IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 29, 2019, accepted November 8, 2019, date of publication November 12, 2019,
date of current version November 21, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2953075

An Intelligent Communication Warning Vulnerability Detection Algorithm Based on IoT Technology

MAO YI[1], XIAOHUI XU[1], AND LEI XU[2]
[1]School of Electronics and IoT, Chongqing College of Electronic Engineering, Chongqing 401331, China
[2]Scientific Research and Social Services Department, Chongqing College of Electronic Engineering, Chongqing 401331, China

Corresponding author: Lei Xu (201430003@cqcet.edu.cn)

This work was supported in part by the 2018 Chongqing Technology Innovation and Application Demonstration Major Theme Special Project under Grant cstc2018jszx-cyztzxX0034, and in part by the Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJZD-K201803101.

ABSTRACT This paper mainly studies the vulnerability intelligent early warning technology in the IoT environment, and studies the network security assessment method based on the attack graph association analysis of the IoT environment, and analyzes the attack graph generation algorithm. Firstly, it uses the attack graph technology to establish a network security evaluation model based on the vulnerability association analysis in the IoT environment. The attack graph generation algorithm is improved. The key attack path of the attack graph in the IoT environment is searched according to the node weight value. The key attack path of the network attack graph is used to measure the whole network security, and the security under the IoT environment is given. The measurement calculation model is used to realize the quantitative analysis of the security status of the IoT environment by using the attack graph. Secondly, an intelligent early warning vulnerability detection algorithm based on the dynamic stain propagation model in the IoT environment is proposed, focusing on the introduction of stains and the inspection of stains. A static detection method for early warning vulnerabilities based on the counter-example of the IoT is proposed. Through the flow detection and context sensitive detection, a possible buffer early warning vulnerability is discovered. The driver crawler realizes automatic detection, and uses function hijacking to detect the execution of the stain data. In the experimental environment, compared with the existing tools, the experimental data shows that the algorithm improves the accuracy, recall rate and efficiency of the unfiltered vulnerability of intelligent early warning detection, and proves that the proposed algorithm can effectively detect the vulnerability.

INDEX TERMS Security measurement calculation model, IoT, intelligent early warning, vulnerability mining detection.

I. INTRODUCTION

Although today's computer network defence technologies such as security anti-virus software and network firewalls are relatively mature, with the frequent occurrence of hacker attacks on network, computer network systems are exposed with more and more security vulnerabilities. Many network anti-virus software and firewalls also Incompetent and powerless. Therefore, how to effectively discover network security vulnerabilities and reduce the negative impact of hackers and computer viruses on network security, namely vulnerability exploitation and vulnerability prevention, has become a hot topic in the world of information security [1]–[4]. In response to cyberattacks, an effective method is to use a vulnerability database that is more complete, and the vulnerability database is faster to update and monitor the information assets under its jurisdiction, eliminating hidden dangers and ensuring information security [5]–[7].

The IoT technology originated from advanced countries in the West and has matured and has been widely applied to environmental monitoring, medical systems, and intelligent control. In recent years, the Chinese government has made important plans for the development of China's IoT, and has formulated its ambitious goals [8]–[11], and widely applied

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

IoT technology to power facilities, transportation security, financial services, security and other industries. As the extension and extension of the Internet [12]–[15], the IoT (IoT) mainly realizes the information collection, transmission and processing of objects through various existing transmission means, and truly realizes the connection of objects and the exchange of information between people and things. A reflective vulnerability, that is, malicious data embedded in a page, immediately follows the request and is immediately returned from the server to the browser. The storage type can also return malicious data when accessing the vulnerability page. The biggest difference between security vulnerability is that it does not require the server to return the submitted malicious code. The browser receives the malicious data input and parses it locally. The object method and attribute of the dynamic update page cause the security vulnerability attack [16]–[19]. Another important feature of the security vulnerability is that the malicious code does not echo back in the return page source, but runs directly. When viewing the source code of the page, the original page script is seen. This attack script may not appear in Wet. page of the HTML source code [20]–[22]. Therefore, security vulnerabilities [23]–[25] cannot be detected by the method of feature matching for the above two XSS vulnerabilities, which brings challenges to automated vulnerability detection.

