Hindawi
Security and Communication Networks
Volume 2021, Article ID 7525092, 14 pages
https://doi.org/10.1155/2021/7525092

Research Article

ICSTrace: A Malicious IP Traceback Model for Attacking Data of the Industrial Control System

Feng Xiao,[1] **Enhong Chen**,[1] **Qiang Xu**,[2] **and Xianguo Zhang**[3]

[1]*Anhui Province Key Laboratory of Big Data Analysis and Application School of Computer Science and Technology, University of Science and Technology of China, Hefei, China*
[2]*Electronic Engineering Institute of Hefei, Hefei, China*
[3]*School of Cyberspace Security, University of Science and Technology of China, Hefei, China*

Correspondence should be addressed to Feng Xiao; xiaof686@mail.ustc.edu.cn

Received 4 June 2021; Accepted 15 July 2021; Published 2 August 2021

Academic Editor: Jingyu Feng

Considering that the attacks against the industrial control system are mostly organized and premeditated actions, IP traceback is significant for the security of the industrial control system. Based on the infrastructure of the internet, we have developed a novel malicious IP traceback model, ICSTrace, without deploying any new services. The model extracts the function codes and their parameters from the attack data according to the format of the industrial control protocol and employs a short sequence probability method to transform the function codes and their parameters into a vector, which characterizes the attack pattern of malicious IP addresses. Furthermore, a partial seeded K-means algorithm is proposed for the pattern's clustering, which helps in tracing the attacks back to an organization. ICSTrace is evaluated based on the attack data captured by the large-scale deployed honeypots for the industrial control system, and the results demonstrate that ICSTrace is effective on malicious IP traceback in the industrial control system.

1. Introduction

With the rapid development of the Internet of Things (IoT), more and more Industrial Control Systems (ICS) are connected into the Internet. As the key bond between the virtual signal and the real equipment, an Internet-connected ICS makes the production process be more accurate and agile. But it also narrows the distance between the cyberattacks and the industrial infrastructure. As we know, Stuxnet worm was disclosed to be the first worm attacking the energy infrastructure [1, 2] in 2010. In 2014, the hackers attacked a steel plant in Germany so that the blast furnace cannot be closed properly [3]. On December 23, 2015, the Ukrainian power network suffered a hacker attack, which was the first successful attack to the power grid, resulting in hundreds of thousands of users suffering power blackout for hours [4]. In 2017, the security vendor ESET disclosed an industrial control network attack weapons named as win32/Industroyer, which implemented malicious attacks on power substation system [5].

ICSs are highly interconnected and interdependent with the critical national infrastructure [6], and thus the attackers have noticed the high returns to attack ICS in recent years. The attackers are diverse in identity. They may be hackers, members of organized criminal groups, or even a hostile country. The worse situation is that ICS has become the new target of terrorists to gain the influence by destroying the real physical world. As traditional ICS is physically isolated from the Internet, most researches just focus on the functional safety of the system rather than the security consideration of the network. There are not any special protective measures, not to mention the attribution mechanism for tracing the attack back [7]. Security researchers are now committed to the intrusion detection technology for ICS. They want to

identify, intercept, and alert the threats, before a severe attack occurs. These intrusion detection technologies can be divided into several categories as follows: state-based [8], behavior-based [9], rule-based [10], characteristic-based [11], model-based [12], and ML-based (machine learning) [13, 14].

Because ICS plays an important role in the critical national infrastructure, the cyberattacks against ICS are mostly organized and premeditated actions. It is significant not only to determine whether there is a threat in ICS but also to trace the attack back. Furthermore, locating the initiators and their motivations before or during an attack is crucial for deterring and cracking down the premeditated and organized attackers.

Attribution is one of the most intractable problems of an emerging field, created by the underlying technical architecture and geography of the Internet [15]. The current dominant IP traceback technologies include packet marking mechanism [16], packet logging mechanism [17], and their hybrid [18, 19]. Packet marking mechanism needs the routers to write a tag (e.g., IP address) into some fields of every packet. The target retrieves all the tags from the received packets and finds out the routing path. Packet marking mechanism includes two categories: probabilistic packet marking (PPM) [16] and deterministic packet marking (DPM) [20]. Packet logging mechanism needs the routers to record all the forwarded packets so as to reveal the routing path. Apparently, this mechanism consumes a lot of storage space. All these IP traceback technologies above need to redesign the Internet or to deploy new services. There is still no applicable IP traceback system to deploy over the network.

The ultimate goal of attribution is identifying an organization or a government, not individuals [15]. Our study identifies an organization by zooming down to a single IP level and then zooming back out to an organization or a unit level without changing the Internet architecture or deploying new services. Instead of tracing back to the source of a packet directly, we just recognize the malicious IP addresses which belong to the same organization.

In this study, we present a malicious IP traceback model (ICSTrace) for industrial control system, and this model makes the following contributions:

(1) Based on the deep analysis of ICS protocol S7, the function codes and their parameters are extracted from the attack data.

(2) A feature vector of the function codes and their parameters are designed to represent the attack patterns.

(3) The slide window method is adopted to reduce the dimension of those multidimensional samples.

(4) A partial seeded K-means clustering algorithm is proposed based on K-means algorithm.

(5) ICSTrace is proven to be effective basing on the real attack data captured by the large-scale deployed honeypots for ICS.

Section 2 introduces the research background and our previous work on the attack data collection. Section 3 describes the architecture of our IP traceback model. Sections 4 and 5 introduce the attack pattern extraction method and partial seeded K-means algorithm for clustering, respectively. In Section 6, we evaluate our IP traceback model basing on the real attack data. Section 7 is our related works, and Section 8 is the conclusion.

2. Background

ICS is a business process management and control system which is composed of various automatic control and process control components. It collects and monitors real-time signals to ensure the function of the automatic operation or the process control. Its application fields include program automation, industrial control, intelligent building, power transmission and distribution, smart meter, and car communication. ICS protocol refers to the communication protocol used in ICS. The most well-known ICS protocol includes S7, Modbus, BACnet, and DNP3.

S7 protocol is a Siemens proprietary protocol [21] running on programmable logic controllers (PLCs) of Siemens S7-200, 300, and 400 series. It is suitable for either Ethernet, PROFIBUS, or MPI networks. Because the objects of this study are those industrial control systems which are accessed to the Internet, we only discuss the TCP-based S7 protocol in Ethernet networks. As shown in Figure 1, S7 protocol packets are packed by COTP protocol and then packed by TPKT protocol package for TCP connection.

As shown in Figure 2, the communication procedure of S7 protocol is divided into three stages. The first stage is to establish COTP connection, the second stage is to set up S7 communication, and the third stage is to exchange the request and the response for function code.

The Magic flag of the S7 protocol is fixed to 0×32, and the following fields are S7 type, data unit ref, parameters length, data length, result info, parameters, and data. In parameters field, the first byte stands for the function code of S7. Table 1 shows the optional function codes of S7. Communication Setup code is used to build a S7 connection; Read code helps the host computer to read data from PLC; Write code helps the host computer to write data to PLC. As for the codes of Request Download, Download Block, Download End, Download Start, Upload, and Upload End, they are designed for downloading or uploading operations of blocks. PLC Control code covers the operations of Hot Run and Cool Run, while PLC Stop is used to turn off the device.

When the function code is 0×00, it stands for system function which is used to check system settings or status. The details are described by the 4-bit function group code and 1-byte subfunciton code in the parameters field. System functions are further divided into 7 groups, as shown in Table 2. Block function is used to read the block, and time function is used to check or set the device clock.

