will be generated. When we use only one feature as a threshold, the result is not satisfactory. For example, the total file number and bytes of the operator's is recorded along the time line as in Fig.5. From the picture we can see that the total bytes is always around 1000M, while the file number has two peak points that seem abnormal. Traffic-based detection mechanism can only find a small proportion of anomalies while our method can achieve a very high accuracy. What's more, our method provides the analyzer with a description of the user and device. In other words, we correlate all the actions to a specific user and device using the enhanced Mahalanobis distance.

(2) Not Doing What He Should Do

In SCADA system, the number of alarms generated by the system is with- in a small range [13]. The operator need to handle these alarms timely. In order to monitor the operator's alarm-handling actions, we set the anomaly detection of negative direction, which means we want the operator to perform his duties and handle the alarm timely. The baseline vector is defined as $x = \{\text{number, duration}\}$, with its value of $x = \{5, 50\}$, which means the operator handles 5 alarms per day and the total time cost is 50min. Test vector is $v = \{5, 200\}$. The probability of this action is 0.01, so an alert is generated.

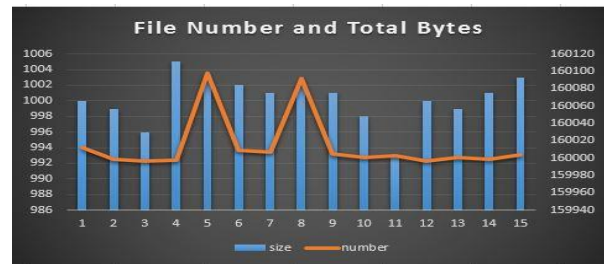


Fig. 5. File number and total bytes.

Also we can use his peers of the same group to confirm the result. Note that we don't need to compare all the staff with him but only those of most similar features. From the profiles, we find that only 3 operators has the most similarities. The philosophy of simplification has been explained in detail in section 4.

(3) Wrong Operation Order

The operation is sequential in SCADA system. So if the sequence is not as normal, disastrous consequence may occur. For instance, he normal action order is action 1, action 2, action 3, and action 4. As showed in Fig.6 (a).

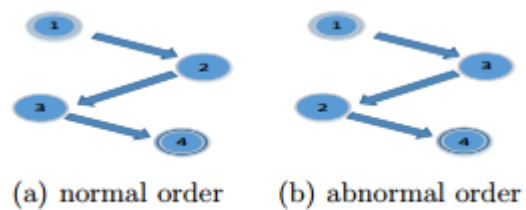


Fig. 6. Action sequence anomaly detection

However, we find that the operation order is action 1, action 3, action 2, and action 4 on October 10, 2016 (Fig.6 (b)). Although this action causes no damage to our simulated system, but it is devastating in real production environment.

5.2. Anomaly detection using device profile

The device profiles can tell us a specific device is operated by whom and performs what function and has a communication with which another device.

(1) Communication Anomaly Detection

According to the traffic of all the devices, the communication vector between device 1 and device 2 can be modeled as $x = \{n1, n2, \text{bytes}\}$, which means that there are two protocols between device 1 and device 2, and the packet number of the protocols are n1 and n2 respectively. If these two protocols have 100,000 and 20,000 packets respectively and the total bytes is 1000M, then

$x = \{100000, 200000, 1000\}$. When the test vector is $v = \{100000, 500000, 2500\}$, we calculate the probability to be 10^{-3} , so an alarm is generated. We can also detect whether a device has something wrong by check the profiles of its own and its communication peers.

(2) Abnormal Tasks

By using the agent, we can have a detailed monitoring of the resource usage of each device. When an unknown increase or decrease of usage occurs, an

anomaly may happen. Since SCADA operations are often cyclical and the communications are deterministic, modeling normal behavior by mining its specific features is feasible^[26]. The baseline vector is defined as $\mathbf{x} = \{\text{CPU, IO, RAM, bandwidth}\}$, with its value of $\mathbf{x} = \{0.3, 0.6, 0.6, 0.5\}$, which means the usage of CPU, IO, RAM, bandwidth is 0.3, 0.6, 0.6, 0.5 respectively. Test vector is $\mathbf{v} = \{0.7, 0.8, 0.8, 0.6\}$. We calculate the probability to be 0.02, so an alarm is generated.

6 Conclusion

In this paper, we propose a general mechanism to build user and device profiles. We make adjustment on Mahalanobis distance by using samples to estimate the whole and using SVD to simplify the calculation. After getting the baseline, we take advantage of SVD to greatly reduce the number of peers to comparison. And all the variables have been attached with a weights w to show its importance, and the anomaly detection can be set in a positive or negative direction, which is very preferable in real application. Last but not the least, we calculate the probability of the new actions. Experimental results (detection rate up to 99%) illustrate that the proposed method is effective for abnormal actions in SCADA system.

7 Discussion and future network

However, when users cause problems with security, it may often be difficult to judge the degree to which the problem was caused by ignorance, negligence, or mischievous intent^[27]. Reference [28] illustrates that attacks of users (such as social engineering) is a real-world threat and we need to develop better technical defenses against them, and learn how to effectively teach end users about these risks. Reference [29] gives a structured review of all the standards and shows us that all standards or regulations emphasize 'management' greatly. That is corresponding with the motto Security is 30 percent by technology, and 70 percent by management.

In future, some other machine learning methods and fine-grained feature extraction will be tested to improve detection accuracy..

Acknowledgements

We thank our shepherds—Zhen Xu, Liming Wang in our research group, for providing insightful feedback of the draft that helped improve the final paper. We would also like to thank Kai Chen, Zelong Chen and Zhenbo Yan for their help in early discussions and providing insightful comments. This work was supported by Research on Core Technologies of

national key infrastructure security supervision platform. Beijing Municipal Science & Technology Commission of China under grant No.Z161100002616032, for which we are grateful.

