IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 9, 2020, accepted April 14, 2020, date of publication April 20, 2020, date of current version May 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2988691

Exploring Shodan From the Perspective of Industrial Control Systems

YONGLE CHEN[1], (Member, IEEE), XIAOWEI LIAN[1], DAN YU[1], SHICHAO LV[2,3], SHAOCHEN HAO[1], AND YAO MA[1]

[1]College of Information and Computer, Taiyuan University of Technology, Taiyuan 030024, China
[2]Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100089, China
[3]Beijing Key Laboratory of IoT Information Security Technology, IIE CAS, Beijing 100089, China

Corresponding author: Dan Yu (yudan@tyut.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB0803402, in part by the Natural Science Foundation of Shanxi Province under Grant 201701D111002, and in part by the Key Research and Development Program of Shanxi Province under Grant 201903D121121.

ABSTRACT As an essential component of the critical infrastructure, the Industrial Control System (ICS) is facing increasing cyber threats. The emergence of the Shodan search engine also magnified this threat. Since it can identify and index Internet-connected industrial control devices, the Shodan search engine has become a favorite toolkit for attackers and penetration testers. In this paper, we use honeypot technology to conduct a comprehensive exploring on Shodan search engine. We first deploy six distributed honeypot systems and collect three-month traffic data. For exploring Shodan, we design a hierarchical DFA-SVM recognition model to identify Shodan scans based on the function code and traffic feature, which is adapted to find the Shodan and Shodan-like scanners superior to the predominant method of reverse resolving IPs. Finally, we conduct an in-depth analysis for Shodan scans and evaluate the impact of Shodan on industrial control systems in terms of scanning time, scanning frequency, scanning port, region preferences, ICS protocol preferences and ICS protocol function code proportion. Accordingly, we provide some defensive measures to mitigate Shodan threat.

INDEX TERMS Shodan, industrial control systems, honeypot, traffic recognition.

I. INTRODUCTION

Industrial control systems (ICS) are widely deployed in critical fields such as oil and gas transportation, water supplies, and power facilities [1]. With the rapid development of industrial intelligence, a large number of industrial control devices have been connected to the Internet [2], [3]. Therefore, traditional network attacks have gradually penetrated into the industrial control field, causing serious threat to industrial control systems [4], [5]. The appearance of the Shodan search engine has further magnified this threat. In 2009, Shodan was launched by John Matherly [6], which is the first Internet-wide device search engine with a graphical user interface that can identify Internet-connected devices. Unlike traditional search engines that focus on searching web content, Shodan can identify devices with accessible IP addresses, including computers, printers, web cameras, industrial control equipment, etc. Shodan runs with 24/7 scanning and collects data on approximately 500 million Internet-connected devices each month [7]. It stores the collected device information into a searchable database that can be accessed through a web interface or Shodan API. It will make attackers easy to search Internet-connected devices from the Shodan database using a series of filters, such as countries, host names, operating systems, services and ports.

Based on the ability of identifying Internet-connected devices, Shodan can find thousands of Internet-connected ICS devices, which will bring significant security risks. CNN [8] described that Google is currently considered the most powerful search engine, but Shodan is the scariest search engine on the Internet. In fact, Shodan provides a powerful reconnaissance tool for attackers. By Shodan, an attacker can easily find information about industrial control devices exposed on the Internet, such as IP address, open services and existing vulnerabilities associated. The attacks launched based on these vulnerabilities will cause severe damage to the industrial control system.

To block the attacks from Shodan, we need to in-depth analyze Shodan scanning scheme on industrial control system. In this paper, we adopt ICS honeypot technology to

The associate editor coordinating the review of this manuscript and approving it for publication was Rongxing Lu.

capture Shodan scans. We design and deploy six honeypots on Internet simulating 4 programmable logic controllers (Modicon (BMX P34 2020), s7-400, Oles LGR25 and ABB PM573-ETH) and 4 industrial control protocols (Modbus, S7comm, IEC 60870-5-104 and BACnet). Based on honeypot data, we propose a hierarchical DFA-SVM recognition model to identify Shodan scans based on function code and traffic feature. We first use a DFA recognition model to classify the Shodan and non-Shodan scans based on the function code feature, and then design a SVM recognition model to further classify the Shodan and Shodan-like scans based on the traffic feature. In other words, we can use DFA model to find Shodan-like scanners, and further use SVM model to confirm the Shodan scanners in these Shodan-like scanners. The experimental results show our DFA-SVM model has a recognition accuracy over 95.6% for Shodan scanners. Besides, we still can find 16 new Shodan-like scanners just using DFA recognition model. Finally, we conduct a comprehensive Shodan analysis in terms of scanning time, scanning frequency, scanning port, region preferences, ICS protocol preferences and ICS protocol function code proportion. The main contributions of this paper are as follows:

- We develop a distributed ICS honeypot system to capture attack data and recognize a large amount of Shodan scan traffic from these attack data.
- We propose a hierarchical DFA-SVM traffic recognition model based on the function code and traffic features, which can improve the ability to identify Shodan and Shodan-like scans in honeypot data. We ultimately find 29 Shodan scanners and 16 Shodan-like scanners verified by threat intelligence.
- We analyze Shodan scans and evaluate the impact of Shodan on industrial control systems. We also provide some measures to mitigate Shodan threat, especially based on our DFA-SVM traffic recognition model.

The rest of this paper is structured as follows. Section 2 introduces the work related. Section 3 describes the design and deployment of our ICS honeypots. Section 4 describes in detail our hierarchical DFA-SVM model to identify Shodan traffic. Section 5 shows the Shodan analysis and blocking measures. Section 6 summarizes our work.

II. RELATED WORK

A. SHODAN APPLICATION

As a favorite reconnaissance tool for attackers, Shodan have a huge threat to the safety of industrial control systems. However, most existing works of Shodan focus on guiding how to apply Shodan to find more information. Genge *et al.* [9] design and develop a Shodan-based vulnerability assessment tool ShoVAT. Experiments are implemented on 1501 services in 12 different institutions and industries. It analyzes the service-specific data from Shodan queries and detected a total of 3922 known vulnerabilities. Phan *et al.* [10] develop an immersion visualization tool, ShodanVR, to query and display text records from an Internet-connected device in Shodan database. Ercolani *et al.* [11] utilize Gephi to visualize open port on IP addresses in Shodan and provide a method for identifying SCADA devices. By visualizations, we will have a better understanding of the devices on the Internet. So far, there are few works to analyze the Shodan scanning scheme. Bodenheim *et al.* [12] investigate the ability of the Shodan search engine. They deploy four Allen-Bradley ControlLogix programmable logic controllers (PLCs) in an Internet-facing configuration to evaluate the indexing ability of Shodan. However, all the PLCs are just exposed two services, HTTP and EtherNet/IP, and lack of detail analysis of ICS protocol scans from Shodan.