At present, there is not any ICS attacking dataset for security research. Therefore, we developed a high interactive ICS honeypot named as S7commTrace in previous work [22], based on Siemens' S7 protocol.

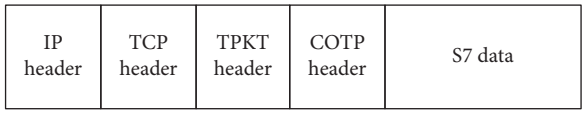

Figure 1: Header format of the S7 communication packet.

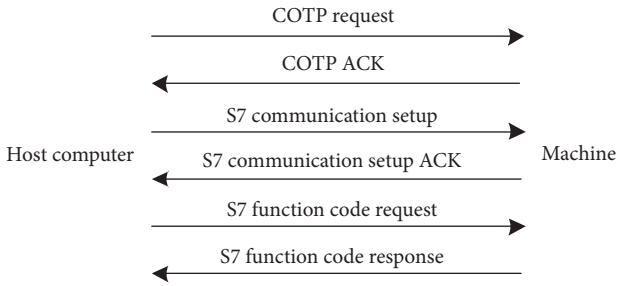

Figure 2: Communication procedure of the S7 protocol.

Table 1: S7 protocol function code and the corresponding function.

Code	Function
0x00	System functions
0x04	Read
0x05	Write
0x1a	Request download
0x1b	Download block
0x1c	Download end
0x1d	Download start
0x1e	Upload
0x1f	Upload end
0x28	PLC control
0x29	PLC stop
0xf0	Communication setup

Table 2: When the function code is 0×00, it is system function and further divided into 7 groups.

Function group code	Function	Subfunction code	Subfunction
1	Programmer commands	1	Request diag data
		2	VarTab
2	Cyclic data	1	Memory
3	Block function	1	List blocks
		2	List blocks of type
		3	Get block info
4	CPU function	1	Read SZL
		2	Message service
5	Security	1	PLC password
6	PBC BSEND/ BRECV	None	None
7	Time function	1	Read clock
		2, 3	Set clock
		4	Read clock (following)

Honeypot is a kind of security resource that is used to attract the attacker for illegal application without any business utility [23]. Honeypot technology is a method to set some hosts, network services, or information as a bait, to induce attackers, so that the behavior of the attacks can be captured and analyzed [24]. Honeypot can be used to better understand the landscape of where these attacks are originating [25].

S7commTrace poses as a real PLC device by simulating the S7 protocol to capture the probing and attacking data. It can be divided into four modules, including TCP Communication module, S7 Protocol Simulation module, Data Storage module and User Template, as shown in Figure 3.

The main function of TCP Communication module is to listen on TCP port 102, submit the received data to the Protocol Simulation module, and reply to the remote peer. S7comm Protocol Simulation module parses the received data according to the protocol format and obtains the valid contents at first. Then, S7comm Protocol Simulation module generates the reply data referring to User Template. At last, the reply data are sent back to TCP Communication module to be packaged. User Template records all the user-defined information such as PLC serial number and manufacturer. The Data Storage module handles the request and the response of data storage.

We deployed S7commTrace honeypots in United States, China, Germany, Russia, Japan, Singapore, and Korea at the same time. Each S7commTrace ran for 272 days on average. At last, we captured 110,501 requests of S7comm protocol, as shown in Table 3. In fact, not all requests are in accordance with S7comm format. Ignoring them, S7commTrace records a total of 46492 valid requests. If we define an uninterrupted TCP communication connection as a session, S7commTrace records 5797 sessions and 4224 valid sessions. Furthermore, a valid IP address indicates that this IP has at least one valid session.

According to the DNS query results, we find that there are 26 IP addresses pointing to Shodan.io, 19 IP addresses pointing to eecs.umich.edu, 16 IP addresses pointing to neu.edu.cn, and 5 IP addresses pointing to plcscan.org, as shown in Table 4. This means 573 valid IP addresses belong to four organizations at least.

Shodan.io [26] is the domain suffix of Shodan which is a search engine in cyberspace. In addition to retrieving traditional web services, Shodan has used the ICS protocol directly to crawl the ICS devices on the Internet and visualizes their location and other information. Eecs.umich.edu is the domain suffix of the Department of Electrical and Computer Science (EECS) Department of University of Michigan, which is one of the agencies developing Censys [27, 28]. Censys scans the devices in the Internet and stores the results in its database. It provides not only web and API query interfaces but also raw data to download. Neu.edu.cn is the domain suffix of Northeastern University of China which develops a search engine name as Detecting [29]. Detecting is capable of providing accurate information of ICS devices and their locations. Plcscan.org is the domain suffix of Beacon Lab [30] which is committed to the research

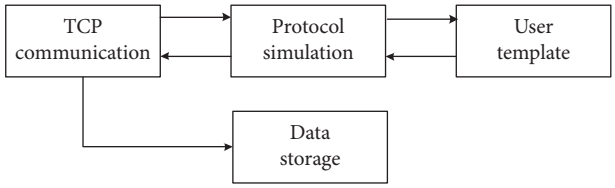

Figure 3: Structure of the ICS honeypot (S7commTrace).

Table 3: Count of all attack data and valid attack data after 13 honeypots were run for 272 days.

Item	Count
Request	110,501
Valid request	46,492
Session	5797
Valid session	4224
IP address	897
Valid IP address	573

Table 4: IP statics by DNS reverse lookup.

Domain	Organization	IP number
Shodan.io	Shodan	26
eecs.umich.edu	Censys	19
neu.edu.cn	Detecting	16
plcscan.org	Beacon Lab	5
Others	Unknown	507

and the practice related to ICS security. These four organizations are the well-known security research institutes. They are scanning the devices in the Internet all the time, including the ICS devices. Except for the 66 IP addresses belonging to four well-known organizations, there are still 507 IP addresses which are resolved to be dynamic domain name or none domain name.

3. Structure of the ICSTrace Model

When an attacker launches the attacks, he usually hides the IP address of his own resorting to the anonymous communication networks such as springboard host, VPN, and other measures. As shown in Figure 4, after an ICS suffered an attack from the Internet, the security personnel can only see the last IP address connected to ICS instead of the real IP address of the attacker, not to mention the organization which belongs to.

ICSTrace transforms the features of data from each IP address into a one-dimensional eigenvector. This eigenvector stands for the unique pattern of an attack. Therefore, the problem of attribution turns into a problem of clustering the patterns.