In view of the above problems, this paper firstly uses the attack graph technology to establish a network security assessment model based on vulnerability correlation analysis. The attack graph generation algorithm is improved. The key attack path of the attack graph is searched according to the node weight value. The key attack path of the attack graph is searched according to the node weight value. The key attack path of the network attack graph is used to measure the whole network security. The network security metric calculation model is given and the attack graph is realized. Quantitative analysis of network security status, followed by an intelligent early warning vulnerability detection algorithm based on dynamic pollution propagation model in the IoT environment, analyzing the dynamic propagation path of the stain, finding the input point and injecting the stain data, and adopting the function at the output point The method of hijacking and monitoring special functions monitors the taint data, and designs and implements the intelligent early warning vulnerability mining detection algorithm, and verifies the effectiveness of the algorithm. This paper studies the existing IoT security assessment methods. The traditional network security assessments are mostly the superposition of vulnerability risk quantification, and lack of correlation analysis of vulnerabilities in the whole network. This paper studies the existing IoT security assessment methods. The traditional network security assessments are mostly the superposition of vulnerability risk quantification, and lack of correlation analysis of vulnerabilities in the whole network. This paper studies the network security assessment method based on the attack graph association analysis of the IoT environment, and analyzes the attack graph generation algorithm. The rest of this paper is organized as follows. Section II discusses Vulnerability mining detection key technology, followed by the Intelligent Communication Early Warning Vulnerability Detection Based on IoT in Section III. Section IV shows the simulation experimental results, the fifth section summarizes the paper and proposes the future research direction.

II. VULNERABILITY MINING DETECTION KEY TECHNOLOGY

A. RESEARCH PRINCIPLE

Intelligent early warning detection technology provides network security management personnel with specific information about system vulnerabilities, and helps to formulate corresponding security policies, which can effectively prevent loopholes from being exploited by malicious attackers and causing system damage. With the development of network technology and the emergence of new types of vulnerabilities, vulnerability detection technology has also exposed various shortcomings. The existing vulnerability detection tools mostly detect the vulnerability in batches, and do not find the relationship between the vulnerabilities. The simple vulnerability risk overlay does not reflect the security status of the entire network. In addition, the vulnerability detection only detected the known vulnerability, and did not pay attention to the unknown vulnerability.

Vulnerability risk awareness warning mainly relies on vulnerability scanning technology. Vulnerability scanning technology can be generally divided into host-based security vulnerability scanning, network-based vulnerability scanning, target-based vulnerability scanning and application-based vulnerability scanning, among which host-based security vulnerability scanning And network-based vulnerability scanning is the most common vulnerability scanning technology [26]. Host-based security vulnerability scanning technology refers to the use of an agent running in a computer host system for vulnerability scanning. The technology consists of a vulnerability scanning server and a vulnerability detection agent. Host-based vulnerability scanning technology has the advantages of communication process encryption, high scanning accuracy and easy management. The network-based security vulnerability scanning technology is mainly applied to the enterprise environment. It uses different types and characteristics of security vulnerabilities. It uses network servers to generate network data packets and transmits them to multiple targets in the network in various forms of propagation. Whether are the specified vulnerability exists. The detection process of network-based vulnerability scanning is similar to the ''bottoming'' work before the actual attack by the hacker on the attacking site. The security administrator or network administrator actively implements the security vulnerability scanning to detect and analyze the security threats or security existing in the computer system [27]–[29].

As shown in Figure 1, the basic process of vulnerability awareness warning is divided into the following steps:

TABLE 1. New host environment evaluation factors.