B. TRAFFIC RECOGNITION

We try to utilize the attack traffic from ICS honeypots to analyze Shodan. Although we can find the Shodan scan traffic by IPs from the *Shodan.io* domain (the domain where most Shodan scans originate), none contains a complete IPlist of all *Shodan.io* Source IPs [13]. Therefore, we survey three main kinds of traffic recognition technology to find Shodan traffic: port-based recognition, deep packet inspection (DPI) recognition, and machine learning-based recognition. Port-based traffic identification uses specific ports corresponding to network protocols and applications to identify network traffic. For example, web applications based on the HTTP protocol usually open server port 80 or 8080. Due to Shodan selects scanning port randomly, the port-based method is not adapted to Shodan traffic recognition. DPI method extracts the payload feature in the packet to identify the traffic, which can achieve high accuracy. Although the DPI method is time-consuming to resolve each packet for recognizing traffic, it is adapted to off-line analysis for honeypot data [14]. Machine learning-based traffic recognition can extract a series of payload-independent statistical features to train a traffic recognition model. Moore *et al.* [15] propose 248 statistical features and design different machine learning algorithms for traffic recognition. Bujlow *et al.* [16] utilize the C5.0 decision tree algorithm to train upstream and downstream traffic ratios for traffic identification. Alshammari *et al.* [17] use the AdaBoost algorithm to classify encrypted traffic. For encrypted traffic such as SSH and Skype, the average arrival time of upstream and downstream packets and the average size and number of data packets are identified. In [18], many algorithms, such as K-means, K-nearest neighbors, and Expectation Maximization, are compared according to their pros and cons in traffic recognition. Statistical characteristics also include packet size, uplink-downlink traffic ratio and so on. Ghofrani *et al.* [19] propose a new Internet traffic identification scheme. It discretizes the size of the first four packets of each stream based on an entropy algorithm and use KNN, SVM and Naive Bayes classifiers to label the unknown stream. Finally, the outputs of the three classifiers are combined to make the final decision based on the labels of the unknown stream. Yaguan *et al.* [20] improve the accuracy and recall rate of the traffic classifier by constructing an

TABLE 1. Honeypots deployment.

Honeypot	Location	Port
A1	US West	502 (Modbus)
A2	US East	102 (S7comm)
A3	Russia	2404 (IEC 60870-5-104)
A4	Singapore	47808 (BACnet)
A5	Brazil	502 (Modbus)
A6	China	102 (S7comm)

independent feature with the optimal discernibility of each binary SVM and training it into its own feature. However, the machine learning traffic recognition method has a limited accuracy due to the lack of application layer features. Grimaudo *et al.* [21] adopt a hierarchical classification structure and build a hierarchical self-learning DPI classification model for Internet traffic. Its recognition achieves the accuracy of traditional DPI technology, meanwhile, in combination with machine learning method to mitigate the time-consuming shortcomings of DPI technology. In this paper, we also combine the DPI and machine learning method based on the function code and traffic feature respectively to recognize Shodan scans.

III. ICS HONEYPOTS DESIGN AND DEPLOYMEN

One of the predominant methods for collecting ICS attack data is honeypot technology [22], [23]. Many honeypots have been designed and deployed for trapping IoT attracts. For example, IoTPOT [24] is an IoT honeypot to find that attacks against the increasing Internet of Things. HosTaGe is an open source low-interaction honeypot for detecting multi-stage attacks and generating response signatures. Conpot [25] is an low-interaction honeypot that can be easily deployed, modified and extended to capture ICS cyber-attacks, which is also adapted to capture Shodan ICS protocol scans.

To collect a large-scale Shodan ICS traffic data, we developed a distributed honeypot system comprising of 6 honeypots, which simulates 4 programmable logic controllers (Modicon BMX-P34-2020, s7-400, Oles LGR25 and ABB PM573-ETH) and 4 industrial control protocols (Modbus, S7comm, IEC 60870-5-104 and BACnet). Each honeypot was developed on the basis of Conpot, which can respond to corresponding ICS protocol requests and capture all interactions from attackers. Each honeypot integrates hpfeeds (an open source certified publish-subscribe protocol) and forwards the captured data to a Mongodb database. In addition, in order to make the honeypot more deceptive, we changed the hard-coded characteristics of the original Conpot honeypot, causing Shodan to misidentify our honeypot as a real industrial control system.

Considering the deployment area coverage of the honeypot, we utilize the VPSs of multiple virtual server providers such as Vultr, Aliyun, Linode to deploy six industrial control honeypots worldwide, which cover the areas including the United States, China, Singapore, Russia, and Brazil, as shown in Table 1. In this way, more countries and regions are covered to collect worldwide industrial control attack data. Each industrial control honeypot simulates only one industrial control protocol, thereby making the deployed honeypot more realistic. Each honeypot is directly accessible on the Internet without any protection in order to capture and collect more attack data.

Generally, the VPS server will enable the SSH service for users to login remotely, and the default SSH port number 22 is used. Experienced hackers can easily use the opening of port 22 to determine if target host is a VPS server or not, and then discover the honeypot. It is also possible to login the host where the honeypot is located and use brute force or other means to damage or take it as a springboard to attack other services in the network. Therefore, considering the security of the honeypot, each VPS server deploying the honeypot uses iptables to redirect the default SSH port from 22 to 40,000. Because scanning tools such as Nmap usually scan the ports within 10,000 by default. After redirecting the ports to 40,000, we can make the SSH service not be discovered easily by the scanning tools.

After collecting three-month data, a total of 145,720 traffic packets were received. In the traffic packets, we can find there is a lot of traffic including Shodan scans. Therefore, it is still a challenge how to identify the Shodan scans from all traffic.

IV. HIERARCHICAL DFA-SVM MODEL

Since the existing threat intelligence cannot contain a complete IPlist of all *Shodan.io* source IPs, we propose a hierarchical DFA-SVM recognition model to find all Shodan scanners. We first design a deterministic finite automaton (DFA) model to match the function code sequence in the payload, which stems from the fact that each of ICS protocol has a specific feature of the function code. Besides, we also find some Shodan-like scanners cannot be identified by DFA directly, for example Censys [26], PLCscan [27] or other ICS scans from research institutions. Therefore, we further propose a SVM recognition model based on traffic statistical feature to classify the Shodan and Shodan-like scans. The details are described in the following sections.

A. DFA RECOGNITION MODEL

Deterministic finite automata (DFA) is a type of automata capable of state transition. Given a state and action that belong to the automaton, it can be moved to the next state according to the transition function. Deterministic finite automata can be represented by the following quintuple $(Q, \Sigma, \delta, q^0, F)$. Where Q is a non-empty finite state set, Σ is a non-empty finite action set, δ is a transition function, q^0 is an initial state, and F is a final state set. In this paper, we define the resolving ICS function code as the action. The initial state is the first Shodan scan of a specific ICS protocol. The finite automaton is determined to start from the initial state, resolve function code one by one into a function code sequence, and move to the next state according to the given transition function. After reading the function codes, if the automaton stops at a

TABLE 2. Shodan scanning process for S7comm protocol.

