Hindawi
Security and Communication Networks
Volume 2024, Article ID 9850571, 1 page
https://doi.org/10.1155/2024/9850571

Retraction

Retracted: Network Interconnection Security Buffer Technology for Power Monitoring System

Security and Communication Networks

Received 8 January 2024; Accepted 8 January 2024; Published 9 January 2024

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] J. Wang, J. Wu, W. Tao, W. Zhu, and W. Qiu, "Network Interconnection Security Buffer Technology for Power Monitoring System," *Security and Communication Networks*, vol. 2022, Article ID 6371062, 11 pages, 2022.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 6371062, 11 pages
https://doi.org/10.1155/2022/6371062

Research Article

Network Interconnection Security Buffer Technology for Power Monitoring System

Jifeng Wang, Jinyu Wu, Wenwei Tao, Wen Zhu, and Weijie Qiu

China Southern Power Grid Co., LTD, Huangpu District, Guangzhou, Guangdong 510623, China

Correspondence should be addressed to Jinyu Wu; wujinyu0301@163.com

Received 7 March 2022; Revised 14 April 2022; Accepted 22 April 2022; Published 30 May 2022

Academic Editor: Zhiping Cai

In recent years, the risk of malicious attacks on power monitoring systems has increased, and there have been many attacks on power systems in the world. Aiming at the network interconnection security problem of the core control system, the concept of "security buffer" is introduced, and a network security buffer method forpower monitoring system is proposed, which is composed of three parts: paradigm check, behavior analysis, and dynamic conversion and jointly realizes the multilevel security inspection of interconnection requests. Experimental verification results show that the proposed method has a protective effect on malicious attacks of power monitoring system.

1. Introduction

In recent years, an increasing number of security incidents have happened to the industrial control system, especially to the power monitoring system. In 2010, Stuxnet invaded the Iranian nuclear power station, disabling 20 percent of centrifuges and severely impeding the implementation of Iran's nuclear power plan. Stuxnet was a destructive worm specifically targeting the industrial control system. It aimed to attack the PLC, and data acquisition and supervisory systems of Siemens, steal its system permission, and further maliciously changed control parameters. In 2015, the Black Energy left more than half of Ukraine without power. In 2018, TSMC's machine equipment was used for blackmail, which got its chip production in trouble. Frequent industrial control security incidents have attracted extensive attention from home and abroad. China and European and American countries have included the industrial control system in their national strategies [1–3].

Some studies have been conducted targeting network security of industrial control systems. The studies comprise two aspects: on the one hand, the learning algorithm is used to train the model and detect the attack behavior based on extracted data features or traffic characteristics. The literature [4, 5] used SVM to model the flow interval and the length of data packets for the network traffic of industrial control system and designed an intrusion detection system; Zhao Guicheng [6] proposed building a behavior model based on function code and start address in Modbus protocol and applying SVM algorithm to the analysis of abnormal behavior. Zhu et al. [7] designed and achieved a multiclass SVM algorithm for the intrusion detection in the perspectives of function code or behavior characteristics; Li Wei et al. [8] proposed a SCADA system intrusion detection approach, which sets out intrusion detection rules by the white list and based on analysis of behavior protocol; Parvania et al. [9] presented a behavior-based intrusion detection system for communication behaviors and protocol specifications of smart grid system by means of statistical analysis of traditional network features and specification-based detection. However, as the attack has turned to slow penetration, statistics of network flow cannot satisfy the demand. At present, there are also some scholars who propose the addition of relevant parameters (such as control command) and semantic descriptions (such as trusted measured values) to the detected characteristics to detect system attacks such as wrong command injection and tampering messages. On the other hand, protocols are subject to the uniform description by protocol analysis in order to detect noncompliant protocols. Suda et al. [10] put

forth an intrusion detection algorithm of time-series features extracted based on time characteristics of series, which extracts effectively the time series features by recurrent neural network (RNN); by virtue of time series loop structure of RNN, and the temporal dependence of samples, Yan Binghao et al. [11] proposed an intrusion detection model based on deep recurrent neural network (DRNN) and region adaptive synthetic oversampling algorithm. But the jobs give little consideration for the behavioral interdependence among control commands. The protocol descriptions, which are either too complicated to popularize or less expressive to explain complex protocols and have slow protocol analysis problems, are unsuitable for the scenes of the power monitoring system.

Therefore, the concept of "security buffer" is introduced to this paper, and a network security buffer method for power monitoring systems is proposed. A security buffer is a memory area that is used between the input and output devices and the CPU to store safety data. It enables the low-speed input/output devices and the high-speed CPU to work in coordination, avoiding the low-speed input/output devices from taking up the CPU and freeing up the CPU so that it can work efficiently. The method is composed of three parts: paradigm check, behavior analysis, and dynamic conversion and jointly realize the multilevel security check on network requests. The paradigm check module examines message format and filters data packets that fail to meet the standard message specification; the behavior analysis module analyzes the sequence of packets and blocks request sequence targeting multiple packets' abnormal behaviors; the dynamic conversion module utilizes format conversion or confusion to implement data structure changes and unload the attacker's attack modes such as buffer overflow attacks.

Compared with existing works, the main contributions and innovations of this paper are as follows:

(1) A unified description language for defining interconnection protocol packets of power monitoring systems is given, which supports the description and parsing of complex heterogeneous protocols and provides a basis for subsequent unified analysis.

(2) Introducing the idea of redundant heterogeneity and adding a dynamic conversion function in the security buffer, which can prevent attackers from trying to speculate the normal working mode and then carry out precise attacks by dynamically adjusting the conversion strategy.

(3) Experimental evaluation of the proposed method shows that the proposed method has a high accuracy rate of detecting multipacket anomalous behavior and the proposed dynamic conversion strategy is effective for offloading buffer overflow attacks.