As shown in Figure 5, the input of ICSTrace is a malicious IP and its packets. The output is a cluster containing multiple IP addresses, which indicates an organization. ICSTrace model consists of three stages, including protocol resolution, attack pattern extraction, and partial seeded K-means clustering. The main function of protocol resolution

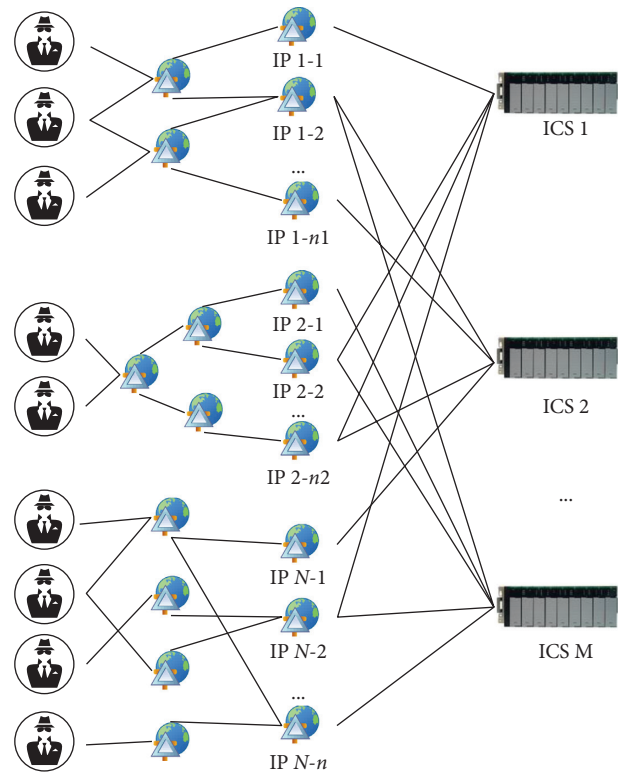

Figure 4: Schematic diagram of attacking flow.

is to parse the packets and extract the function codes and their parameters. Attack pattern extraction transforms the function codes and their parameters into one-dimensional vector as the attack pattern of a certain IP address. Partial seeded means is used to cluster the attack patterns so that those IP addresses with the same patterns are aggregated into one cluster. And then, the cluster is labeled as a certain organization according to some auxiliary information (e.g., domain name or geographical location) of the IP addresses in it.

The function codes are used to achieve the effects of operations in most industrial control protocols like S7, modbus, bacnet, and DNP3. Since the inputs of ICSTrace model are function codes and their parameters, the model is applicable for other industrial control protocols besides S7. As for the general Internet protocol, they transmit information not only by function codes and they are much more complex than the industrial control protocols. Therefore, ICSTrace model is not completely applicable for the general Internet protocol.

4. Attack Pattern Extraction

After an attacker has constructed the connection with ICS, he will carry out a series of delicate operations on purpose, which are expressed by the function codes and their parameters [22]. Therefore, the attacking features, which are extracted from the function codes and their parameters of S7comm protocol data, can reveal the intention of the attacker effectively.

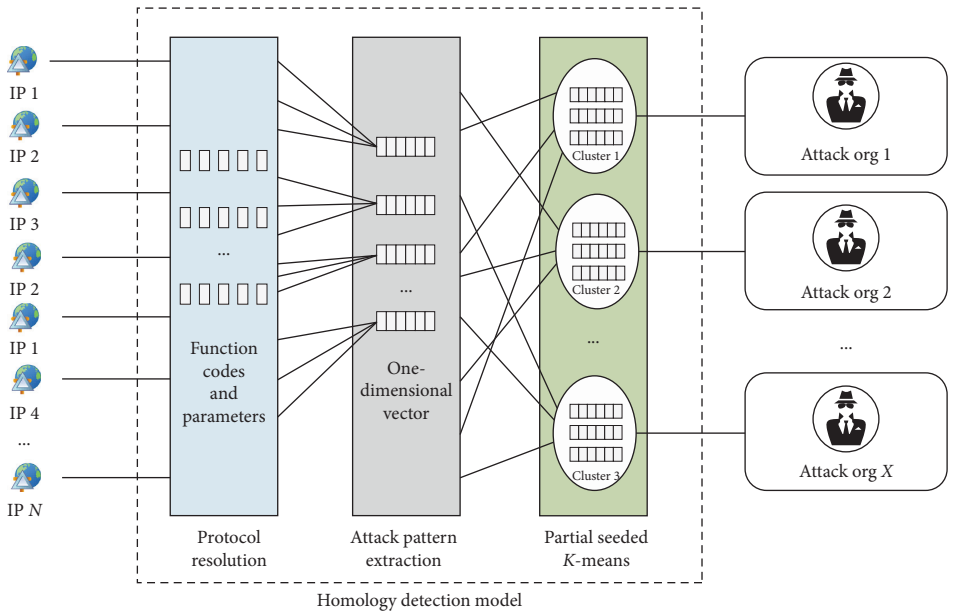

Figure 5: Structure of the ICSTrace model.

As shown in Figure 6, one attacker may have several IP addresses to launch attacks. We have defined an uninterrupted TCP communication as a session, and one IP address may attack one or more ICSs for more than one time. Thus, a single-source IP may build several sessions. We call a packet sent by the attacker as a request, and there are several packet interactions, so a session usually contains many requests.

The function codes and their parameters of S7comm protocol are included in these requests, so we extract these from the communication data package, which is sent by the attacker to the receiver, as the feature of the attacker to construct IP traceback model.

4.1. Mean Count of Function Codes and Parameters. Mean count of function codes (MCFC) refers to the average amount of the function codes of each session from the same IP address. Different attackers have different motivations, objectives, and methods while conducting a cyberattack. As a result, quantities of requests and function codes are very different in different sessions:

$$\mathrm{MCFC} = \frac{1}{n}\sum_{i=1}^{n}(\mathrm{Count\_of\_funciton\_codes})_{\mathrm{session}_i}, \quad \mathrm{session}_i \in \mathrm{IP}. \tag{1}$$

Mean count of the parameters (MCP) refers to the average amount of the parameters used in the function codes of each session from the same IP address. Some function codes do not need parameters, and some function codes need one or more parameters, so different attackers use different amounts of parameters:

$$\mathrm{MCP} = \frac{1}{n}\sum_{i=1}^{n}(\mathrm{Count\_of\_parameters})_{\mathrm{session}_i}, \quad \mathrm{session}_i \in \mathrm{IP}. \tag{2}$$

4.2. Function Code Sequence and Parameter Sequence. Function code sequence (FCS) indicates the change rule of the function codes in all sessions from a single IP address. Different attackers may use the same kind of function codes while lunching an attack, but the chronological order is different. As shown in Figure 7, the Function code $C_1, C_2, \ldots, C_i$ can be arrayed to form a Markov chain in chronological order.

Array the function codes in the session to form a function code sequence according to the chronological order:

$$F_{\mathrm{session}_i} = (C_1, C_2, \ldots, C_i), \quad \mathrm{session}_i \in \mathrm{IP}. \tag{3}$$

For some sessions may belong to the same source IP address, we combine the function codes serials and parameter serials of all sessions from the same IP address into a set of function code sequence:

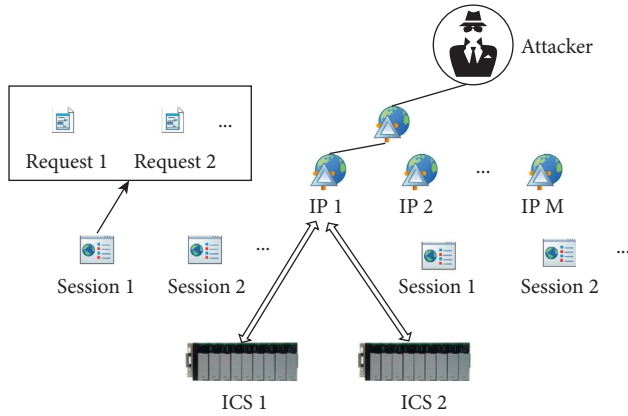

Figure 6: Schematic diagram of attack IP session request.