	PDU-Type	Function	Function Group	Sub-Function	Packet Function
1					TCP three-way handshake
2	0xe0				Establish COTP connection
3	0xf0	0xf0			Establish S7comm connection
4	0xf0	0x00	0x44	0x01	Read system status list, request Module Identification
5	0xf0	0x00	0x44	0x01	Read system status list, request Component Identification
6					Close the connection and finish the scan

final state belong to F, it will accept the sequence. Otherwise, it will reject the sequence.

In order to match the function code sequence in ICS protocols, we set an incoming traffic of the protocol as an action and construct a series of function code sequences as the states. We take the S7comm protocol as an example to execute the DFA recognition process. Shodan scanning process for S7comm protocol is shown in Table 2. The Shodan scanner first establishes a TCP connection with the target device through a three-way handshake. Then it establishes a connection-oriented transport protocol (COTP) connection and an S7comm connection. Furthermore, it uses two interactions to read system status list requesting module and component identification. Finally, it will close the connection and finish the scanning process. We call the interaction from establishing to closing the connection as a complete interaction. Affected by network jitter, a complete interaction between Shodan scanner and honeypot may not cover the whole Shodan scanning process in Table2. We will define some function code subsequences belongs to whole Shodan scanning process as the final state of a DFA model. In order to capture more function code features, we set the subsequence contain over two function codes.

The ICS protocol function code sequence matching method is implemented by DFA serial logic judgment. According to the industrial control protocol, the function code is used to indicate the purpose of a traffic packet, that is, to indicate the function of an interaction, which is usually specified in a fixed field of the protocol data packet. The function code sequence refers to a sequence of function codes extracted from a plurality of consecutive data packets during the communication process. Since the Shodan scanning traffic sequence is more stable, we can establish a DFA model to generate some function code subsequences for each industrial control protocol. We just match these function code subsequences in DFA model with interactive packets to judge the state transition, which can distinguish whether the captured interactive packets belong to Shodan scan.

B. HIERARCHICAL DFA-SVM RECOGNITION MODEL

Although DFA recognition model can distinguish the Shodan and non-Shodan traffic, Shodan scanning scheme is similar with other IoT search engine, such as Censys, PLCscan or other ICS scans from research institutions. We need to design a new recognition model to further distinguish the Shodan and Shodan-like traffic. In this paper, we propose a hierarchical DFA-SVM recognition model to identify Shodan scans by using function code and traffic feature. We first use a DFA recognition model to filter non-Shodan scans, and then design a SVM recognition model to further filter Shodan-like scans based on the traffic feature. Rest scans belong to Shodan scans. We detail the SVM recognition model as following.

1) TRAFFIC FEATURE EXTRACTION

There are many redundant and irrelevant attributes in the network traffic, which will not only reduce the classification accuracy, but also increase the computational load of the SVM classification model. In this paper, referring to the 248 traffic statistical features proposed by Moore [15], we utilize the Relief feature selection algorithm to remove irrelevant or redundant features. The Relief algorithm is a feature weighting algorithm that assigns different weights to features based on the correlation of each feature. The feature of a weight less than the threshold will be removed. In order to balance the accuracy and efficiency of SVM recognition model, we remove the features whose feature weight is less than 0.01. Finally, we obtain 13 traffic features shown in Table 3.

TABLE 3. Network traffic feature.

	Expression	Description
1	Src_IP	Source IP address
2	Dst_IP	Destination IP address
3	Dst_Port	Destination port
4	IP_Length	IP packet header length
5	Src_Pack	Total number of packets from source to destination
6	Dst_Pack	Total number of packets from destination to source
7	Src_Bytes	The bytes from destination to source
8	Dst_Bytes	The bytes form source to destination
9	Duration	Connection duration
10	Min_IAT	Minimum packet arrival interval
11	Max_IAT	Maximum packet arrival interval
12	Mean_IAT	Average packet arrival interval
13	Var_IAT	Packet arrival interval variance

2) SVM RECOGNITION MODELING

In order to distinguish the Shodan and Shodan-like traffic, we design a classifier to identify Shodan scans based on above-mentioned traffic features. Due to the high-dimensional and non-linear characteristics of industrial control traffic, the Shodan traffic recognition model should consider these characteristics of industrial control traffic to achieve better recognition accuracy. SVM is a machine learning model with the advantages of high detection rate of small

samples and strong generalization ability, which is suitable for handling high-dimensional and non-linear Shodan traffic from a small amount of Shodan scanners.

We utilize SVM algorithm to model Shodan traffic recognition. First, based on the above-mentioned traffic statistical features, we establish a training set and a test set of recognition model, and set the algorithm model parameters to train the training set to obtain the decision function of Shodan traffic recognition model. The specific implementation steps are as follows.

1) We preprocess the sample set to construct a training set and a test set by the above-mentioned features extracted method.

2) We select the appropriate kernel function $K(x, y)$ and set the parameters of the kernel function and the penalty coefficient C, where C is the weight used to adjust the preference of the two indicators (interval size, classification accuracy) in the optimization direction, and introduce the Lagrangian function, where α is the Lagrangian multiplier vector, and x_i and y_i are the sample points. Then we construct and solve a convex quadratic programming optimization problem as follows:

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{N}\alpha_i$$
$$\text{s.t.} \sum_{i=1}^{N}\alpha_i y_i = 0, \quad 0 \le \alpha_i \le C, \ i = 1, 2, \ldots, N \quad (1)$$

3) After obtaining the optimal solution of α, $\alpha^* = (\alpha_1^*, \alpha_2^*, \ldots, \alpha_N^*)^T$, we further to calculate the b^* as follows, where b^* is a parameter of the classification hyperplane.

$$b^* = y_j - \sum_{i=1}^{N}\alpha_i^* y_j K(x_i \cdot x_j) \quad (2)$$

4) Next, we construct the optimal classification function $f(x)$ as follows:

$$f(x) = sign(\sum_{i=1}^{N}\alpha_i^* y_i K(x \cdot x_i) + \mathrm{b}^*) \quad (3)$$

5) Finally, we can use the established decision function to classify the test dataset. If it meets the requirements of training accuracy, it is our decision function of Shodan traffic recognition model. If it does not meet the requirements of detection accuracy, we will optimize the parameters and restart training process to build a new SVM recognition models.