The other parts of this paper are organized as follows: Section 1 introduces problematic scenes that this method targets; Section 2 deals with detailed design of the method; Section 3 makes an assessment of the proposal by experiment; at last, Section 4 concludes the paper and discusses work to be done next.

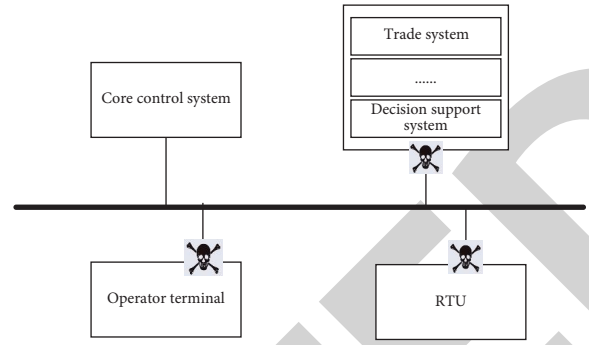

Figure 1: Security threats to the network interconnection of power monitoring system.

2. Problematic Scenes

With computers, communication equipment, measurement and control units as basic tools, the power monitoring system provides a basic platform for real-time data acquisition, switch status detection, and remote control of power generation, transmission, transformation, and distribution systems. The system, together with detection and control equipment, can make up any complex supervisory system.

The network interconnection of power monitoring system is mainly exposed to the following security risks, as shown in Figure 1:

(1) The network architecture of the current power monitoring system is relatively simple. Equipment and core control system are directly accessible through network protocols by operators and at data acquisition places, which provides a springboard for attackers to use the vulnerability of the core control system to attack the system and then destroy the power security. For the primary technological means, hackers exploit vulnerabilities of application protocols in the power monitoring system and create attack load elaborately, triggering buffer overflow vulnerability; then they inject attack loads such as viruses and Trojan horses into the core control system, thus undermining the system security.

(2) The current power monitoring system and other systems on the main network side are accessible. Attackers can first break through other systems, and then use this springboard to scan vulnerabilities of the core control system, operating system, and middleware; then, they use the vulnerabilities to launch brute force attack and remote code injection and finally destroy the security of the core control system.

Therefore, it poses a great risk by directly exposing the core control system of the power monitoring system to operators, data acquisition points, or other business systems. For this reason, this paper proposes a security buffer before the core control system to offload the attacks towards the core control system and secure it.

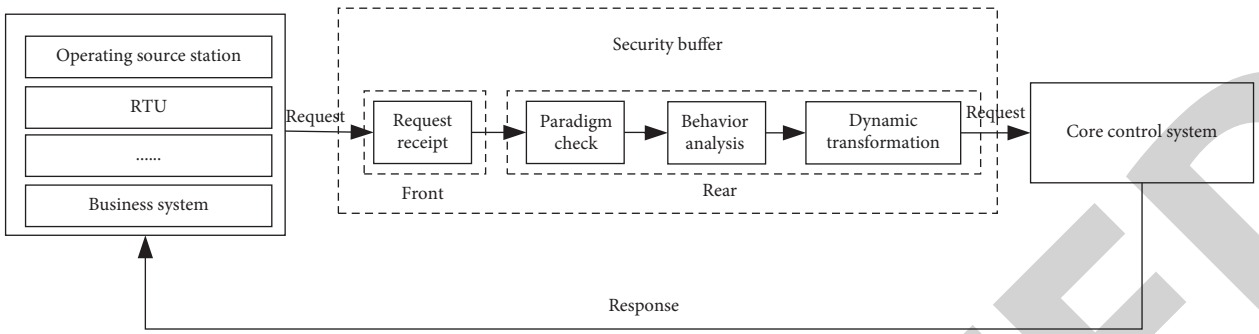

Figure 2: Architecture of network interconnection security buffer technology for the core control system of power monitoring system.

Table 1: Descriptions of XML-based Internet protocol packet paradigm for the power system.

Tag	Meaning	Remarks
<data></data>	Protocol packet	None
<filed></filed>	Field	The field includes two attributes: name of the attribute field and length (in bytes); if the field is a fixed value, it is written directly in the tag
<rule></rule>	Rule	There are three types of rules: 1 (0 : 1024) indicating: 0 ≤ value ≤ 1024; 2 [a,b,c] means that the value can be a, *b*, or *c*; 3 (like '%1') indicates that the field is restricted to those ending with 1;
![CDATA[]	Special tags	Information in [ ] is passed on in the integrity by the parser to the applications, without parsing any control marks in this message.

3. Method Design

3.1. How It Works. Based on the above analysis, this paper presents a network interconnection security buffer technology for the core control system of power monitoring system. This method adds a security buffer between the core monitoring system and other systems or operating terminals to defend against malicious attacks. The method architecture is shown in Figure 2. The security buffer deploys three main functional modules: paradigm check, behavior analysis, and dynamic conversion.

3.2. Paradigm Check. Paradigm check is to examine the protocol specifications of network interconnection packets of the power monitoring system. This section offers a packet paradigm, which supports uniform descriptions of varied network interconnection protocol packets of IEC 104, IEC, and 101 power monitoring systems. The paradigm can be used to define specifications for the Internet protocol data, i.e., rules for the analysis and check of request packets. The data packet will continue to be carried forward depending on the subsequent analysis of the request packet and the check that the packet matches protocol data specification.

3.2.1. XML-Based Uniform Description of Internet Protocols. To support checks of more Internet protocol data specifications, this paper presents a multiprotocol packet paradigm based on the extensible markup language (XML), which is applied for the uniform description of various Internet protocol data specifications of the power system. The descriptions of the XML-based Internet protocol packet paradigm for the power system are shown in Table 1.

Table 2: Example packet paradigm of the XML-based IEC 104 protocol.