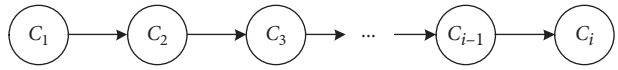

Figure 7: Function code sequence.

$$F_n = \begin{pmatrix} F_{\text{session}_1} \\ F_{\text{session}_2} \\ \cdots \\ F_{\text{session}_n} \end{pmatrix} = \begin{pmatrix} C_1 & C_2 & \cdots & C_{a_1} \\ C_1 & C_2 & \cdots & C_{a_2} \\ \cdots & & & \\ C_1 & C_2 & \cdots & C_{a_n} \end{pmatrix}, \quad \text{session}_i \in \text{IP}, \ 1 \le i \le n. \tag{4}$$

Different amounts of sessions originate from each source IP, and various methods are adopted by the attackers for each time, which results in the different function code sequences in each session. Therefore, F_n of different source IP addresses are two-dimensional matrix vectors with unequal rows and columns.

These FCSs with uncertain amount and unequal length cannot be handled directly, for clustering algorithms like K-means need samples with same dimensions. In this study, we propose a method to convert these sequences with uncertain amount and unequal length into the vectors with the same length; the detailed process is as follows:

Step 1: add the start and the end status to the sequence.

For a sample set of sequence F_n, there are n sequences with unequal length and the length of which are $a_1, a_2, \ldots, a_n$, $a_i \ge 1$, $i \in [1, n]$, respectively. Add the start and the end status to each sequence in F_n, and then get F_n'. Now the length of each sequence is no less than 3:

$$F_n' = \begin{pmatrix} S & C_1 & C_2 & \cdots & C_{a_1} & E \\ S & C_1 & C_2 & \cdots & C_{a_2} & E \\ S & \cdots & E & & & \\ S & C_1 & C_2 & \cdots & C_{a_n} & E \end{pmatrix}. \tag{5}$$

Step 2: obtain the unrepeatable set of short sequences.

Setting the window length equals 3 and the stride equals 1, use the slide window method to process each sequence in F_n'. Then, get $a_1, a_2, \ldots, a_n$ short sequences with the same length of 3, $a_i \ge 1$, $i \in [1, n]$. Then, remove the duplicate sequences and add the short sequences into set $S = (s_1, s_2, \ldots, s_m)$, $m \le \sum_{i=1}^{n} a_i$.

Step 3: obtain the short sequences set of all sample sets.

Process all of the sequence sample sets according to Steps 1 and 2, and get a short sequence set $S = (s_1, s_2, \ldots, s_k)$ without duplication.

Step 4: express the probability vector of the sequences with uncertain amount and unequal length:

$$P_n = \begin{pmatrix} C_1 & C_2 & \cdots & C_{b_1} \\ C_1 & C_2 & \cdots & C_{b_2} \\ \cdots & & & \\ C_1 & C_2 & \cdots & C_{b_l} \end{pmatrix},$$

$$P_n' = \begin{pmatrix} S & C_1 & C_2 & \cdots & C_{b_1} & E \\ S & C_1 & C_2 & \cdots & C_{b_2} & E \\ S & \cdots & E & & & \\ S & C_1 & C_2 & \cdots & C_{b_l} & E \end{pmatrix}. \tag{6}$$

For a sequence set P_n corresponding to a certain IP, there are l function code sequences with unequal length and the lengths of them are $b_1, b_2, \ldots, b_l$, $b_i \geq 1$, $i \in [1, l]$. By adding the start and the end status to each sequence, we get P'_n. Then, process all the function codes sequences with the slide window method to construct a feature vector X_{ip} according to the frequency of these short sequences:

$$X_{ip} = \left(X_{s_1}, X_{s_2}, \ldots, X_{s_k}\right), \quad \sum_{i=1}^{k} X_{s_i} = 1. \tag{7}$$

The method for FCS feature vector processing is shown in Figure 8. We make an improvement on the short sequence processing method in literature [13]. The improved method has the following advantages: firstly, we transform the FCS with uncertain amount and unequal length from the same IP into feature vectors with the same length, and we retain the information of the function codes and their parameters resorting to the frequency characteristics of the short sequence. Secondly, when the length of the short sequence is set to 3, we can process the sequences with unequal length including the length of 1 or 2, by adding the start and the end status.

Parameters sequence (PS) indicates the change rule of the parameters in all the function codes used by the sessions from the same IP, and it is arrayed by chronological order. Similar to FCS, we use the same method to process PS.

5. Partial Seeded *K*-Means Algorithm

We have tried machine learning methods for malicious IP traceback. Commonly used machine learning methods include decision tree, SVM, and neural network, but all these methods need supervised training samples. But in the homology test of attacking data, the attack source is unknown, and therefore the sample data has no labels. Unsupervised learning can reveal the inherent nature and law of data by learning the unlabeled training samples. Clustering is the most widely used method in unsupervised learning. Clustering is to divide the data samples into multiple classes or clusters, so that the samples in the same cluster have a higher degree of similarity and the samples in different clusters are more different from one another.

K-means [31] algorithm is one of the most classical clustering methods based on partition. The basic idea is to cluster around K points as centers in space, by classifying other samples which are the closest to them. The values of each cluster center are updated iteratively until the best clustering results are obtained. In application, the clustering effect of *K*-means algorithm is greatly influenced by the initial center selection method.

Considering that the clustering performance can be improved by using labeled samples to assist the initial center selection, Wagstaff et al. [32] proposed the COP *K*-means algorithm. By constructing the two constraint sets of Must-List and Cannot-Link, the samples were constrained when they were added to clusters, but the selection of the initial center point was not constrained. Basu et al. [33] proposed seeded/constrained *K*-means algorithm. It constrained the choices of initial center through seed, and the constraint was also valid when a sample was added into a cluster. However, in this method, each cluster needs a preexisting seed.

In the IP traceback process, it is possible to know that some IP addresses belong to a certain organization. However, it is very hard to know all the organizations in advance. That means that some clusters do not have preexisting seed. Therefore, we designed a partial seeded *K*-means algorithm to solve this problem (see Algorithm 1).

Partial seeded *K*-means algorithm utilizes some sample subsets with known cluster partition (which is partial seed) as seed, to determine the initial center point. Considering there may be a variety of attack modes in an organization, constraints on seed are not applied while adding a sample into the clusters. That means the samples with known cluster partition may be classified into the original cluster or a new cluster during the process of clustering. The purpose of using partially seeded clustering here is not to determine whether the other unknown clusters are correct, but to maintain the known clusters being stable while adjusting the parameters of algorithm.

6. Evaluation

6.1. IP Recall Rate of the Known Organizations. We use the IP addresses of the four known organizations to check how many IP addresses of the same organizations are recalled in the same cluster. The four curves in Figures 9–12 show how the recall rate varies with different *K* values. Apparently, the IP addresses of Shodan, Censys, and Beacon Labs are all grouped into the same cluster, when the cluster number *K* is set between 20 and 25. However, the highest recall rate of Detecting's IP addresses is about 40%. That means Detecting's IP addresses are divided into different clusters, and there may be multiple attack modes in the samples of Detecting.

6.2. Similarity between the Predicted Value and the True Value. Given the knowledge of the ground truth class assignments labels_true and our clustering algorithm assignments of the same samples labels_pred, Adjusted Rand Index (ARI) [34] is a function that measures the similarity of the two assignments, ignoring permutations, and with chance normalization. Mutual Information is a function that measures the agreement of the two assignments, ignoring permutations. Adjusted Mutual Information (AMI) is normalized against chance [35].

We use the 66 IP addresses of the known organizations out of 573 valid IP addresses to compare the similarity between the predicted value and the true value. Figure 13 shows how ARI and AMI scores between the predicted and the true values of the 66 IP addresses vary with different *K* values. Apparently, the clustering works best when the number of clusters *K* is set between 20 and 29.