C. DFA-SVM RECOGNITION MODEL EVALUATION

Our three-month honeypot data contains a large number of single incoming traffic without interactions. Due to our recognition model constructed by traffic interactions, we first filter these single incoming traffic data and obtain 32,522 interactive packets from all 145,720 traffic packets. In fact, each Shodan scanner can scan our honeypots many times in three months, so we still can identify all Shodan scanners even we filter out the single incoming traffic. Then we need to label the Shodan scans in 32,522 interactive packets to train our DFA-SVM model. The general approach is to obtain the domain name by reverse resolving IPs and check whether the corresponding pointer record (PTR) belongs to the subdomain of Shodan. But it is time-consuming to check 32,522 interactions, especially most of PTR values are null probably. Instead, we search and discover two open Shodan scanner IPlist on the Internet Storm Centre [28] and website Romcheckrail [29] to filter the Shodan IPs. After obtaining domain names of these Shodan IPs, we will use the NSLOOKUP tools to verify their correctness. Finally, we recognize a total of 29 Shodan scanners and 9883 Shodan interactive packets, which are used as test dataset to ten-fold cross-validation classify Shodan scans for four ICS protocols. Analyzing the 29 Shodan scanners IPs, we find that they are deployed around the world, where 16 in the United States, 7 in the Netherlands, and 2 in Romania, Iceland and Germany respectively.

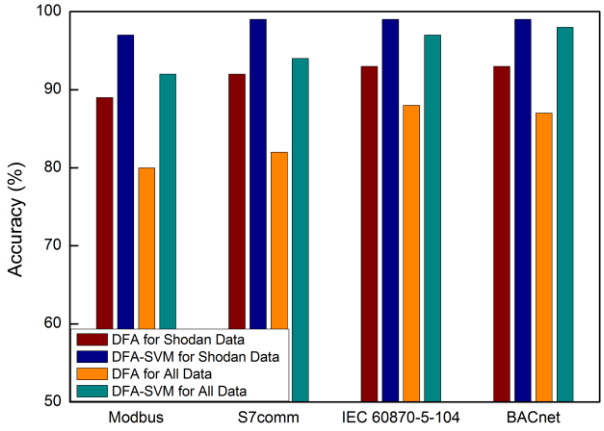


FIGURE 1. DFA and DFA-SVM recognition accuracy for four protocols.

In Figure 1, our hierarchical DFA-SVM recognition model first use DFA model to filter non-Shodan scans from the test dataset, and then use SVM recognition model further to filter Shodan-like scans. We use two datasets of 9883 Shodan interactions and all 32,522 interactions to evaluate our model. We classify the experimental data according to the protocol simulated on honeypot, and perform experiments on each protocol. The results show that average recognition accuracy of the DFA-SVM model on two datasets achieves 99.3% and 95.6%, while the recognition accuracy of the DFA model for above two datasets just achieves 91.8% and 84.5%. High recognition accuracy of our DFA-SVM model also verifies the assumption that Shodan scanning scheme was stable. Meanwhile, we can find the DFA model has a poorer performance, especially for the data of all 32,522 interactions. Because there are a larger number of Shodan-like interactions we cannot distinguish. It also indicates that these

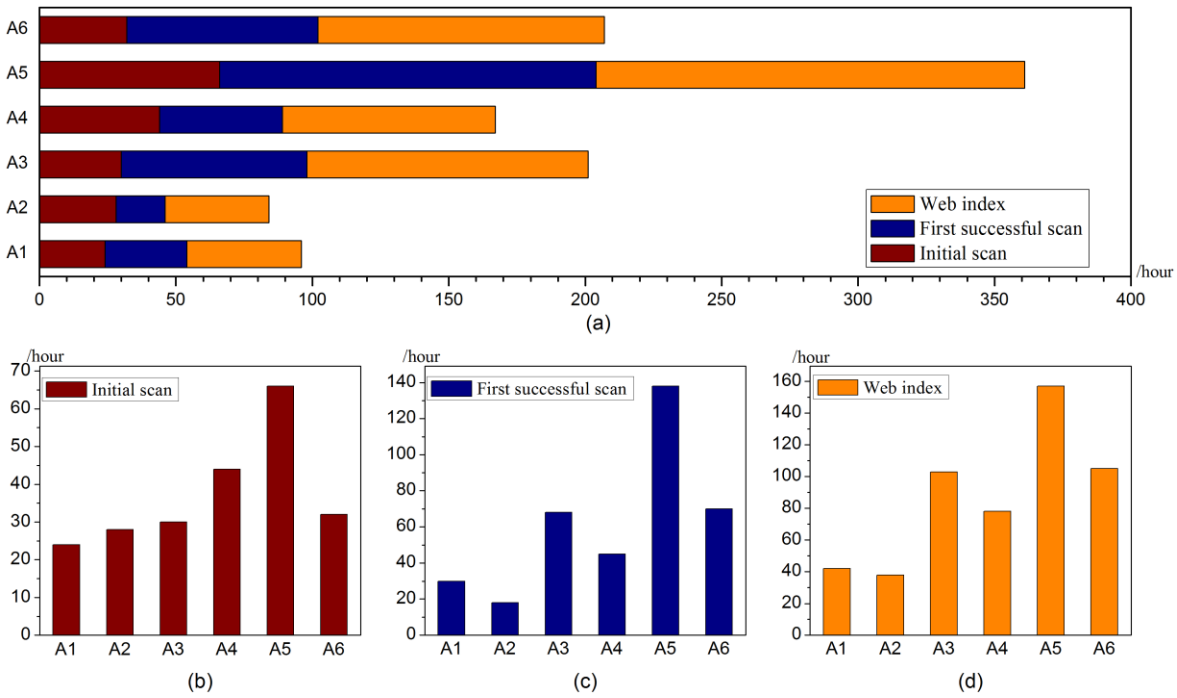


FIGURE 2. Shodan scanning time.

Shodan-like scanners have a similar scanning scheme with Shodan scanners. But recall value of my DFA method for Shodan scanners still achieves 100%. In other words, our DFA method is adapted in identifying the Shodan-like scans in all 32,522 interactions. After DFA recognizing, we find another 16 new Shodan-like scanners except for 29 Shodan scanners. We check the domain name of 16 new Shodan-like scanners by reverse resolving IPs. We find their PTR records are null so that we cannot obtain their domain name by reverse resolving IPs. Therefore, we introduce the threat intelligence method to further identify their real sources. We use IBM X-Force [30] threat intelligence and an open abuse IP database of AbuseIPDB [31] to identify the 16 new Shodan-likes scanners, where 10 belongs to Censys, 6 belongs to PLCscan which is a PLC scanning and identifying platform launched by Beacon Lab.

V. SHODAN ANALYSIS

We have described six deployed honeypots(A1-A6) in Section 3. This section performs a Shodan scans analysis for three-month ICS honeypot data. We will show the Shodan analysis results of scanning time, scanning frequency, scanning port, region preferences, ICS protocol preferences and ICS protocol function code proportion. To the best of our knowledge, Shodan can scan the Internet-scale IoT devices without interruption. In order to conceal itself, Shodan first generates an IPV4 address and a scanning port randomly, and then performs a SYN scanning. If the scan is successful, Shodan will grab the banner information and save the captured information to its database. If the scan fails, Shodan generates a random IP and port to continue a new SYN scanning.