<data>Tag
<filed name = "Boot_character" length = "1">68*H*</filed> <filed name = "APDU_Length" length = "1"> <rule>![CDATA[(0 : 253)]</rule> </filed> <filed name = "Control1″ length = "1"></filed> <filed name = "Control2″ length = "1"></filed> <filed name = "Control3″ length = "1"></filed> <filed name = "Control4″ length = "1"></filed> <filed name = "Type" length = "1"></filed> <filed name = "Determiner" length = "1"></filed> <filed name = "TransferReason" length = "1"></filed> <filed name = "DataAddr" length = "2"></filed> <filed name = "MessageBody" length = " APDU_length-10"></filed> </data>

Taking "68 0E 00 00 00 00 64 01 06 00 01 00 00 00 00 14" (master call request that master station sends to slave station in the mainstream IEC 104 protocol of current power system) as an example, a packet paradigm for the XML-based IEC 104 protocol is presented, as shown in Table 2.

The security buffer administrator gives a uniform description of the specifications for Internet protocol data based on the paradigm above. The protocol data specification file may be named by "protocol name.xml", e.g., IEC104.xml, and the captured request packets are subsequently analyzed and checked according to the rules in the specification.

3.2.2. Packet Analysis and Format Check. Like traditional Internet protocols, the Internet protocol packets of the

Table 3: Paradigm check algorithm for request packets.

Input: Request data, protocol name. XML file
Output: 1 and compliant packets/0, discard noncompliant packets; "1" indicates the request packet meets the protocol data specification, and "0" means noncompliance
1 input RequestData
2 analyze protocol name.XML, and generate data structure "S" and rule "R"
3 analyze and unify format of RequestData according to the structure of "S"
4 for $i = 1$ to n/* traverse the rule set "R" * /
5 for $j = 1$ to m/* traverse the set "S" and search the corresponding field based on the field name in the rule set*/
6 if S[j].name = R[i].name/* compare the fields, if the field names are the same*/
7 The field if conforms to the protocol rule defined by the user
8 i++
9 else data frame is discarded, return 0/* discard the data in case of inconformity*/
10 end if
11 else j++
12 end if
13 end for
14 end for
15 output the parsed data structure "S" in the format defined in Table 2
16 return 1

power system are encapsulated top-down in a sequence of the application layer, transport layer, network layer, data link layer, and physical layer. Therefore, the analysis and format check of Internet data of power system mainly consists of analyzes and checks of the network layer, transport layer, and application layer, which are shown as follows.

The capture time of data packets shall be saved prior to protocol analysis, as the control behavior sequence in the power monitoring system is sensitive to time.

<filed name = "time"></filed>

Analysis of network layer mmainly to check the source IP address (SourIP) and the destination IP address (DesIP) on the network layer of packets. The identity information of visitors can be acquired through IP address detection, which provides support for access control and intrusion detection. The information below is saved:

<filed name = "SourIp" ></filed>

<filed name = "DesIp"></filed>

Check of transport layer mainly to examine the source port number (SourPort) and the destination port number (DesPort). Different applications usually use different ports for communication. Port check may help discover some application's connection and access to the target application resources. The following information is saved after analysis:

<filed name = "SourPort" ></filed>

<filed name = "DesPort" ></filed>

Analysis and check of application layer: it is the focus of check on the request packet paradigm, which mainly examines protocol information on the application layer, including the function codes and field values that represent the control behavior.

The paradigm check algorithm of request packets is shown in Table 3. The Internet protocol data specification defined based on the paradigm in 3.2.1 (e.g., IEC104.xml) is first parsed to construct the set $S = \{S1,S2,S3,\ldots,Sm\}$, where Sj $\{j = 1,\ldots,m\}$ represents a field in the protocol in the form of a key-value pair of name and value, i.e., Sj = (name, value), and a rule set $R = \{(R1,R2, R3,\ldots,Rn\}$ is generated as well, where Ri$\{i = 1,\ldots,n\}$ is the specification in the protocol data, representing the specification requirements of a particular field; then the captured request packets are formatted according to S for unification, and finally S is checked according to R to see, for example, whether the function code is compliant and whether the value of the data is out of the range of values.

The master station sends the master call request "68 0E 00 00 00 00 64 01 06 00 01 00 00 00 00 00 00 14" to the slave station, which is subject to algorithm check before output in the format, as shown in Table 4.

The compliant request packets through analysis and check on the application layer, plus the information field extracted from the transport and network layers are saved based on the XML paradigm shown in Table 2, and noncompliant packets are directly discarded.

3.3. Behavior Analysis

3.3.1. Extraction of Behavior Sequence. The indexes of control behavior sequence mainly focus on the control operation interaction process between every two devices on the network of power monitoring system. For the purpose of real-time monitoring and calculation, it is necessary to depend on the analysis result of request packets over a period of time. As stated in Section 3.2, a compliant request packet that has passed paradigm check corresponds to an XML file, whose format is shown in Table 4. Through the time window of time span, we captured the packet analysis result corresponding to

Table 4: Paradigm check output files.

<filed name = "Boot_character" length = "1">68H</filed>
<filed name = "APDU_Length" length = "1">0*E*</filed>
<filed name = "Control1" length = "1">00</filed>
<filed name = "RandomNumber" length = "1">04</filed>
<filed name = "Control2" length = "1">00</filed>
<filed name = "Control3" length = "1">00</filed>
<filed name = "Control4" length = "1">00</filed>
<filed name = "Type" length = "1">64</filed>
<filed name = "Determiner" length = "1">01</filed>
<filed name = "TransferReason" length = "1">06</filed>
<filed name = "DataAddr" length = "2">00 01</filed>
<filed name = "MessageBody" length = " APDU_Length-10">00 00 00 00 14</filed>

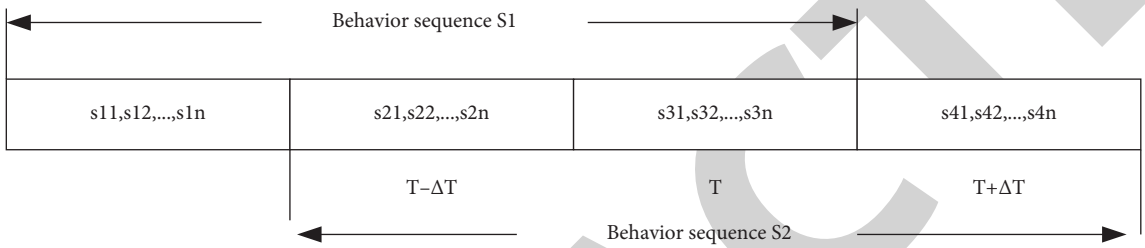

Figure 3: Principle of the incremental window extraction mechanism.

this period of time and extracted a phased behavior sequence. At the time of extraction of behavior sequence, it is necessary to extract the source IP (SourIP), destination IP (DesIP), source port (SourPort), destination port (DesPort), application layer protocol) (Proto), and function code that represents control behavior (Control) and capture time (time).