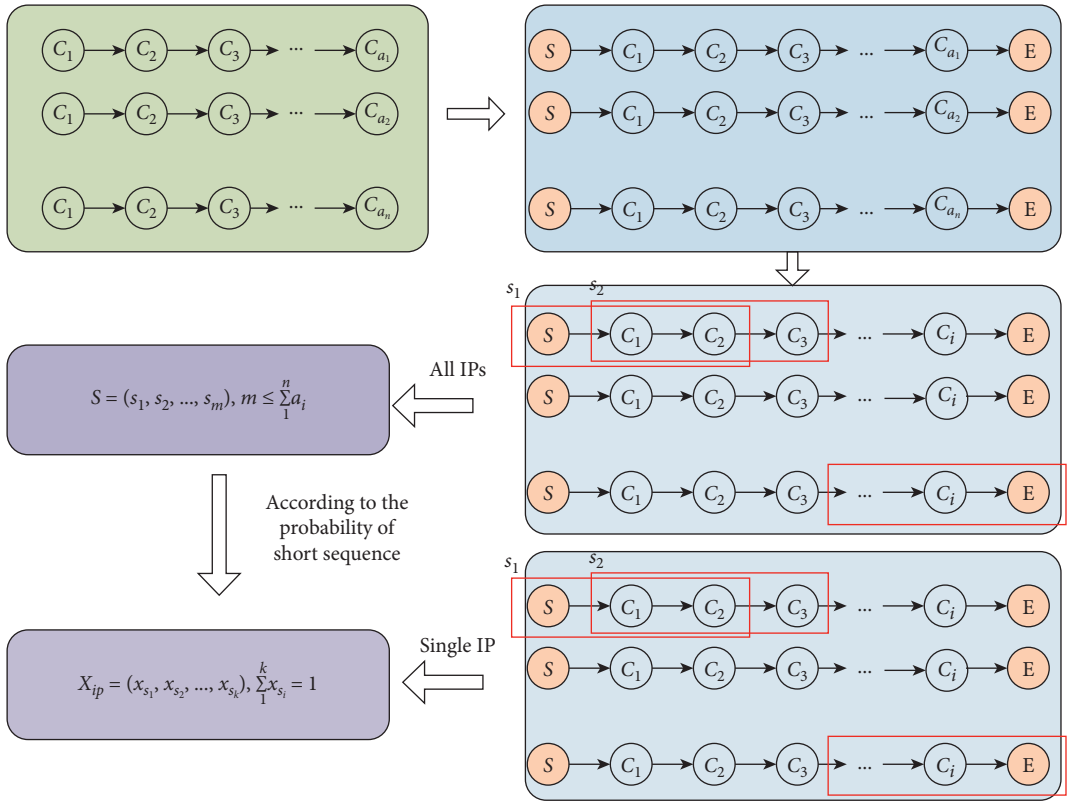

Figure 8: Method for FCS feature vector processing.

Input: given a sample set $D = \{x_1, x_2, \ldots, x_m\}$, the clustering number k, the known clustering number l, $k \le l$, the sample subset of known cluster partition $D' = \{x_1, x_2, \ldots, x_n\}$, and the sample subset of unknown cluster partition $D - D'$.

(1) Calculate the mean of the samples in each known cluster $C_i\,(1 \le i \le l)$: $\mu_i = (1/|c_i|)\sum_{x \in c_i} x$.
(2) Calculate the distance from each sample $x_j\,(1 \le j \le m - n)$ in $D - D'$ to the known mean $\mu_i\,(1 \le i \le l)$, and choose the largest value which equals mean distance added minimum distance as the new initial mean μ_{l+1} and let μ_{l+1} as known mean.
(3) Repeat Step 2, until $k - l$ samples are chosen as the initial mean vector $\{\mu_{l+1}, \mu_{l+2}, \ldots, \mu_{l+k}\}$, make $\mu_i\,(i \le i \le l)$ and $\{\mu_{l+1}, \mu_{l+2}, \ldots, \mu_{l+k}\}$ to be the initial mean vector with k means.
(4) Calculate the distance $d_{ij} = \|x_j - \mu_i\|_2$ which is from each sample $x_j\,(1 \le j \le m - n)$ in $D - D'$ to each mean vector $\mu_i\,(1 \le i \le k)$.
(5) Choose the cluster label for the sample x_j according to nearest initial vector $\lambda_j = \text{argmin}_{i \in 1,2,\ldots,k-l} d_{ji}\,(1 \le j \le m - n)$, and add x_j into corresponding cluster $C_{\lambda_i} = C_{\lambda_i} \cup \{x_j\}$.
(6) Calculate new mean vector $\mu_i' = (1/|c_i|)\sum_{x \in c_i} x$, if $\mu_i' \ne \mu_i$ and update μ_i to μ_i'.
(7) Repeat Steps 4–6, until no mean vector to update.

Output: cluster partition $C = \{C_1, C_2, \ldots, C_k\}$.

Algorithm 1: Partial seeded K-means.

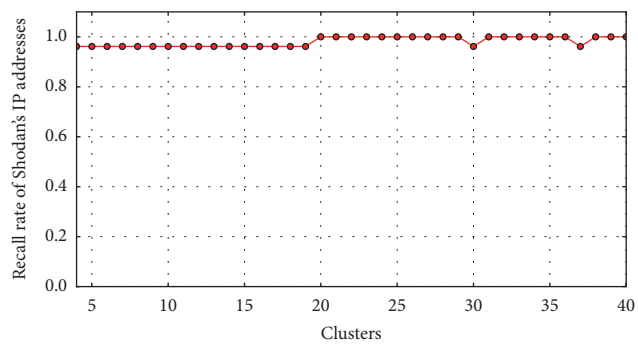

Figure 9: The recall rate of Shodan's IP addresses.

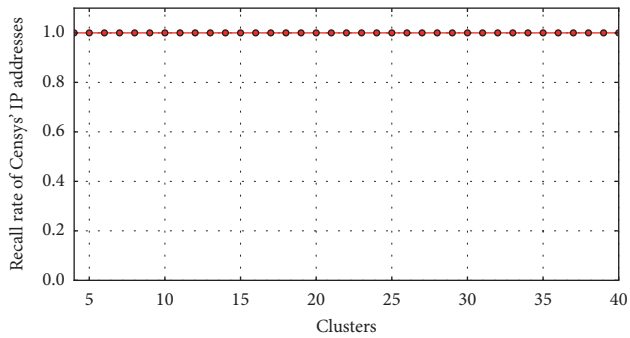

Figure 10: The recall rate of Censys' IP addresses.

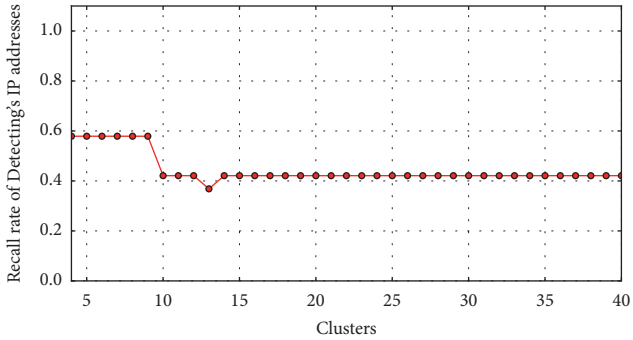

Figure 11: The recall rate of Detecting's IP addresses.