A. SCANNING TIME

In Figure 2, we analyze the scanning time of Shodan, which contains three parts: the initial scan time, first successful scan time, and web index time. The initial scan refers to Shodan's first SYN scanning to scan a new deployed honeypot. The initial scan time is the interval from being deployed to being initially scanned for honeypots. The first successful scan refers to Shodan's first banner grabbing of target available services after receiving SYN traffic. The interval between initial scan and first successful scan is the first successful scan time. Web index is defined as the successful scanned device can be indexed by the Shodan web interface or Shodan API. After Shodan scans the Internet-connected devices successfully, the devices information will be stored in the Shodan database, but the devices information cannot be obtained immediately through the web interface or API. We call this interval as web index time.

We show the Shodan scanning time of the initial scan, first successful scan, and web index in Figure 2. From Figure 2(b), we find that A5 honeypot received the initial scan in the longest 66 hours after deployment. The A1 honeypot was scanned initially by Shodan only in the shortest 24 hours. Therefore, we confirm that Shodan scan the whole Internet one time more than 66 hours. In Figure 2(c), all honeypots are first scanned successfully by Shodan within 6 days. The A2 honeypot take the shortest first successful scan time of 18 hours, and the A5 honeypot is first scanned successfully

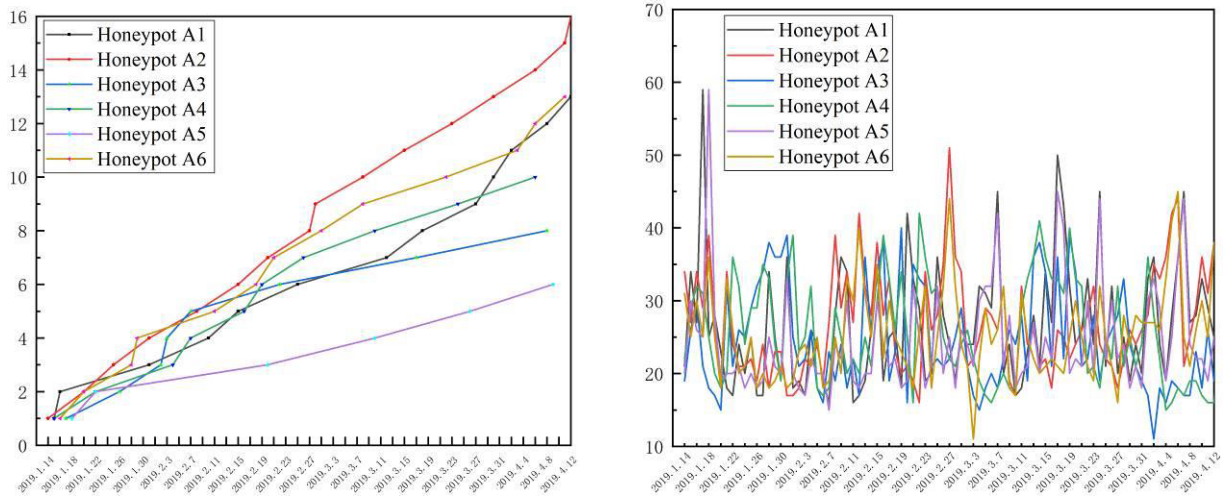


FIGURE 3. Scanning frequency.

by Shodan in the longest 138 hours. From Figure 2(d), all six honeypots are web indexed by Shodan within 7 days. Similarly, the A2 honeypot take the shortest web index time of 38 hours, while the A5 honeypot take the longest web index time of 157 hours. Finally, in Figure 2(a), we can calculate the longest time from deployment to web index is about 15 days of A5 honeypot, and the shortest time is 3.5 days of A2 honeypot. Besides, we found that both A1 and A5 honeypots simulate the same protocol of Modbus, but they have the biggest difference in scanning time. It indicates that Shodan scanning time is not relative with protocols.

B. SCANNING FREQUENCY

We also analyze Shodan's scanning frequency. As shown in Figure 3, we count the cumulative successful scans on the left and daily scans on the right of Shodan for six honeypots in three months, where the x-axis represents the scanning date, and the y-axis represents the scanning frequency of Shodan. We find that the scan frequency of Shodan is not changeless, and the number of successful scans on different honeypots is also different. In left Figure 3, the honeypot A2(S7comm) has the most successful scans up to 16, while the honeypot A5(Modbus) has only 6 successful scans in three months. It inversely relates to Shodan scanning time, that is, the lower scanning frequency leads to the longer Shodan scanning time. Meanwhile, we find that the S7comm protocol has a higher successful scanning rate, while Modbus protocol has a lower successful scanning rate. We infer that the reason is the function codes of Modbus are more complex than S7comm. Due to the average successful scans of all honeypots are about 11, we can further infer that the average successful scans interval of a honeypot is about 8 days. In right Figure 3, we can find the maximum daily scans are 59 on A1 and A5 honeypot, and the average daily scans of all honeypots are about 25. Meanwhile, we can also find that the wave changing is more similar between A2 and A6 honeypots, as well as A1 and A5 honeypots. It is because A2 and A6 simulate the same ICS protocol of S7comm, while A1 and A5 simulate the same ICS protocol of Modbus. It also indicates that Shodan scanning is service-specific strategy.

C. SCANNING PORT

We count all three-month Shodan scanning traffic to analyze scanning port distribution. Figure 4 shows the scanning port scatter diagram of 29 Shodan scanners, where the x-axis represents all 29 Shodan scanners with 3-level subdomain name and the y-axis represents the port number. We find that Shodan is prone to scan common ports within 10000. For ports over 10000, Shodan does not scan all the ports, but only scans some specific ports, such as 11211(MemCache), 27017(MongoDB) and 47808 (BACnet). The results show that the Shodan has made some restrictions on the scanning port, which is conducive to reducing the scanning time and improving search efficiency. Besides, we find that there are six Shodan scanners with the subdomains of *Shodan.io*, such

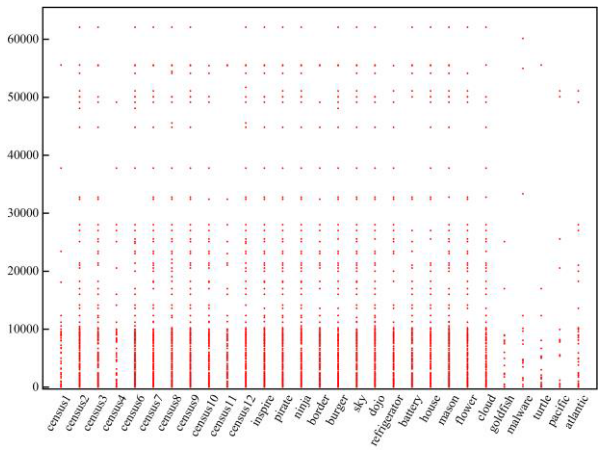


FIGURE 4. Scanning port distribution.

as *goldfish*, *malware*, *turtle*, *pacific* and *Atlantic*, including fewer scanning ports than other Shodan scanners. According to the semantics of these domain names, we infer that these Shodan scanners may have other tasks for identifying specific Internet-connected devices.