The IP address and port number at both ends of the control behavior sequence and the protocol type are used as identifiers to distinguish the control behavior sequence. The packets in the time window are grouped according to the quintuple of identification fields (<SourIP>, < DesIP >, < SourPort >, < DesPort >, <Proto>), and all control behaviors are sorted according to time to obtain the behavior sequence [<Controlk>] ranked by control operations, thus obtaining the characteristic data of the control behavior sequence.

<SourIP>,<DesIP>,<SourPort>,<DesPort >,<Proto>: [<Controlk>].

This characteristic data embody the features of control behaviors in a period of time.

When the time window strategy is used to capture packets, to avoid mis-segmentation of multiple single control operations of a continuous related control behavior, the extraction accuracy of the control behavior sequence can be improved based on the partition length "T" and the incremental window of "ΔT" length. The principle of the incremental window extraction mechanism is shown in Figure 3:

3.3.2. Abnormal Behavior Recognition. The information acquired in power monitoring systems may have problems such as inconspicuous data labels and the noisy samples. Given that One-Class Support Vector Machine (OCSVM) algorithm has the features of not requiring neither any algorithm for modeling nor abnormal samples and being robust to noisy samples during training, this paper introduces OCSVM, which has significant advantages over other unsupervised learning methods, to identify the anomalous behavior of network interconnection in power monitoring systems.

The basic idea of OCSVM is to apply the training samples of the same class, map the input space of the training samples into a high-dimensional space by a kernel function to find the optimal classification hyperplane, maximize the distance from the hyperplane to the origin, and to obtain the probability density region of the data in the feature space. Check whether the new input sample point is in the region of the training data point; corresponding quadratic programming problems are shown in formulas (1) and (2).

$$\min_{\omega,\xi_i,\rho} \frac{1}{2}\|\omega\|^2 + \frac{1}{\nu l}\sum_{i}^{l}\xi_i - \rho. \tag{1}$$

Thus

$$\Phi(x_i)\omega \geq \rho - \xi_i, \xi_i > 0, i = 1, \ldots, l. \tag{2}$$

where the training samples $x_1, x_2, \ldots, x_l \in X$, l is the total number of training samples, $\Phi: X \longrightarrow H$ is the mapping of the original feature space to the high-dimensional space, and ω and ρ are the normal vector and compensation of the desired hyperplane in the feature space, respectively. $\nu \in (0,1]$ is the upper limit of the marginal error score and the lower limit of the support vector fraction, and ξ_i is the relaxation variable.

A decision function that represents classification hyperplane is finally obtained as below:

$$f(x) = \operatorname{sgn}(\Phi(x)\omega - \rho). \tag{3}$$

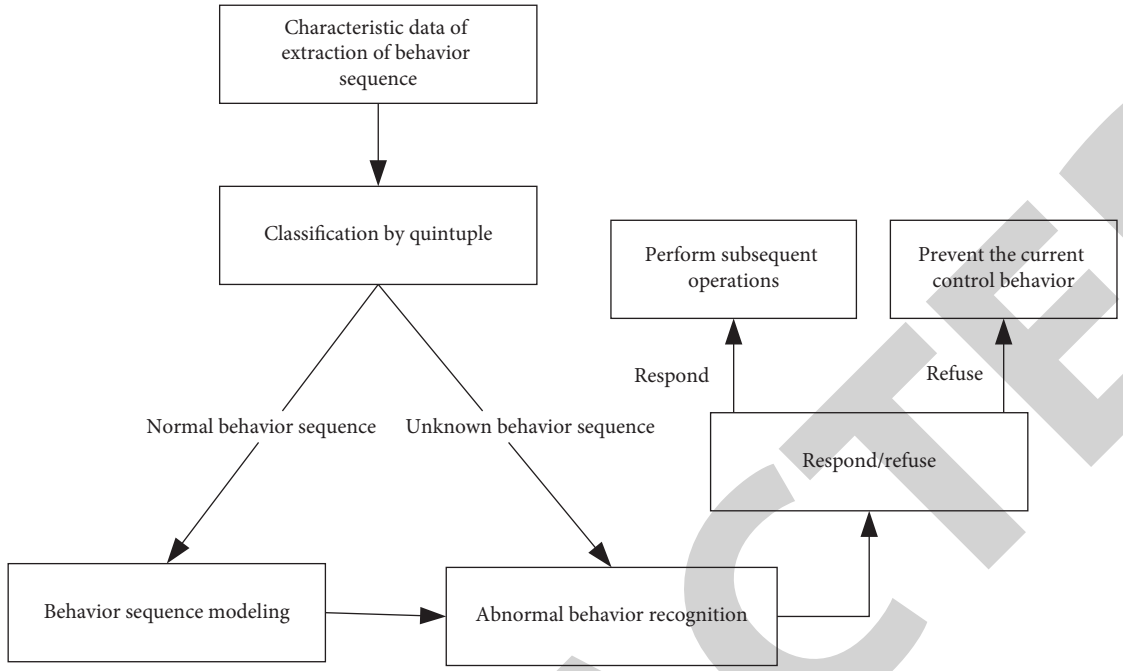

Figure 4: Recognition process of abnormal behavior.