Metric	Evaluation	Optional value	Evaluation value
ServerType	Host service	Ordinary client/business machine/server	0.5/0.8/1.0
OS Type	Host operating system	Unix system / Linux system	0.5/0.8/1.0
Parameters	α	The weight of Impact evaluation	
	β	The weight of the exploitability evaluation in the basic evaluation	0.4
	γ	the weight of the basic evaluation for the Host Environment	0.2

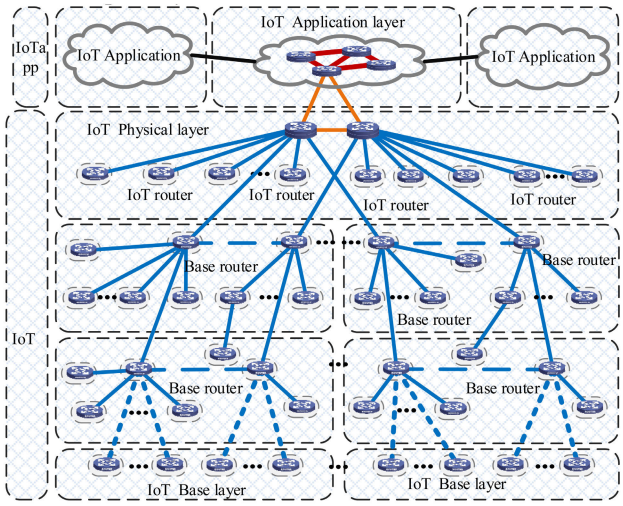


FIGURE 1. IoT early warning vulnerability detection framework.

(1) The information system vulnerability scanning engine actively probes the target host or network connection device, and collects related information and working status;

(2) The information system vulnerability scanning engine opens the network port detection module, collects and organizes the working network connection device port or the target host system to work in a sunny state, and obtains network information in real time;

(3) The information system vulnerability scanning engine uses the vulnerability detection technology to open the security vulnerability detection information to the target network device or the host system and wait for and receive feedback from the target system;

(4) If the information system vulnerability scanning engine receives the feedback from the target host and compares it with the information in the vulnerability database, if the matching is successful, it can verify that the target system has a security hole;

(5) The information system vulnerability scanning engine sends the detection result and the disposal suggestion to the vulnerability situation-aware early warning system after the completion of the detection, and the system matches the guided asset library to realize the vulnerability detection alarm and situational awareness warning of the information system.

Security administrators can use the vulnerability-aware early warning system to discover open ports and services, system configuration information, known security vulnerabilities, and related protocols in the target host system or network system to efficiently detect potential security risks in the target host system. Security scanning technology generally provides users with the ability to discover the existing security breaches, but it cannot prevent hackers from exploiting unknown vulnerabilities. Therefore, security scanning technology needs to cooperate with vulnerability assessment system to provide enterprises with systems. The ability of vulnerability assessment, although not completely solving the problem of security vulnerabilities, can promote the generation of new security patches to a certain extent.

B. RESEARCH ON VULNERABILITY MINING DETECTION TECHNOLOGY

Through the comparison and research on some existing evaluation methods of security vulnerabilities, it is found that the difficulty in evaluating the risk level of existing vulnerabilities is mainly the quantification of evaluation factors. The evaluation methods of the above analysis do not have too many evaluation factors. Too strong to quantify, these methods are not very practical in an automated vulnerability assessment system [30]–[34].

Foundation Evaluation: Add the Host Environment evaluation factor to the basic evaluation, as

shown in Table 1, which includes two evaluation elements: Server and OS Type.

The Host Environment evaluation elements are described in detail as follows:

1) Service Type (Server Type)

The service category is to evaluate some services provided by the host, which directly affects the importance and possession status of the host in the network. The service

TABLE 2. Security vulnerability evaluation elements.