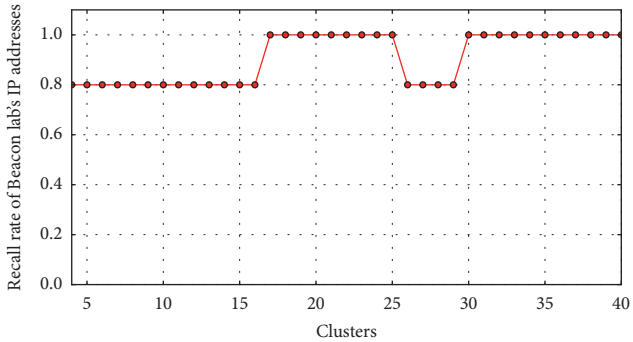

Figure 12: The recall rate of Beacon Lab's IP addresses.

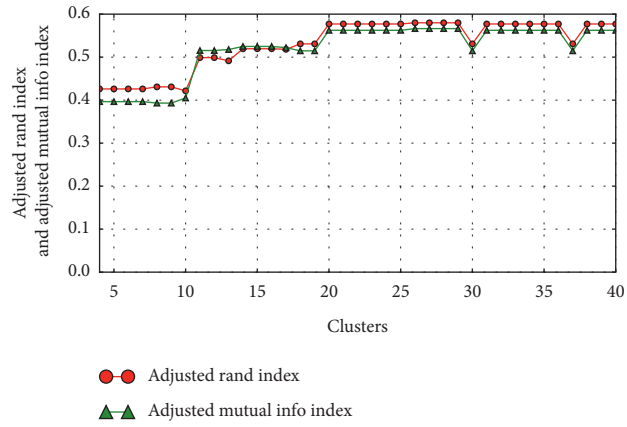

Figure 13: ARI and AMI scores between the predicted and the true values of the 66 IP addresses vary with different K values.

6.3. Clustering Performance. In the previous sections, we have evaluated the clustering effect using the samples with known labels. If the ground truth labels are unknown, evaluation must be performed using the model itself. The Silhouette Coefficient [36] is an example of such an evaluation, where a higher Silhouette Coefficient score relates to a model with better defined clusters. Calinski-Harabasz index [37] can be used to evaluate the model too, where a higher Calinski-Harabasz score relates to a model with better defined clusters.

Figures 14 and 15, respectively, show the curves of Silhouette Coefficient score and Calinski-Harabasz score, when the number of clusters K is set differently. Apparently, the clustering works best when K is set to 20.

6.4. Attack Pattern Recognition. Figure 16 shows the total number of clusters, in which those IP addresses of the four known organizations are grouped. No matter what value K is set, the maximum number of clusters is always 6. It indicates that there are only 6 attack patterns at the most in the samples with known organization labels.

The attack pattern of Shodan, Censys, and Beacon Lab is unique, when the cluster number K is set between 20 and 25. But Detecting's attack mode is not unique. All the IP addresses of Detecting belong to three different clusters, except that four IP addresses are labeled as Shodan and two IP addresses are labeled as Censys. The specific distribution of these IP addresses is shown in Figure 17.

6.5. Organization Identification. We set the cluster number K to be 20 for clustering and get 20 clusters at last. That means we find 20 kinds of attack patterns. However, these 20 attack patterns do not indicate that there are 20 organizations because an organization may have multiple attack patterns, and some different organizations may also share a common attack pattern. The DNS query results and the geographical locations of IP Addresses are helpful to identify the organizations. If the IP addresses in a cluster point to the same static domain name or they are very close geographically, we can name this cluster with these labels.

As shown in Table 5, there are 20 clusters with no less than 9 IP addresses in each of them. According to the DNS query results, some IP addresses in clusters 1, 2, 3, and 4 point to a static domain name, and some IP addresses in the clusters 11, 14, and 17 point to a dynamic domain name. There is no domain name for reference in clusters 15, 18, 19 and 20. However, they are located in a particular country or a region, so we can name these clusters with the geographical labels. Furthermore, clusters 3 and 13 are labeled as Detecting, which confirms the existence of multiple attack patterns in a single organization.

7. Related Work

7.1. ICS Intrusion Detection. Khalili and Sami [8] have proposed the SysDetect, which is a systematic approach to Critical State Determination, to solve the problem of determining the critical states in the state-based intrusion detection. This system built a well-established and iterative data mining algorithm, that is, Apriori. Kwon et al. [9] have proposed a novel behavior-based IDS for IEC 61850 protocol using both statistical analysis of traditional network features and specification-based metrics. Yang et al. [10] have presented a rule-based IDS for IEC 60870-5-104 driven SCADA networks using an in-depth protocol analysis and a Deep Packet Inspection (DPI) method. McParland et al. [11] have proposed the characteristic-based intrusion detection, which is an extension of the specification-based method, by defining a set of good properties and looking for behavior outside those properties. A specification-based intrusion detection model is designed to enhance the protection from both outside attacks and inside mistakes through combining the command sequence with the physical device sensor data. Mo et al. [12] have developed the model-based techniques which are capable of detecting integrity attacks on the sensors of a control system. It is assumed that the attacker wishes to disrupt the operation of a control system in steady state, to which end the attacker hijacks the sensors, observes and records their readings for a certain amount of time, and repeats them afterward to camouflage his attack. The model-based techniques can effectively prevent such attacks. Shang et al. [13] have presented PSO-SVM algorithm which optimizes parameters by advanced Particle Swarm Optimization (PSO) algorithm. The method identifies anomalies of Modbus TCP traffic according to appear frequencies of the mode short sequence of Modbus function code sequence. Zhou et al. [14] have designed a novel multimodel-based anomaly intrusion detection system with embedded intelligence and resilient coordination for the field control system in industrial process automation. In this system, a multimodel anomaly detection method is proposed, and a corresponding intelligent detection algorithm is designed. In addition, in order to overcome the shortcomings of anomaly detection, a classifier based on intelligent hidden Markov model is designed to distinguish the actual attacks and failures.

7.2. IP Traceback. Savage et al. [16] have described a general purpose traceback mechanism based on probabilistic packet marking. Routers probabilistically mark packets with partial path information when they arrive. By combining a modest number of such packets, a victim can reconstruct the entire path. Snoeren et al. [17] have presented a hash-based technique for IP traceback that generates audit trails for traffic within the network and can trace the origin of a single IP packet delivered by the network in the recent past. Belenky and Ansari [20] have proposed a deterministic packet marking algorithm, which only requires the border router to mark the 16-bits Packet ID field and the reserved 1-bit flag in the IP header. Therefore, the victim can obtain the corresponding entry address and the subnet where the attack source is located. This method is simple and efficient compared to Probabilistic Packet Marking algorithm. Bellovin et al. [38] have proposed an ICMP Traceback Message. When forwarding packets, routers can, with a low probability, generating a traceback message that is sent along to

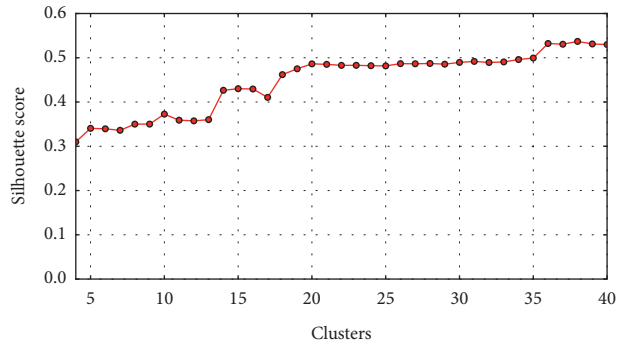

Figure 14: Silhouette Coefficient score varies with different K values.