With the Lagrangian function and the Gaussian kernel function introduced, the dual problem of the objective quadratic programming problem can be obtained as below:

$$\min_{\alpha} \frac{1}{2} \sum_{i}^{l} \sum_{j}^{l} \alpha_i \alpha_j K\left(x_i, x_j\right). \tag{4}$$

Thus

$$0 \le \alpha_i \le \frac{1}{vl}, i = 1, \ldots, l \sum_{i}^{l} \alpha_i = 1. \tag{5}$$

where, $K(x_i, y_j)$ is the kernel function, and the vectors that satisfy $0 \le \alpha_i \le 1/vl$ are called support vectors; the final decision function obtained is shown as Formula (6), in which NSV is the number of support vectors.

$$f(x) = \operatorname{sgn}\left(\sum_{i}^{NSV} \alpha_i K\left(x_i, x_j\right) - \rho \right). \tag{6}$$

The process of building an abnormal behavior recognition model based on OCSVM is shown in Figure 4. First, the extracted behavior sequence feature data are classified according to the quintuple, and the behavior sequence si (that is, $\text{Control}_k >$ above) is obtained by time window partition as the data set "S". The sequence si in S is vectorized and transformed into a feature vector x_k of specified k dimension to generate a training sample set X. The OCSVM model is obtained according to X training. When the unknown type of behavior sequence s' is obtained, it is vectorized to generate x', and the resulting feature vector is substituted into the training model to check whether the output x ' is a normal behavior; thus, the recognition of abnormal behavior sequences is achieved. The specific algorithm is shown in Table 5.

When getting the detection result, the security buffer decides whether the current control behavior is allowed or blocked depending on the result; and if not, this packet is discarded.

3.4. Dynamic Conversion. After analyzing the behavior, the data information carried by the packet will be subject to dynamic conversion. Generally, attacks are a pattern of attacks that are carefully designed by the attacker to make the attack successful after he or she is familiar with the system. Therefore, this paper designs and introduces a dynamic conversion module to the security buffer to converse transmitted data according to a predefined policy, so that the attack mode is changed and the data entering the system does not make an attack on the system, which is equivalent to an effective defense against the corresponding attack. By reference to the idea of "redundant heterogeneity", which refers to the use of multiple functionally or performance-equivalent heterogeneous components in parallel, multiple conversion policies are designed in the dynamic conversion module. A policy is selected randomly each time, and the policies are updated from time to time so that the dynamic conversion module itself can remain effective.

3.4.1. Format Conversion of Protocol Data. After a parsed request packet is obtained, select a number randomly from the parsed request packet. The positions of the front and back fields are swapped centering on the "random number" to transform the protocol data format. The reason for random number is to ensure security and prevent tampering

Table 5: Abnormal behavior recognition algorithm.

Input: a Training set of normal behavior sequence S and a unknown behavior sequence *s*'
Output: 1/0, 1 represents s', belonging to normal behavior, and 0 indicates abnormal behavior
1 read the training set of normal behavior sequence *S*
2 The constructed vector model transform it into k-dimensional feature vector x_k to generate the training sample set X
3 train the OCSVM model based on the training sample set
4 vectorize the unknown behavior sequence *s*' to obtain the feature vector *x*'
5 substitute *x*' into OCSVM model and check whether *x*' is normal behavior
6 If it is normal behavior of the model
7 output 1
8 else output 0

Table 6: Format conversion algorithm of protocol data.

Input: Data output from paradigm check
Output: OutData after structural adjustment
1 read data, saved as InData in array format
2 get the array length
3 generate a random, with the random < length
4 create a new array OutData[length]
5 OutData = reverse (InData, random)/ * swap positions of the contents in the front and back according to the random
6 add a new field to the converted data (<filed name = "RandomNumber">random</filed>: Describing the specific value of the generated random)
7 output OutData

Table 7: Format of converted data structure.

<filed name = "Control3" length = "1">00</filed>
<filed name = "Control4" length = "1">00</filed>
<filed name = "Type" length = "1">64</filed>
<filed name = "Determiner" length = "1">01</filed>
<filed name = "TransferReason" length = "1">06</filed>
<filed name = "DataAddr" length = "2">00 01</filed>
<filed name = "MessageBody" length = " APDU_Length-10">00 00 00 00 14</filed>
<filed name = "Control2″ length = "1">00</filed>
<filed name = "Boot_character" length = "1">68*H*</filed>
<filed name = "APDU_Length" length = "1">0*E*</filed>
<filed name = "Control1" length = "1">00</filed>
<filed name = "RandomNumber" length = "1">04</filed>

Table 8: Redundancy-based data obfuscation algorithm.

Input: Data output from paradigm check
Output: OutData after redundancy is added to the control domain
1 input data, and extract the control field action in the data
2 action is an eight-digit number, with storage location is 76543210 from low to high; the eight-digit number is divided into four parts, i.e., 76,54,32, and 10, which are, respectively, the first, the second, the third, and the fourth bytes in the 32-digit number
3 generate a random 32-digit number "ActionPro", each digit of which is random; two maximum digits are taken from each byte
4 The first two bits of each byte are sequentially spliced together to form a new data
5 data mod 7 = *z*
6 then, starting from the zth bit of each byte (from left to right), the numbers 76,54,32, and10 in aciton are stored sequentially, generating AcitonPro
7 write AcitonPro back to the data to generate OutData
8 output OutData

attacks during data transmission. The specific algorithm is shown in Table 6.

To take data in Table 4 as an example, if the random is 04, the format of converted data in the application layer is shown in Table 7.

The corresponding packet changes to "00 00 64 01 06 00 01 00 00 00 00 14 00 68 0E 00 05".