Metrics	Evaluation	Score
Attack route	Remotely	0.71
Attack complexity	No need	0.704
Certification	part of	0.275
Confidentiality	part of	0.275
Pear	part of	0.275
Availability	WWW server	Score
Service type	Windows and other systems	0.71
Influence operating system	low	0.704

provided by the host is divided into three levels. If the host is a WWW server, an FTP server, a TELNET server, or an SMTP server, these services involve a large amount of data storage and some user information, and the security is the highest, the level is the highest; if the host is a service terminal, the service data information is often processed. Data confidentiality and integrity are also required to be higher, then the level is medium; if the host is a normal client, does not involve some important data processing and communication, the level is low.

2) Impact on operating system type (OS Type)

The basic evaluation, the new evaluation and the environmental evaluation are respectively scored. The scores of the several are given to give the final score of the vulnerability. Finally, according to the vulnerability evaluation score, the vulnerability risk is determined. The higher is the score, the more dangerous the vulnerability, and the lower the score, the threat of the vulnerability.

Since vulnerabilities have platform dependencies, the Windows operating system has more than 90% of users. The number of vulnerabilities found under Windows is the highest, which is worse than other operating systems. Other non-windows systems, such as Linux, have relatively high security.

The basic evaluation formula is as follows:

$$\begin{aligned} V_B &= (\alpha * I + \beta * E + \gamma * HS - \delta) * f(I) \\ I &= \lambda * (1 - (1 - I_C)) * (1 - I_i) * (1 - I_A) \\ E &= u * AV * AC * Au \\ HS &= v * ST * OT \end{aligned} \tag{1}$$

$$f(I) = \begin{cases} 0, & I = 0 \\ 1.176, & I \neq 0 \end{cases} \tag{2}$$

Among them, α is the weight of Impact evaluation, which is based on empirical analysis and calculation.

β is the weight of the exploitability evaluation in the basic evaluation, and the general value is 0.4 based on empirical analysis and calculation;

γ evaluates the weight of the basic evaluation for the Host Environment, based on empirical analysis and calculation of the general value of 0.2;

The available attack code probabilities for a given vulnerability are calculated by the Pareto distribution:

$$TE = 1 - (\frac{k}{x})^{\alpha}, \quad \alpha = 0.26, \quad k = 0.00161 \tag{3}$$

where x is the time from the discovery to the present (days), and the available patches were proposed by Frei conform to the Weibull distribution:

$$RL = 1 - \exp(-\frac{x}{\lambda})^{k}, \quad \lambda = 0.209, \quad k = 4.04 \tag{4}$$

As shown in Table 2, identify non-standard ports and accurately scan for service vulnerabilities. In the security management of IT systems, it is often encountered that the default application service port is changed due to business needs. Changing the default port of the protocol can avoid business conflicts, reduce equipment investment, and make full use of resources, but some protocols are on non-standard ports. How to identify and scan has also become a problem that security management products need to solve. The vulnerability situational awareness early warning research adopts advanced non-standard port identification technology and rich protocol fingerprint database, which can quickly and accurately identify the application service types on non-standard ports, and further detect the vulnerability, which greatly avoids the scanning process.

C. VULNERABILITY MINING SYSTEM AND VULNERABILITY MINING BASED ON IoT TEST TECHNOLOGY

From the perspective of security vulnerabilities, security issues caused by protocols can be classified into many types, including denial of service vulnerabilities, buffer warning vulnerabilities, cross-site scripting vulnerabilities, information disclosure vulnerabilities, code injection vulnerabilities, encryption problems, boundary condition vulnerabilities, and so on. How to effectively exploit these vulnerabilities and take corresponding remedial measures against these vulnerabilities is one of the important means to ensure the security of communication protocols and data security. The detection process of intelligent communication early warning security vulnerabilities based on the IoT is as shown in Figure 2:

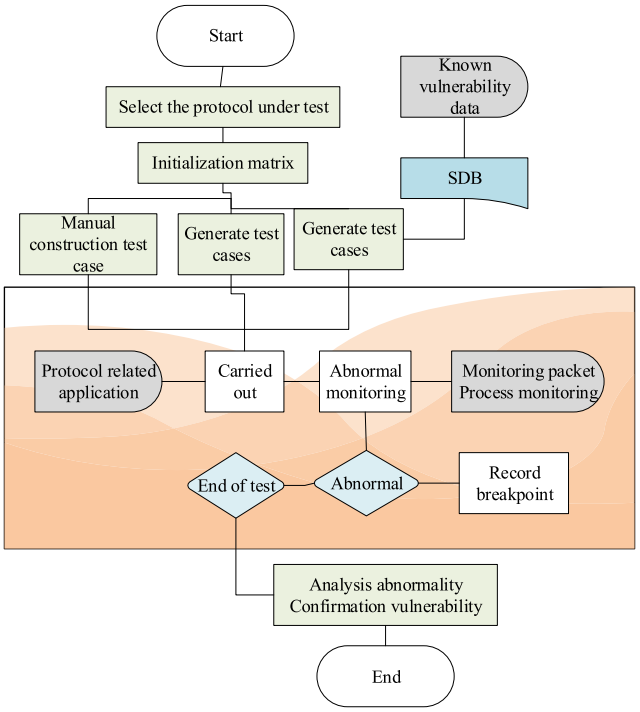


FIGURE 2. Intelligent early warning detection flow chart.

III. INTELLIGENT COMMUNICATION EARLY WARNING VULNERABILITY DETECTION BASED ON IoT

A. VULNERABILITY DETECTION ATTACK PATTERN ALGORITHM FOR IoT

Through the analysis of the vulnerability attack graph model in the IoT environment, the attack graph has the basic characteristics of the directed graph. The establishment of the attack graph can refer to the traversal method of the directed graph. The traversal of the directed graph mainly includes two algorithms: depth-first search traversal and breadth-first search traversal, and the traversal of the attack graph is to acquire all existing state relationships, that is, to

obtain the in-degree association node and the out-degree association of each node. Node, therefore, the traversal of the attack graph is more suitable for the breadth-first search traversal algorithm.

1) VULNERABILITY MINING DETECTION ATTACK GRAPH BREADTH-FIRST SEARCH ALGORITHM

The forward search algorithm starts from the initial state node and searches for the next state node that can be attacked. When there is no new available node, the search stops and the current search direction ends. The forward search algorithm does not have a certain target state from the perspective of the attacker, so the search range is relatively broad, and the generated attack graph size is also large. From the perspective of protecting specific resources, the backward search algorithm first determines the targets that may be attacked in the network system, that is, searching backwards from the target state to find the previous state that leads to the target state, searching upwards in turn, and finally reaching the initial state node or Terminates beyond the maximum number of attack hops. The backward search algorithm has a clear target, and the state nodes that are not related to the target state are excluded. Therefore, the generated attack graph is small in scale, and only shows the attack graph sequence that attacks the specified target, which can help security managers concentrate on these attack paths.

In some complex network environments, the important resources in the IoT environment are distributed and distributed. Especially in recent years, the development of cloud computing technology, the distributed storage of network resources is very common. This makes the attack target uncertain and diversified. The forward search algorithm from the attacker's point of view, although the traversal is wide and the generated attack map is more complex, it can all the possible attack sequences and attack targets in the network. The reverse search algorithm cannot give all the attack sequences in the network, but it can simplify the attack graph for the specified attack targets and give the security status of the network resources that the managers are most concerned about.

2) ATTACK GRAPH GENERATION ALGORITHM BASED ON FORWARD-BACKWARD SEARCH

1) Enter the attack node and initialize the attack queue.

2) Take a host as the attack initiation node, use the forward search algorithm to search for adjacent attack nodes, and find the attack path. Each time a node that can successfully infiltrate is found, the node is added to the current attack sequence.

3) If the attack sequence reaches the target node or the number of attack hops exceeds the set maximum value, the sequence is searched backwards to remove redundant nodes and extraneous nodes, and the current attack sequence is reduced.