D. SCANNING REGION PREFERENCES

We use the heat map to describe the scanning frequency of six honeypots from 29 Shodan scanners. Since six honeypots are deployed in different regions, we can infer the Shodan scanning frequency in different regions. In Figure 5, 29 Shodan scanners are represented by their 3-level subdomain names of *Shodan.io*. The color bar indicates the scanning times from Shodan scanners. We can find each honeypot can be scanned by most of Shodan scanners within three months. Specifically, there are 15 Shodan scanners scanned all six honeypots, and most of other 14 Shodan scanners also scanned five honeypots. According to the deployment regions of honeypots, the honeypot A4 in Singapore receive the most Shodan scans, followed by A1 and A2 honeypots in the United States, while A3 and A5 honeypots in Russia and Brazil receive fewer scans. Besides, the honeypot A6 in China receive the largest-wide scanning from 28 Shodan scanners exclusive of *pacific.shodan.io*. It can be concluded that Shodan scanning frequency is diverse for different regions. From an intercontinental perspective, Shodan scanning frequency will be sorted as follows: Asia > North America > Europe > South America.

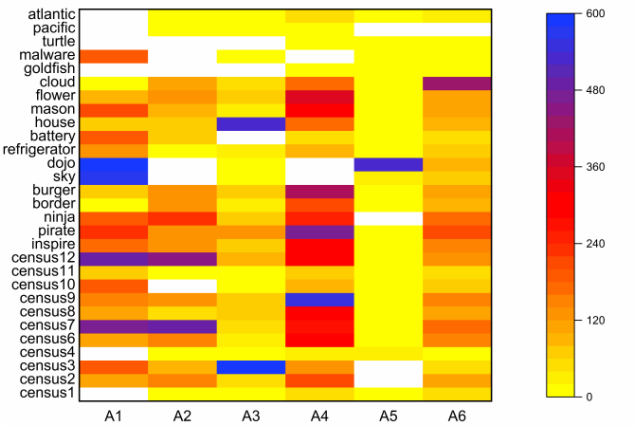


FIGURE 5. Shodan scans on each honeypot.

E. SCANNING ICS PROTOCOL PREFERENCES

In Figure 6, we show the scanning preferences of Shodan scanners for four ICS protocols. Due to the successful scans is contributed to identify honeypots for Shodan, we just select 18 Shodan scanners with successful scans to analyze scanning ICS protocol preferences of Shodan.

In Figure 6, the color bar represents the number of successful scans. There are 11 Shodan scanners without any successful scans for the four protocols, such as *census1*, *census4*, *census11*, which are not shown in the figure. We can find the *census7* is the only Shodan scanner scanned all four ICS protocols. The *census3*, *census7*, *census12*, and pirate are the Shodan scanners successfully scanned three ICS protocols. Besides, some Shodan scanners are only interested in a specific ICS protocol, such as *dojo.census* is only interested in the Modbus protocol. From these results, we also find that the successful scans of Modbus and S7comm protocols are far more than BACnet and IEC 60870-5-104 protocols. It indicates that Shodan scanners have specific scanning preference for ICS protocols.

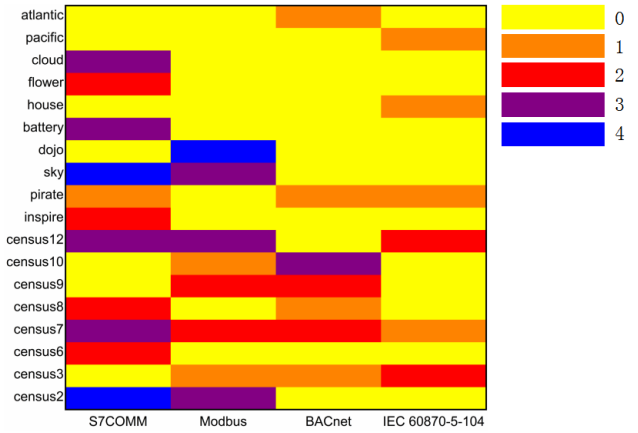


FIGURE 6. Successful scans on four protocols.

F. ICS PROTOCOL FUNCTION CODE PROPORTION

In the Shodan scans, the protocol function code used for scanning the four industrial control protocols is stable. It is an important reason why we can use the function code sequence as a Shodan scans classification feature. In Figure 7, Modbus and IEC-60870-5-104 protocols use three function codes to scan ICS devices, while S7comm and BACnet protocols use only two function codes to scan ICS devices. Figure 7 also shows the detail ICS function code proportion in Shodan scans. We can find that the function codes of each industrial control protocol are different. Before reading devices

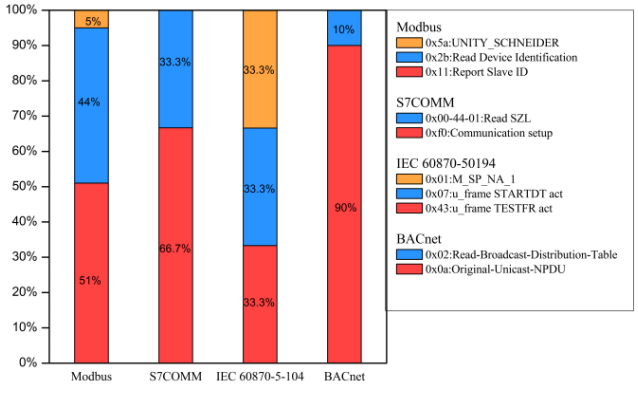


FIGURE 7. The ratio of each protocol function code in Shodan scans.