3.4.2. Data Obfuscation. An attacker may modify the control fields in the application layer protocol to achieve illegal control of the device or host. Therefore, protection can be provided by adding redundant bits of data. For example, the type identifier of the behavior in IEC 104 is a 1 byte 8-digit number; it can be converted into 32 bit. The specific algorithm is shown in Table 8.

Table 9: Obfuscated data.

<filed name = "Boot_character" length = "1">68*H*</filed>
<filed name = "APDU_Length" length = "1">0*E*</filed> <filed name = "Control1" length = "4">54 80 C0 30</filed> <filed name = "Control2" length = "4">54 80 C0 30</filed> <filed name = "Control3" length = "4">54 80 C0 30</filed> <filed name = "Control4" length = "4">54 80 C0 30</filed> <filed name = "Type" length = "1">64</filed> <filed name = "Determiner" length = "1">01</filed> <filed name = "TransferReason" length = "1">06</filed> <filed name = "DataAddr" length = "2">00 01</filed> <filed name = "MessageBody" length = " APDU_Length-10">00 00 00 00 14</filed>

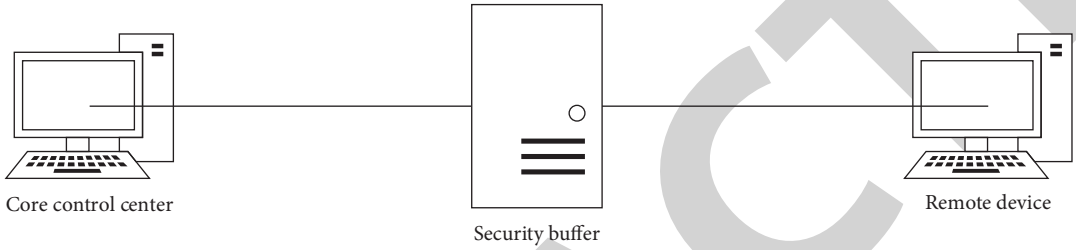

Figure 5: Experimental topology.

Still taking the data in Table 4 as an example, the random 32 bit number is 56 82 C2 32; when the control domain is obfuscated, the converted data format in the application layer is as shown in Table 9.

The corresponding packet changes to "68 0E 54 80 c0 30 54 80 C0 30 54 80 C0 30 54 80 C0 30 64 01 06 00 01 00 00 00 00 14".

It should be noted that the implementation of conversion policies affects the performance of the power system to a certain extent; thus, the policies can be dynamically increased or decreased depending on the specific scenario and different security protection requirements.

4. Experimental Evaluation

4.1. Experimental Environment. We performed simulation experiments on three computers with Windows10, Intel Core i7-9700F, 3.0 GHz CPU, and 32 GB memory, in which one was used as a security buffer functional computer to implement and deploy paradigm check module, behavior analysis module, and dynamic transformation module; one to build Ubuntu16.04 virtual machine to simulate the attacked core control center; and one as a remote device to launch an attack on the target computer. The experimental topology is as shown in Figure 5.

4.2. Effect of Behavior Analysis. In this experiment, we used OCSVM as the learning algorithm for security buffer behavior analysis. Based on UNSW-NB15 data set, we simulated the control behaviors under various scenarios including remote control, remote signaling, remote regulation, and telemetry in the power monitoring system, a total of 1500 sequences of control behavior under normal operation conditions, to give normal sequence model training. Meanwhile, for the common types of attack on the core control systems, and considering the difficulty in obtaining abnormal sequences, abnormal control behavior sequences generated by several attack types such as random operations, repetitive instructions, inversion of time series, and unknown commands were stimulated in the experiment based on construction, clipping, swapping, and falsification for normal behavior sequences. Abnormal behavior sequences and some normal behavior sequences are selected to generate a test set.

The experiment adopted precision and recall as indexes to test the effect of the behavior analysis method proposed in this paper. The computing method is as below [15, 16]:

$$\text{precision} = \frac{\text{true positive}}{\text{predicted positive}} \times 100\%. \quad (7)$$

$$\text{recall} = \frac{\text{true positive}}{\text{total positive}} \times 100\%. \quad (8)$$

We applied Gaussian radial basis function in the OCSVM modeling, in which the parameter gamma indicates degree of nonlinear mapping [17, 18]. The upper bound of modeling error nu[0,1] is the upper limit of the marginal error score and the lower limit of the fraction of support vector. This parameter is used to adjust the precision of the model description of the sample distribution. To make the model contain as many samples as possible, it is necessary to set the upper bound of this error at a low level; and if there is too much noise in the data set, nu may be heightened properly to avoid serious overfitting. Generally, nu is 0.1. To describe the model precisely, $u = 0.01$ was selected as a matched group. Gamma is 0.5 by default. Based on the value, we expanded the value range as a matched group.

Test the behavior analysis effect of the experiment according to the formulas (7) and (8); Figure 6 displays changes of precision along with gamma, and Figure 7 shows changes of recall with gamma.

According to the above experimental results, the behavior analysis method proposed in this paper put in a good performance on precision, which can be above 90 with the changes of gamma's value, but the recall remains to be improved. Considering the difficulty in obtaining and marking malicious samples in the actual power monitoring system, abnormal behavior identification based on OCSVM is still an effective and feasible solution.

To further test the effectiveness of OCSVM, it was compared with the unsupervised learning methods K-Means clustering algorithm [19] and PCA algorithm [20], where gamma = 0.5 and nu = 0.1, as shown in Table 10. As can be seen from Table 10, OCSVM has significant advantages over other methods in terms of accuracy and recall and is suitable for the security protection system of power monitoring systems.

4.3. Effect of Dynamic Conversion. The experiment demonstrates buffer overflow attack, granting common users root privileges to the core control systems and displays the effect of attack uninstallation by dynamic conversion strategy. For the purpose of better exhibition of the effect, the address randomization of the virtual machine in the core control center of the simulation was shut down during the experiment, the StackGuard protection scheme was disabled at the time of code compile, and nonexecutable stacks were turned off.