4) If the target node is not reached and the number of attack hops is less than the set maximum value, loop 2 is performed until all nodes in the attack queue complete the search.

Calculate the PR level value of the current webpage, calculate the PR value of each page pointing to the current webpage, and determine the weight of the current page by the access link weight and the access link. The more the link access or the access webpage the higher the PR rating, the more important the current web page and the higher the PR rating.

$$PR(W) = (1-d)/N + d * (PR(W_1)/C(W_1) + \ldots PR(W_n)/C(W_n)) \quad (5)$$

The calculation model is applied to the attack graph state node weight calculation, and the calculation formula is as follows:

$$R(S) = (1-d)/N + d * (R(S_1)/C(S_1) + \ldots + R(S_n))/C(S_n)) \quad (6)$$

where: N is the number of all state nodes in the attack graph R(S) represents the weight of the attack graph state node S R(Si) indicates that the degree arc points to the S of the state node S, and the weight of the node C(Si) represents the state node S, the number of exit arcs. d is the damping coefficient, $0 < d < 1$, generally 0.85.

The risk assessment of critical attack paths calculates the average loss of the critical attack path. Let the probability that the state node S_i penetrates into the next state node S_{j+1} is $P_{i,j+1}$, then the probability of successful attack from the initial state node to the state node S_i is:

$$P_{0,i} = \prod_{k=0}^{i-1} P_{k,k+1} \tag{7}$$

Let the loss of the state node S_i be L_i, then the average loss on the critical attack path of the attack graph is:

$$L = \sum_{i=0}^{n} (L_i, P_{0,J}) \tag{8}$$

L is the risk value of the critical attack path of the attack graph. It also serves as a metric for the security of the entire network. It provides quantitative reference indicators for the security status of the target network, and guides the security management personnel to deploy efficient and accurate security policies.

B. DYNAMIC MODEL BASED ON INTELLIGENT COMMUNICATION EARLY WARNING VULNERABILITY DETECTION

Early warning vulnerabilities in the IoT environment are closely related to receiving external input data. The use of unverified external input data by web applications is the key to generating vulnerabilities. External input data is also known as tainted data. The process of tracking the propagation and utilization of taint data is called stain propagation analysis.

Stain propagation analysis can usually be divided into static stain propagation analysis and dynamic stain propagation analysis. Static analysis relies on source code, and needs to extract syntax and semantic features. In many cases, source code is difficult to obtain, and there is a problem of high false positive rate. Dynamic stain propagation analysis includes key technologies such as automatic marking of stains, dynamic propagation and monitoring of stains, and key data recording. When the program is executed, it immediately detects whether the data is contaminated and executed. The detection accuracy is high, and the detection process does not require the program source. Therefore, this paper adopts the method of dynamic stain propagation analysis in the process of intelligent early warning vulnerability mining detection, which mainly involves the dynamic propagation and monitoring technology of stains.

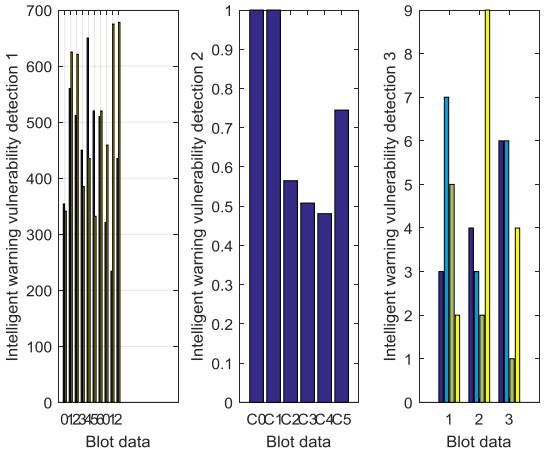


FIGURE 3. Stain data dissemination intelligent communication early warning vulnerability detection.