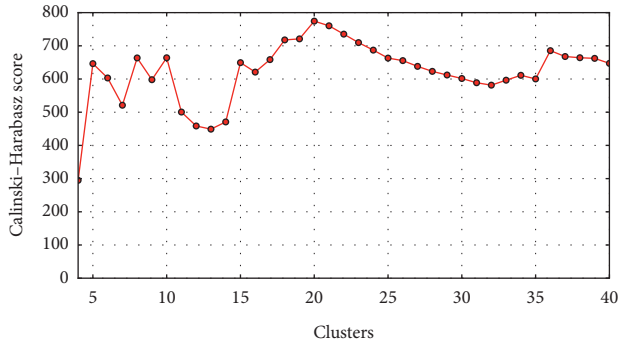

Figure 15: Calinski–Harabasz score varies with different K values.

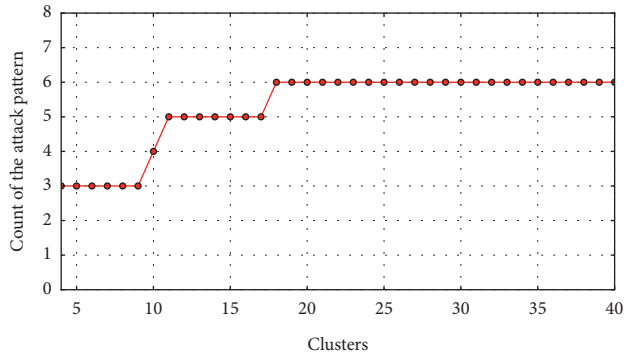

Figure 16: The total number of clusters, in which those IP addresses of the four known organizations are grouped.

the destination or back to the source. With enough traceback messages from enough routers along the path, the traffic source and path of forged packets can be determined. Goodrich [39] has presented a new approach to IP traceback based on the probabilistic packet marking paradigm. This approach, which is called randomize-and-link, uses large checksum cords to link message fragments in a way that is highly scalable, for the cords serve both as associative addresses and data integrity verifiers. The main advantage of this approach is that attacker cannot fabricate a message and it has good scalability. Gong and Sarac [18, 19] have presented a novel hybrid IP traceback approach based on both packet logging and packet marking. They maintain the single-packet traceback ability of the hash-based approach and, at the same time, alleviate the storage overhead and access time requirement for recording packet digests at routers. Their work improves the practicability of single-packet IP traceback by decreasing its overhead. Yang and Yang [40] have proposed a traceback scheme that marks routers' interface numbers and integrates packet logging with a hash table (RIHT) to deal with the logging and marking issues in IP traceback. RIHT has the properties of low storage, high efficiency, zero false positive, and zero false negative rates in attack-path reconstruction. Yu et al. [41] have proposed a marking on demand (MOD) scheme based on the DPM mechanism to dynamically assign marking IDs to DDoS attack related routers to perform the traceback task. They set up a global mark distribution server (MOD server)

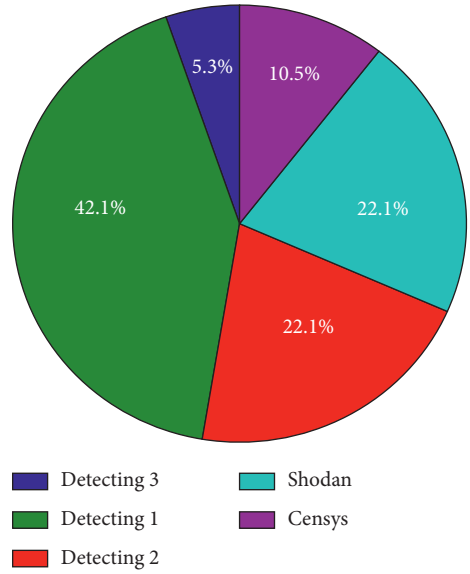

Figure 17: Distribution of Detecting's IP addresses.

Table 5: Clusters and their labels of organization.

Cluster	IP count	Auxiliary information	Organization
1	93	22 IPs are mapped to the domain name shodan.io	Shodan
2	180	14 IPs are mapped to the domain name eecs.umich.edu	Censys
3	75	8 IPs are mapped to the domain name neu.edu.cn	Detecting
4	43	5 IPs are mapped to the domain name plcscan.org	Beacon Lab
11	51	26 IPs are mapped to the dynamic domain name binaryedge.ninja	binaryedge.ninja
13	11	4 IPs are mapped to the domain name neu.edu.cn	Detecting
14	17	6 IPs are mapped to the dynamic domain name amazonaws.com	amazonaws.com
15	20	17 IPs are located in China	China Org
17	35	25 IPs are mapped to the dynamic domain name members.linode.com	linode.com
18	14	11 IPs are located in China	China Org
19	14	12 IPs are located in Europe	Europe Org
20	9	7 IPs are located in China	China Org

and some local DDoS attack detector. When there appear suspicious network flows, the detector requests unique IDs from the MOD server and embeds the assigned unique IDs to mark the suspicious flows. At the same time, the MOD server deposits the IP address of the request router and the assigned marks, which are used to identify the IP addresses of the attack sources, respectively, into its MOD database. Fadel et al. [42] have presented a new hybrid IP traceback framework. This framework is based on both marking and logging techniques. In the marking algorithm, every router is assigned a 12-bit-length ID number; it helps in deploying pushback method to permit legitimate traffic flow smoothly. In the packet logging technique, a logging ratio is managed by changing a value k specified in the traceback system. This framework can save more than 50% of the storage space of routers. Cheng et al. [43] argue that cloud services offer better options for the practical deployment of an IP traceback system. They have presented a novel cloud-based traceback architecture, which possesses several favorable properties encouraging ISPs to deploy traceback services on their networks. This architecture includes a temporal token-based authentication framework, called FACT, for authenticating traceback service queries. Nur and Tozal [44] exploit the record route feature of the IP protocol and propose a novel probabilistic packet marking scheme to infer forward paths from attacker sites to a victim site and enable the victim to delegate the defense to the upstream Internet Service Providers (ISPs). Compared to the other techniques, this approach requires less many packets to construct the paths from attacker sites toward a victim site.

8. Conclusions

IP traceback for cyberattacks usually needs redesigning the Internet deploying new service. In this study, we have proposed a malicious IP traceback model, that is, ICSTrace, for Industrial Control System without changing the Internet infrastructure or deploying any new services. By analyzing the characteristics of the attack data, we extract the numeric features and the sequence transformation features from the function codes and their parameters. Those features are expressed by a one-dimensional vector, which stands for the unique pattern of an attack. As a result, the problem of IP traceback turns into a problem of clustering those patterns.

We also propose a partial seeded K-means algorithm to cluster the IP addresses with the same pattern into a malicious organization. The effectiveness of ICSTrace is proved by experiments on real attack data. Although ICSTrace cannot recover the whole path of the attack, it is significant in the following aspects:

(1) Finding out the malicious IP addresses which belong to the same organization

(2) Revealing the unexposed active IP addresses belonging to the known organizations

(3) Collecting the anonymous communication networks used by the same organization for launching attacks

(4) Providing learning samples for subsequent malicious behavior identification by expressing the attack pattern in the form of feature vector

As we know, the concealing methods of attackers are becoming more complex. As a result, it is very difficult to trace the original IP of the attacker directly. The model proposed in our manuscript is helpful for the security experts in an indirect way of tracing the last IPs of attacks which belong to the same attacker. Therefore, we define this model as an IP traceback model.