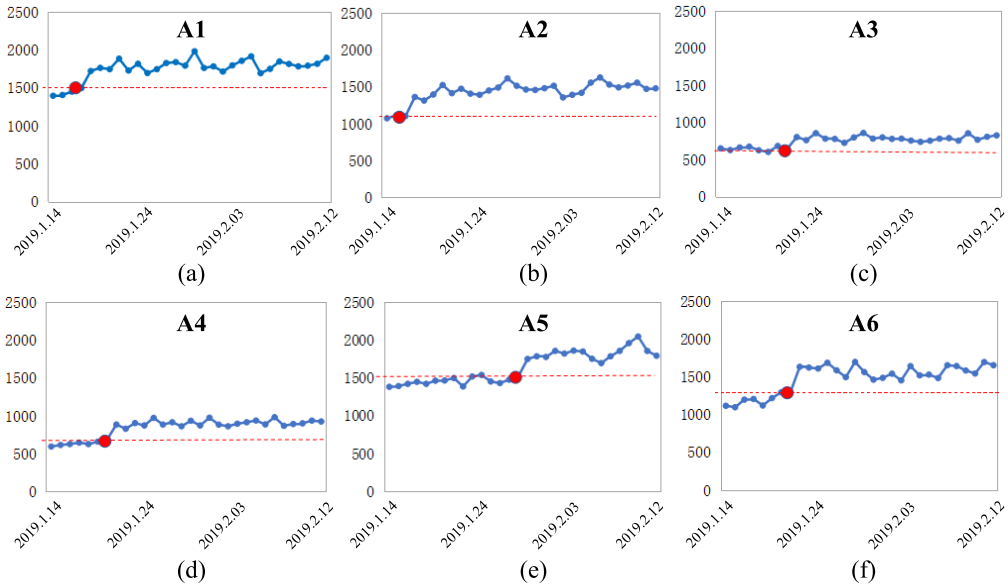


FIGURE 8. The variety in the number of attacks.

information, Shodan will perform a series of necessary preliminary function operations for different ICS protocols. For example, Modbus first report the slave ID, and S7COMM first establish communication. Moreover, the function codes of these preliminary operations account for a large proportion, especially for BACnet protocol, while the function codes of reading devices information have a lower proportion. This is also the reason why the successful scans are fewer.

G. TRAFFIC ANALYSIS OF HONEYPOTS

In Figure 8, we show the first month traffic on the six honeypots from Internet. The red dot indicates the time when the honeypot can be indexed by Shodan web interface. The x-axis represents the date of the first month, and the y-axis represents the daily traffic from the whole Internet. We can find that the traffic of our honeypots has increased significantly after indexed by Shodan. It is a possible reason that attackers exploit Shodan to capture ICS information, which indicates that Shodan has a negative impact on industrial control systems. By exploiting the Shodan web, an attacker can easily find the ICS device information exposed on the Internet, including the device IP address, open ports, geographic location, and vulnerabilities, etc., and use this information to launch an attack.

H. DEFENSIVE MEASURES

Shodan has the ability to provide Internet-connected ICS devices information for attackers, penetration testers, security professionals, and academic researchers, which make it simpler to find exploitable vulnerabilities and launch attacks. We need to take some defensive measures to prevent devices from scanning by Shodan. The intuitive measure is to disconnect the devices that are unnecessary connected to Internet, such as some PLC or DCS devices in the factory. However, with the growing of industrial Internet, more and more ICS devices will be connected to Internet for improving productivity. Therefore, we shall focus on the defensive measures of Internet-connected ICS devices. So far, the predominate measure to block Shodan scans is constructing IP blacklist. Many threat intelligences can provide and update the Shodan scanners' IPlist. We can add these IP addresses to the blacklist of the firewall to prevent Shodan scanning. But the fact is that there are none of IPlist containing a complete list of all Shodan scanners. In addition, Shodan is a banner-based devices identification tool. We can reduce the identification probability of the ICS device by modifying the banner information. However, banner information is solidified into the device by the manufacturer, it is difficult to be artificially modified by users.

In this section, we propose another Shodan defensive measure. We can design an intrusion detection system (IDS) based on our DFA-SVM traffic recognition model to block the Shodan scans. But we know the time efficiency is an important evaluation metric for industrial control system with high real-time requirements. Instead of improving time efficiency of our DFA-SVM model, we can design a bypass IDS to identify Shodan scanners IP addresses, and then update the blacklist of the firewall to block Shodan scans. Meanwhile, our DFA model also identifies the Shodan-like scanners so that our defensive measure can block both Shodan scanners and other Shodan-like scanners.

VI. CONCLUSION

In this paper, we analyze the Shodan search engine using honeypot technology. We develop a distributed ICS honeypot system and deploy it on the Internet up to three months.

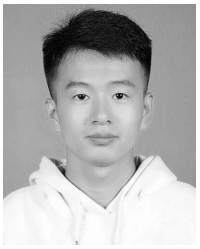
Based on honeypot data, we propose a DFA-SVM recognition model based on a combination of function code and traffic feature, and then identify 29 Shodan scanners. Meanwhile, our DFA model can identify the Shodan-like scanning so that we find 16 new Shodan-like scanners. We conduct an in-depth analysis of Shodan scans and present our analysis results in terms of scanning time, scanning frequency, scanning port, region preferences, ICS protocol preferences and ICS protocol function code proportion. Finally, we evaluate the impact of Shodan on industrial control systems and find the Internet scans on our ICS honeypots significantly increase after indexed by Shodan web interface. Accordingly, we provide some defensive measures to mitigate Shodan threat.