1) DYNAMIC STAIN PROPAGATION MODEL AND INTELLIGENT EARLY WARNING VULNERABILITY DETECTION

According to the above analysis, if there is an intelligent early warning vulnerability in the IoT environment, there must be at least one path from external untrusted input to dangerous operation. In the intelligent early warning vulnerability detection in the IoT environment, the taint data input by the client enters the intelligent early warning vulnerability object method and attribute of the dynamic update page through data transmission, processing, filtering and purification operations, and the execution succeeds or triggers an abnormality. Indicates vulnerability exists.

Therefore, intelligent early warning vulnerability detection in the IoT environment is closely related to the stain data in the application propagation path (Figure 3). Dynamic blot propagation models typically include stain introduction, stain propagation, and stain inspection. The client test script introduces smudges, variable assignment, processing operations, and filter was stain propagation processes. Stain check is to check the execution and exceptions of the test script.

As shown in Figure 3, the blot data in the IoT environment is distributed, transferred, character encoded, and filtered in the program, and then the function is input as a parameter. The horizontal and vertical axis distributions represent blot data and intelligent communication alert vulnerability detection. As shown in the figure, the detection finds that the plot data is marked during the internal propagation of the program. The curve for monitoring and recording the bullet data has a curve number for the discovery of the intelligent early warning vulnerability, so it has little significance for the detection of the intelligent early warning vulnerability.

2) ALGORITHM FLOW

One of the conditions for an intelligent early warning vulnerability attack in the IoT environment is that the page is in an unsafe way to obtain data from the object (or any other object that the attacker can modify), and the object

that the attacker can inject the taint data is the input point. Correspondingly, the DOM object method and attribute of the taint data entering the dynamic update page is another condition of the DOM XSS vulnerability attack. The DOM object method and attribute of the dynamic update page are called output points, including directly modifying the DOM by writing the original HTML function. The function directly executes the script function. If the path from the input point to the output point exists and the taint data is not filtered, the taint data can be executed as an instruction, which proves that the intelligent early warning vulnerability exists. Based on the above analysis, this paper proposes an intelligent communication early warning vulnerability detection algorithm based on dynamic stain propagation model in the IoT.

Algorithm: Intelligent Early Warning Vulnerability Detection Algorithm Based on Dynamic Stain Propagation Model in IoT Environment

Input: Website

Output: A website page with a smart alert vulnerability and a successfully executed vulnerability test script.

a) Crawl the target site's page with input and output points.

b) Injecting taint data into pages containing input and output points.

c) Monitor the output point.

d) If the taint data is executed or an exception occurs, report the intelligent early warning vulnerability, output the website page with the intelligent early warning vulnerability and the script successfully executed, and perform step f); otherwise, perform step e).

e) Judge whether all the taint data has been tested, if yes, perform step f; otherwise, select the next smear data generated by fuzzing, and perform step b).

f) Determine whether all the pages containing the input point and the output point of the target website are tested. If yes, the algorithm ends; otherwise, find the next undetected page and perform step b).

In order to improve the detection efficiency of the algorithm, the stage of obtaining the input point and the output point by the hybrid drive detection adopts the protocol-driven mode. Since the intelligent early warning is a connectionless and stateless object-oriented protocol, only the static content of the webpage is obtained, which is simple and flexible. Features are therefore more efficient. After injecting the stain data into the page, the monitor output point necessarily requires the system to call the parsing engine to listen to the special function in the test script, and the dynamic execution of the page necessarily leads to a decrease in efficiency. Analysis shows that this hybrid drive detection method is more efficient than using a single event-driven detection.