Data Availability

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

Additional Points

Future Work. In the future, we will improve ICSTrace and apply it to other kinds of ICS protocols, even the traditional internet protocols. At the same time, we will use the attack patterns as the learning samples to design and validate the intrusion detection system based on machine learning to solve the difficult problem of unknown threat detection.

Disclosure

This work has been presented by the authors themselves as arXiv in Cornell University according to https://arxiv.org/abs/1912.12828.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors thank Biao Chang, Binglei Wang, and Dazhong Shen for their useful feedback and comments.

References

[1] T. M. Chen and S. Abu-Nimeh, "Lessons from stuxnet," *Computer*, vol. 44, no. 4, pp. 91–93, 2011.

[2] D. Kushner, "The real story of stuxnet," *IEEE Spectrum*, vol. 50, no. 3, pp. 48–53, 2013.

[3] K. Zetter, "A cyberattack has caused confirmed physical damage for the second time ever," 2015, https://www.wired.com/2015/01/german-steel-mill-hack-destruction.

[4] K. Zetter, "Inside the cunning unprecedented hack of Ukraine's power grid," 2016, http://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/.

[5] ESET discovers dangerous malware designed to disrupt industrial control systems. https://www.eset.com/us/about/newsroom/press-releases/eset-discovers-dangerous-malware-designed-to-disrupt-industrial-control-systems/.

[6] K. Stouffer, J. Falco, and K. Scarfone, *Guide to Industrial Control Systems (ICS) Security*, Vol. 800, NIST special publication, Gaithersburg, MD, USA, 2011.

[7] W. Knowles, D. Prince, D. Hutchison, J. F. P. Disso, and K. Jones, "A survey of cyber security management in industrial control systems," *International Journal of Critical Infrastructure Protection*, vol. 9, pp. 52–80, 2015.

[8] A. Khalili and A. Sami, "Sysdetect: a systematic approach to critical state determination for industrial intrusion detection systems using Apriori algorithm," *Journal of Process Control*, vol. 32, pp. 154–160, 2015.

[9] Y. Kwon, H. K. Kim, Y. H. Lim, and J. I. Lim, "A behavior-based intrusion detection technique for smart grid infrastructure," in *Proceedings of the 2015 IEEE Eindhoven PowerTech*, 2015.

[10] Y. Yang, K. McLaughlin, T. Littler, S. Sezer, and H. Wang, "Rule-based intrusion detection system for SCADA networks," in *Proceedings of the 2nd IET Renewable Power Generation Conference (RPG 2013)*, Beijing, China, 2013.

[11] C. McParland, S. Peisert, and A. Scaglione, "Monitoring security of networked control systems: it's the physics," *IEEE Security & Privacy*, vol. 12, no. 6, pp. 32–39, 2014.

[12] Y. Mo, R. Chabukswar, and B. Sinopoli, "Detecting integrity attacks on SCADA systems," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1396–1407, 2014.

[13] W. L. Shang, S. S. Zhang, and M. Wan, "Modbus/TCP communication anomaly detection based on PSO-SVM," *Applied Mechanics and Materials*, vol. 490-491, pp. 1745–1753, 2014.

[14] C. Zhou, S. Huang, N. Xiong et al., "Design and analysis of multimodel-based anomaly intrusion detection systems in industrial process automation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 10, pp. 1345–1360, 2015.

[15] T. Rid and B. Buchanan, "Attributing cyber attacks," *Journal of Strategic Studies*, vol. 38, no. 1-2, pp. 4–37, 2015.

[16] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical network support for IP traceback," *ACM SIGCOMM Computer Communication Review*, vol. 30, no. 4, pp. 295–306, 2000.

[17] A. C. Snoeren, C. Partridge, L. A. Sanchez et al., "Hash-based IP traceback," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 4, pp. 3–14, 2001.

[18] C. Gong and K. Sarac, "IP traceback based on packet marking and logging," in *Proceedings of the 2005 IEEE International Conference on Communications*, vol. 2, Seoul, Republic of Korea, 2005.

[19] C. Gong and K. Sarac, "A more practical approach for single-packet IP traceback using packet logging and marking," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 10, pp. 1310–1324, 2008.

[20] A. Belenky and N. Ansari, "IP traceback with deterministic packet marking," *IEEE Communications Letters*, vol. 7, no. 4, pp. 162–164, 2003.

[21] S7 Communication (S7comm), https://wiki.wireshark.org/S7comm.
[22] F. Xiao, E. Chen, and Q. Xu, "S7commTrace: a high interactive honeypot for industrial control system based on s7 protocol," in *Proceedings of the 2017 International Conference on Information and Communications Security*, pp. 368–380, Beijing, China, 2017.
[23] A. Jicha, M. Patton, and H. Chen, "SCADA honeypots: an in-depth analysis of Conpot," in *Proceedings of the 2016 IEEE Conference on Intelligence and Security Informatics (ISI)*, pp. 196–198, Tucson, AZ, USA, 2016.
[24] L. Spitzner, *Honeypots: Tracking Hackers*, Vol. 1, Addison-Wesley Reading, Boston, MA, USA, 2003.
[25] J.-W. Zhuge, Y. Tang, X.-H. Han, and H.-X. Duan, "Honeypot technology research and application," *Journal of Software*, vol. 24, no. 4, pp. 825–842, 2013.
[26] Shodan. https://www.shodan.io/.
[27] Censys. https://censys.io/.
[28] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, "A search engine backed by internet-wide scanning," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 542–553, Denver, CO, USA, 2015.
[29] Ditecting. http://www.ditecting.com/.
[30] ICS Security Workspace. http://plcscan.org/blog/.
[31] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.
[32] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained k-means clustering with background knowledge," in *Proceedings of the 18th International Conference on Machine Learning*, vol. 1, Williamstown, MA, USA, 2001.
[33] S. Basu, A. Banerjee, and R. Mooney, "Semi-supervised clustering by seeding," in *Proceedings of 19th International Conference on Machine Learning (ICML-2002)*, Sydney, Australia, 2002.
[34] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, 1985.
[35] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance," *Journal of Machine Learning Research*, vol. 11, pp. 2837–2854, 2010.
[36] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.
[37] T. Calinski and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics—Theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.
[38] S. M. Bellovin, M. Leech, and T. Taylor, *ICMP Traceback Messages*, Internet Engineering Task Force (IETF), Fremont, CA, USA, 2003.
[39] M. T. Goodrich, "Probabilistic packet marking for large-scale IP traceback," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 15–24, 2008.
[40] M.-H. Yang and M.-C. Yang, "RIHT: a novel hybrid IP traceback scheme," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 789–797, 2012.
[41] S. Yu, W. Zhou, S. Guo, and M. Guo, "A feasible IP traceback framework through dynamic deterministic packet marking," *IEEE Transactions on Computers*, vol. 65, no. 5, pp. 1418–1427, 2016.
[42] M. M. Fadel, A. I. El-Desoky, A. Y. Haikel, and L. M. Labib, "A low-storage precise IP traceback technique based on packet marking and logging," *The Computer Journal*, vol. 59, no. 11, pp. 1581–1592, 2016.
[43] L. Cheng, D. M. Divakaran, A. W. K. Ang, W. Y. Lim, and V. L. L. Thing, "FACT: a framework for authentication in cloud-based IP traceback," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 3, pp. 604–616, 2017.
[44] A. Y. Nur and M. E. Tozal, "Record route IP traceback: combating dos attacks and the variants," *Computers & Security*, vol. 72, pp. 13–25, 2018.