References

1. Jennifer U. Mills, Steven M. F. Stuban, and Jason Dever. Predict insider threats using human behaviors. *IEEE Engineering Management Review*, 45(1):39–48, 2017.
2. Ramkumar Chinchani, Duc Ha, Anusha Iyer, Hung Q. Ngo, and Shambhu Upadhyaya. Insider threat assessment: Model, analysis and tool. *Network Security*, pages 143–174, 2010.
3. Nathalie Baracaldo and James Joshi. An adaptive risk management and access control framework to mitigate insider threats. *Computers and Security*, 39(4):237–254, 2013.
4. Chee Wooi Ten, Chen Ching Liu, and Govindarasu Manimaran. Vulnerability assessment of cybersecurity for scada systems. *IEEE Transactions on Power Systems*, 23(4):1836–1846, 2008.
5. Mustafa Faisal, Alvaro A. Cardenas, and Avishai Wool. Modeling modbus tcp for intrusion detection. In *Communications and Network Security*, pages 386–390, 2017.
6. Steven Cheung, Bruno Dutertre, Martin Fong, Ulf Lindqvist, Keith Skinner, and Alfonso Valdes. Using model-based intrusion detection for scada networks. *Proceedings of the Scada Security Scientific Symposium*, 2006.
7. Niv Goldenberg and Avishai Wool. Accurate modeling of modbus/tcp for intrusion detection in scada systems. *International Journal of Critical Infrastructure Protection*, 6(2):63–75, 2013.
8. Amit Kleinmann and Avishai Wool. Accurate modeling of the siemens s7 scada protocol for intrusion detection and digital forensics. *Journal of Digital Forensics Security and Law*, 9(2):37–50, 2014.
9. Y. Yang, K. Mclaughlin, T. Littler, S. Sezer, B. Pranggono, and H. F. Wang. Intrusion detection system for iec 60870-5-104 based scada networks. In *Power and Energy Society General Meeting*, pages 1–5, 2013.
10. Jeyasingam Nivethan and Mauricio Papa. Dynamic rule generation for scada intrusion detection. In *Technologies for Homeland Security*, 2016.

11. Hui Lin, Adam Slagell, Catello Di Martino, Zbigniew Kalbarczyk, and Ravis-hankar K. Iyer. Adapting bro into scada: building a specification-based intrusion detection system for the dnp3 protocol. In Eighth Cyber Security and Information Intelligence Research Workshop, pages 1–4, 2013.
12. Payam Mahmoudi Nasr and Ali Yazdian Varjani. Petri net model of insider attacks in scada system. In International ISC Conference on Information Security and Cryptology, pages 55–60, 2014.
13. Payam Mahmoudi Nasr and Ali Yazdian Varjani. Alarm based anomaly detection of insider attacks in scada system. In Smart Grid Conference, pages 1–6, 2015.
14. Justin M. Beaver, Raymond C. Borges-Hink, and Mark A. Buckner. An evaluation of machine learning methods to detect malicious scada communications. In International Conference on Machine Learning and Applications, pages 54–59, 2014.
15. Marco Caselli, Dina Had?iosmanovi?, Emmanuele Zambon, and Frank Kargl. On the Feasibility of Device Fingerprinting in Industrial Control Systems. Springer International Publishing, 2013.
16. David Formby, Preethi Srinivasan, Andrew Leonard, Jonathan Rogers, and Ra- heem Beyah. Who’s in control of your control system? device fingerprinting for cyber-physical systems. In Network and Distributed System Security Symposium, 2016.
17. Dayu Yang, Alexander Usynin, and J Wesley Hines. Anomaly-based intrusion detection for scada systems. 2006.
18. Bill Miller and Dale Rowe. A survey scada of and critical infrastructure incidents. In Conference on Research in Information Technology, pages 51–56, 2012.
19. P. C Mahalanobis. On the generalized distance in statistics. Proceedings of the National Institute of Sciences, 2:49–55, 1936.
20. Z-Score Model. Springer US, 2006.
21. Dimitrios Ververidis and Constantine Kotropoulos. Information loss of the mahalanobis distance in high dimensions: Application to feature selection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(12):2275–81, 2009.
22. R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart. The mahalanobis distance. Chemometrics and Intelligent Laboratory Systems, 50(1):1–18, 2000.
23. G. W. Stewart. On the early history of the singular value decomposition. Society for Industrial and Applied Mathematics, 1993.
24. K. Zhang. Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. Journal of Neuroscience the Official Journal of the Society for Neuroscience, 16(6):2112, 1996.
25. D. M. Hawkins. Identification of outliers. Biometrics, 37(4):860, 1980.
26. Iaki Garitano, Roberto Uribeetxeberria, and Urko Zurutuza. A review of sca- da anomaly detection systems. In Soft Computing MODELS in Industrial and Environmental Applications, International Conference Soco 2011, 6-8 April, 2011, Salamanca, Spain, pages 357–366, 2011.
27. Jeffrey M Stanton, Kathryn R Stam, Paul Mastrangelo, and Jeffrey Jolton. Anal- ysis of end user security behaviors. Computers and Security, 24(2):124–133, 2005.
28. Matthew Tischer, Zakir Durumeric, Sam Foster, Sunny Duan, Alec Mori, Elie Bursztein, and Michael Bailey. Users really do plug in usb drives they find. In Security and Privacy, pages 306–319, 2016.
29. Xiaojun Zhou, Zhen Xu, Liming Wang, and Kai Chen. What should we do? a structured review of scada system cyber security standards. Proceedings of 2017 4th International Conference on Control, Decision and Information Technologies (CoDIT’17), pages 0605 – 0614, 2017.