In the experiment, we compiled stack.c first as a program on the virtual machine of the core control center. The function of the program is to create a 24 byte memory buffer and later transmit data to the buffer via the strcpy() function. Since the strcpy() function does not check the bounds, there is a vulnerability of buffer overflow. What comes next was to compile the program exploit.c that uses stack.c. The main function of the program is to put a piece of shellcode[] (refer to Table 11, more than 24 byte) in the memory, compute its address in the memory, and then work out the return address of stack.c in the program call stack. Through data transmission, exploit.c is sent to the virtual machine at the core control center. When stack.c is executed on the virtual machine, shellcode[] will be saved in the buffer, and due to overflow, the address of shellcode[] will overlay the return address, and the codes in shellcode[] are executed instead.

Normal users on the virtual machine of the core control center can obtain root privilege to the control host by executing exploit.c and then stack.c, until # appears on the command line, as shown in Figure 8.

The dynamic conversion strategy used in the experiment is to swap the contents before and after a certain position in the array. As shown in Table 12, the content of shellcode[] changes, and the contents before and after "50" have their positions swapped.

The code execution results after dynamic conversion is shown in Figure 9. When the stack.c program is executed

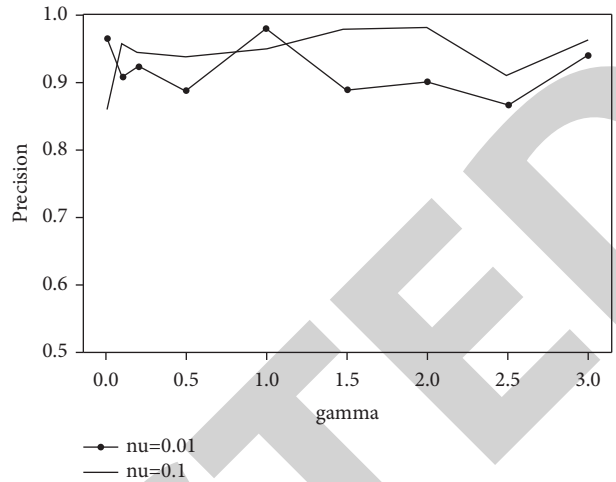

Figure 6: Changes of precision with gamma.

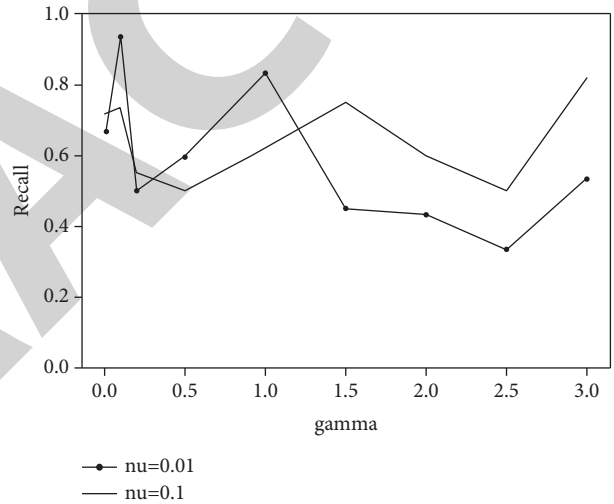

Figure 7: Changes of recall with gamma.

Table 10: Methods comparison.

Methods	Precision	Recall
K-means	0.746	0.375
PCA	0.875	0.428
OCSVM	0.936	0.5

Table 11: Shellcode[] codes.

Char shellcode[] =			
"\x31\xc0"	/*xorl	%eax,%eax	*/
"\x50"	/*pushl	%eax	*/
"\x68""//sh"	/*pushl	$0 × 68732f2f	*/
"\x68""/bin"	/*pushl	$0x6e69622f	*/
"\x89\xe3"	/*movl	%esp,%ebx	*/
"\x50"	/*pushl	%eax	*/
"\x53"	/*pushl	%ebx	*/
"\x89\xe1"	/*movl	%esp,%ecx	*/
"\x99"	/* cdq		*/
"\xb0\x0b"	/*movb	$0 × 0b,%al	*/
"\xcd\x80"	/* int	$0 × 80	*/

```
[09/02/21]seed@VM:~/host$ gcc -o exploit exploit.c
[09/02/21]seed@VM:~/host$ ./exploit
31 c0
50 68
2f 2f
73 68
68 2f
62 69
6e 89
e3 50
53 89
e1 99
b0 b
cd 80
[09/02/21]seed@VM:~/host$ ./stack
#
```

Figure 8: Common users successfully obtain the root privilege.

Table 12: Codes after dynamic conversion.

Char shellcode[] =			
"\x53"	/* pushl	%ebx	*/
"\x89\xe1"	/* movl	%esp,%ecx	*/
"\x99"	/* cdq		*/
"\xb0\x0b"	/* movb	$0 × 0b,%al	*/
"\xcd\x80"	/* int	$0 × 80	*/
"\x50"	/* pushl	%eax swap the contents front and back based on this flag bit	*/
"\x31\xc0"	/* xorl	%eax,%eax	*/
"\x50"	/* pushl	%eax	*/
"\x68" "//sh"	/* pushl	$0 × 68732f2f	*/
"\x68" "/bin"	/* pushl	$0x6e69622f	*/
"\x89\xe3"	/* movl	%esp,%ebx	*/

```
[09/02/21]seed@VM:~/host$ gcc -o exploit exploit.c
[09/02/21]seed@VM:~/host$ ./exploit
53 89
e1 99
b0 b
cd 80
50 31
c0 50
68 2f
2f 73
68 68
2f 62
69 6e
89 e3
[09/02/21]seed@VM:~/host$ ./stack
Returned Properly
```

Figure 9: Attack uninstallation.

once again on the host of the control center, returned properly will appear on the command line, indicating that stack.c is successfully executed; the return address fails to leap to the other memory space, which shows that the conversion strategy takes effect.