REFERENCES

[1] K. Stouffer, J. Falco, and K. Scarfone, "Guide to industrial control systems (ICS) security," *NIST Special*, vol. 800, no. 82, p. 16, 2011.
[2] D. Myers, E. Foo, and K. Radke, "Internet-wide scanning taxonomy and framework," *Artif. Intell. Symbolic Comput.*, vol. 161, pp. 61–65, Jan. 2015.
[3] C. Hu, W. Li, X. Cheng, J. Yu, S. Wang, and R. Bie, "A secure and verifiable access control scheme for big data storage in clouds," *IEEE Trans. Big Data*, vol. 4, no. 3, pp. 341–355, Sep. 2018.
[4] S. Samtani, S. Yu, H. Zhu, M. Patton, and H. Chen, "Identifying SCADA vulnerabilities using passive and active vulnerability assessment techniques," in *Proc. IEEE Conf. Intell. Secur. Informat. (ISI)*, Sep. 2016, pp. 25–30.
[5] A. Alrawais, A. Alhothaily, X. Cheng, C. Hu, and J. Yu, "SecureGuard: A certificate validation system in public key infrastructure," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5399–5408, Jun. 2018.
[6] Shodan. (2020). *The Shodan Search Engine*. [Online]. Available: https://www.shodan.io/
[7] R. O'Harrow, *Cyber Search Engine Shodan Exposes Industrial Control Systems to New Risks*. Washington, DC, USA: The Washington Post, 2012.
[8] D. Goldman, *Shodan: The Scariest Search Engine on the Internet*. Atlanta, GA, USA: CNN Money, 2013.
[9] B. Genge and C. Enächescu, "ShoVAT: Shodan-based vulnerability assessment tool for Internet-facing services," *Secur. Commun. Netw.*, vol. 9, no. 15, pp. 2696–2714, Oct. 2016.
[10] T. Phan, M. D. Krum, and M. Bolas, "ShodanVR: Immersive visualization of text records from the shodan database," in *Proc. IEEE Workshop Immersive Anal.*, Greenville, SC, USA, Mar. 2016, p. 31.
[11] V. J. Ercolani, M. W. Patton, and H. Chen, "Shodan visualized," in *Proc. IEEE Conf. Intell. Secur. Informat. (ISI)*, Sep. 2016, pp. 193–195.
[12] R. Bodenheim, J. Butts, S. Dunlap, and B. Mullins, "Evaluation of the ability of the shodan search engine to identify Internet-facing industrial control devices," *Int. J. Crit. Infrastruct. Protection*, vol. 7, no. 2, pp. 114–123, Jun. 2014.
[13] A. Shori, "To block or not to block? Impact and analysis of actively blocking Shodan scans," SANS Inst., Bethesda, MD, USA, White Paper, Aug. 2018. [Online]. Available: https://www.sans.org/reading-room/whitepapers/networksecurity/block-block-impact-analysis-actively-blocking-shodan-scans-38645
[14] S. H. Yeganeh, R. M. Eftekha, and Y. Ganjali, "CUTE: Traffic classification using terms," in *Proc. 21st Int. Conf. Comput. Commun. Netw. (ICCCN)*, Munich, German, 2012, pp. 1–9.
[15] A. W. Moore and D. Zuev, "Internet traffic classification using Bayesian analysis techniques," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 33, no. 1, p. 50, 2005.
[16] T. Bujlow, T. Riaz, and J. M. Pedersen, "A method for classification of network traffic based on C5.0 machine learning algorithm," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Jan. 2012, pp. 237–241.
[17] R. Alshammari and A. N. Zincir-Heywood, "Can encrypted traffic be identified without port numbers, IP addresses and payload inspection?" *Comput. Netw.*, vol. 55, no. 6, pp. 1326–1350, Apr. 2011.
[18] A. Alalousi, R. Razif, M. Abualhaj, M. Anbar, and S. Nizam, "A preliminary performance evaluation of K-means, KNN and EM unsupervised machine learning methods for network flow classification," *Int. J. Electr. Comput. Eng.*, vol. 6, no. 2, p. 778, 2016.
[19] F. Ghofrani, A. Keshavarz-Haddad, and A. Jamshidi, "Internet traffic classification using multiple classifiers," in *Proc. 7th Conf. Inf. Knowl. Technol. (IKT)*, May 2015, pp. 1–5.
[20] Q. Yaguan, G. Xiaohui, and Y. Bensheng, "Internet traffic classification using SVM with flexible feature space," *Telecommun. Sci.*, vol. 32, no. 5, Jun. 2016, Art. no. 2016132.
[21] L. Grimaudo, M. Mellia, and E. Baralis, "Hierarchical learning for fine grained Internet traffic classification," in *Proc. 8th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Aug. 2012, pp. 463–468.
[22] L. Spitzner, "Honeypots: Catching the insider threat," in *Proc. 19th Annu. Comput. Secur. Appl. Conf.*, 2004, pp. 170–176.
[23] R. McGrew, "Experiences with honeypot systems: Development, deployment, and analysis," in *Proc. 39th Annu. Hawaii Int. Conf. Syst. Sci. (HICSS)*, Jan. 2006, p. 220.
[24] Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, "IoTPOT: Analysing the rise of IoT compromises," in *Proc. 9th USENIX Conf. Offensive Technol. (WOOT)*, Washington, DC, USA, Aug. 2015.
[25] Conpot. (2020). *The Conpot Project*. [Online]. Available: http://www.conpot.org
[26] (2020). *Censys*. [Online]. Available: https://censys.io/
[27] (2020). *ICS Security Workspace*. [Online]. Available: http://plcscan.org/blog/
[28] SANS Institute (2018). *Threatlist*. [Online]. Available: https://isc.sans.edu/api/threatlist/shodan
[29] M. Hiltz. (2017). *Blocking Shodan|Keeping Shodan in the Dark From Scanning-RomCheckFail*. [Online]. Available: http://romcheckfail.com/blocking-shodan-keeping-shodan-io-in-the-dark-from-scanning/
[30] (2020). *IBM X-Force Exchange*. [Online]. Available: https://exchange.xforce.ibmcloud.com
[31] (2020). *AbuseIPDB*. [Online]. Available: https://www.abuseipdb.com/

YONGLE CHEN (Member, IEEE) was born in Weifang, Shandong, China, in 1983. He received the B.S. degree in computer science from Jilin University, in 2007, and the M.S. degree in computer science from the Institute of Software, Chinese Academy of Science, in 2009, and the Ph.D. degree in computer science from the University of Chinese Academy of Sciences, in 2013.

From 2013 to 2015, he was an Assistant Professor with the College of Information and Computer, Taiyuan University of Technology, Taiyuan, China. Since 2015, he has been an Associate Professor. His research interests include wireless sensor networks, indoor positioning, and the IoT security.



XIAOWEI LIAN was born in Jinzhong, Shanxi, China, in 1994. He received the B.S. degree in computer science and technology from Changzhi University, Changzhi, China, in 2017. He is currently pursuing the master's degree with the College of Information and Computer, Taiyuan University of Technology. His current research interests include device identification and the IoT security.

DAN YU was born in Taiyuan, Shanxi, China, in 1983. She received the B.S. degree in electronic engineering from the North University of China, in 2007, and the M.S. degree in electronic engineering from the Beijing University of Posts and Telecommunications, in 2013. She is currently pursuing the Ph.D. degree with the College of Information and Computer, Taiyuan University of Technology, Taiyuan, China. Her research interests include wireless sensor networks and the Internet of Thing (IoT).

SHICHAO LV was born in Baoding, Hebei, China, in 1985. He received the B.S. degree in communication engineering from the Liren College, Yanshan University, in 2009, and the M.S. degree in cryptography from the National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, in 2012, and the Ph.D. degree in information security from the University of Chinese Academy of Sciences, China, in 2018. He is currently a Senior Engineer with the Institute of Information Engineering, Chinese Academy of Sciences. His research interests include ICS security and the IoT security.

SHAOCHEN HAO was born in Taiyuan, Shanxi, China, in 1998. He is currently pursuing the B.S. degree in computer science and technology with the Taiyuan University of Technology, Taiyuan, China.

He is also a Bachelor Student with the College of Information and Computer, Taiyuan University of Technology. His current research interests include ICS Security and the IoT Security.

YAO MA was born in Taiyuan, Shanxi, China, in 1982. He received the B.S. and M.S. degrees in computer science from the Beijing Institute of Machinery Industry, China, in 2004 and 2007, respectively, and the Ph.D. degree in information techonology from Towson University, USA, in 2012.

Since 2012, he has been an Assistant Professor with the College of Information and Computer, Taiyuan University of Technology, Taiyuan, China. His research interests include universal usability, indoor positioning, and web security.