5. Conclusion

This paper puts forward a network interconnection security buffer method targeting the core control system of power monitoring system to address the network interconnection security in the power monitoring of core control system. This method adds a security buffer between the core control system for power monitoring and the other system or the operating terminal. Three functional modules such as paradigm check, behavior analysis, and dynamic conversion are deployed in the security buffer to make multilevel security inspection of interconnection request packets. Among them, a unified description language is given for defining interconnection protocol packets of power monitoring systems, which supports the description and parsing of complex heterogeneous protocols, OCSVM in behavior analysis has significant advantages over other unsupervised learning methods and can be effectively adapted to the power monitoring system environment, by introducing the idea of redundant heterogeneity and adding a dynamic conversion function in the security buffer, the conversion policy can be dynamically adjusted to prevent attackers from trying to speculate the normal working mode and then carry out precise attacks. This method can uninstall attacks against the core control system and secure the system. The proposal increases a security buffer, which can exert a certain influence on the instantaneity of the power monitoring system and may make a few erroneous judgments on the identification of malicious behaviors. In the future, we will study more effective behavior analysis algorithms to guarantee the real-time performance of the power monitoring system and further improve the security protection capability of the system.

Data Availability

The labeled data set used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The author declares no competing interests.

Acknowledgments

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

[1] F. Alrimawi, L. Pasquale, and B. Nuseibeh, "On the automated management of security incidents in smart spaces," *IEEE Access*, vol. 7, Article ID 111513, 2019.

[2] F. Alrimawi, L. Pasquale, D. Mehta, N. Yoshioka, and B. Nuseibeh, "Incidents are meant for learning, not repeating: sharing knowledge about security incidents in cyber-physical systems," *IEEE Transactions on Software Engineering*, vol. 48, no. 1, pp. 120–134, 2022.

[3] D. Schlette, M. Caselli, and G. Pernul, "A comparative study on cyber threat intelligence: the security incident response perspective," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2525–2556, 2021.

[4] M. A. A. Mutha and M. R. R. Tuteja, "Secure and efficient approach for mul-tilayer cyber security based on intrusion detection system," *International Journal of Advent Research in Computer and Electronics*, vol. 2, no. 2, pp. 33–37, 2015.

[5] A. Terai, S. Abe, S. Kojima, Y. Takano, and I.. Koshijima, "Cyber-Attack Detection for Industrial Control System Monitoring with Support Vector Machine Based on Communication Profile," in *Proceedings of the Symposium on Se-Curity and Privacy Workshops (EuroS&PW)*, pp. 132–138, IEEE, Paris, France, April 2017.

[6] G. Zhao, *Research on Intrusion Detection Platform and Algorithm Based on Industrial Control Network Traffic Analysis*, Zhejiang University, Zhejiang, China, 2019.

[7] B. Zhu and S. Sastry, "SCADA-specific Intrusion Detection/Prevention Systems: A Survey and taxonomy," in *Proceedings of the First Workshop on Se-Cure Control Systems*, Stockholm, Sweden, April 2010.

[8] W. Li, J. Li, and X. He, "Research on the determination method of industrial control system network security baseline based on traffic analysis," *Science and Technology Bulletin*, vol. 34, no. 9, pp. 176–179, 2018.

[9] M. Parvania, G. Koutsandria, and V. Muthukumary, "Hybrid control network intrusion detection systems for automated power distribution systems," in *Proceedings of the Forty Fourth Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pp. 774–779, IEEE, Atlanta, GA, USA, June 2014.

[10] H. Suda, M. Natsui, and T. Hanyu, "Systematic intrusion detection technique for an in-vehicle network based on time-series feature extraction," in *Proceedings of the IEEE Forty Eighth International Symposium on Multiple-Valued Logic*, pp. 56–61, Linz, Austria, May 2018.

[11] B. Yan and D. Han, "A combined intrusion detection model based on deep recurrent neural network and improved SMOTE algorithm," *Journal of Network and Information Security*, vol. 4, no. 7, pp. 48–59, 2018.

[12] S. Fong and S. Narasimhan, "An unsupervised bayesian OC-SVM approach for early degradation detection, t, and fault prediction in machinery monitoring," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.

[13] A. Derhab, M. Belaoued, I. Mohiuddin, F. Kurniawan, and M. K. Khan, "Histogram-based intrusion detection and filtering framework for secure and safe in-vehicle networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2366–2379, 2022.

[14] J. Liu, D. Yang, M. Lian, and M. Li, "Research on intrusion detection based on particle swarm optimization in IoT," *IEEE Access*, vol. 9, pp. 38254–38268, 2021.

[15] J. Chen, J. Han, X. Meng, and Y. H. Li, "Graph convolutional network combined with semantic feature guidance for deep clustering," *Tsinghua Science and Technology*, vol. 27, no. 5, pp. 855–868, 2022.

[16] M. Heydarian, T. E. Doyle, and R. Samavi, "MLCM: multi-label confusion matrix," *IEEE Access*, vol. 10, Article ID 19083, 2022.

[17] A. K. Seghouane and N. Shokouhi, "Adaptive learning for robust radial basis function networks," *IEEE Transactions on Cybernetics*, vol. 51, no. 5, pp. 2847–2856, 2021.

[18] S. A. Sivaram and K. J. Vinoy, "Inverse multiquadric radial basis functions in eigenvalue analysis of a circular waveguide using radial point interpolation method," *IEEE Microwave and Wireless Components Letters*, vol. 30, no. 6, pp. 537–540, 2020.

[19] K. P. Sinaga and M. S. Yang, "Unsupervised K-means clustering algorithm," *IEEE Access*, vol. 8, Article ID 80716, 2020.

[20] B. Minnehan, N. Nagananda, and A. Savakis, "GrIP-PCA: g iterative P-norm principal component analysis," *IEEE Open Journal of Signal Processing*, vol. 1, pp. 90–98, 2020.