C. STATIC DETECTION OF BUFFER WARNING VULNERABILITIES IN THE ANTI-EXAMPLE OF THE IoT

The early warning vulnerability detection method based on the counter-example of the IoT firstly detects the possible early warning vulnerabilities and their call stacks through the flow sensitive context sensitive detection, and then performs the path sensitive context sensitive detection under the guidance of the rapid detection results, eliminating the fast Detect false alarms and point the user to a counterexample that can cause a buffer warning. Since the accurate detection is performed under the guidance of the rapid detection result, the algorithm can simultaneously obtain high detection accuracy and detection efficiency. In order to overcome the limitations of the static detection method for early warning vulnerabilities using data stream analysis or constraint analysis, the vulnerability detection algorithm based on the counterexample combines the results of data stream analysis with the results of constraint analysis to jointly complete the early warning vulnerability detection.

TABLE 3. A fast vulnerability detection algorithm.

```
procedure argDep(State  S0，  Node n, Node cs)
begin
1.f=funOf(n)
2.fa=inputArgOf(S0, f)
3.nvs=actualArg(fa, cs)
4.S=<Y(x),Y(Y)>
5.return(S , nvs)
end
```

If the input parameters of the current process are referenced in the constraint state, the result of the check may be "undefined". To this end, it is necessary to continuously establish the constraint relationship between the actual parameters of the calling point and the called function parameter through the bottom-to-process inter-process query process, thereby gradually eliminating the constraint information due to the lack of process parameters, resulting in the detection result. Determine the situation.

The above Table 3 is a fast vulnerability detection algorithm, which uses context sensitive stream sensitivity analysis to detect possible early warning vulnerabilities in the program, where the input parameters are the warning expression, the constraint variable set V to be queried, and the queried instruction n, and the initial value. Empty, contains a set of constraint variables, the initial value of n is the detected memory access instruction itself. The complexity of the algorithm is O((ND + E)L), where N and E are the number of nodes of the process control flow graph, D is the program definition number, and L is the number of process call points in the program.

An accurate detection method, based on the buffer warning vulnerability discovered by the rapid detection and the sequence of call points causing the vulnerability, is analyzed from the bottom up and the process by process. For each process being analyzed, the algorithm performs a path-sensitive analysis at the join point to determine a possible counterexample. Eventually, it creates a counterexample that causes a buffer warning for the detected statement, or the alarm that reports the fast detection algorithm is a false positive. The exact detection algorithms within and between processes are given below.

TABLE 4. Host vulnerability information.

Vulnerability exists host	Vulnerability CVE number	Number of vulnerabilities
H1	CVE-2006-3747 CVE-2007-5079	2
H2	CVE-2005-2558 CVE-2007-0908	2
H3	CVE-2008-1087 CVE-2008-1084 CVE-2008-0078 CVE-2008-0604	4
H4	CVE-2009-0087 CVE-2008-0076 CVE-2009-0088 CVE-2009-0658 CVE-2008-4256	5

The precise detection algorithm in the process combines the path-sensitive constraint analysis and the flow-sensitive data flow cycle analysis result, traverses all the negative examples from the process entry point to the suspicious memory access instruction, and establishes the constraint state set of each counter example in the detected instruction. Since there may be many constraint variables in the process that are not related to the security check of the detected instruction, we first determine the constraint variables related to the vulnerability detection and consider the definition of these constraint variables in the counterexample generation process. This method not only reduces the size of the final state, but also allows us to combine multiple identical input states of an instruction to avoid tracking the counterexamples of multiple identical results, thereby greatly reducing the number of paths that need to be tracked and improving the efficiency of detection.

IV. EXPERIMENTS AND RESULTS

A. DATABASE DESCRIPTION

Through the vulnerability information statistics in the experimental environment, there are two vulnerabilities on the H1 Web server in the IoT environment. The vulnerability is mainly the Apace software vulnerability. There are two vulnerabilities on the H2 MySql database server, mainly for database system vulnerabilities, which will affect the storage data. Security; H3 FTP server host system is Windows system, there are a large number of vulnerabilities, some vulnerabilities are vulnerabilities of FTP service software itself; H4 user host system is Windows XP, as the mainstream user operating system has a large number of vulnerabilities, it is likely to be Attackers use as a springboard to attack other servers.

Table 4 shows the specific vulnerability information on the host: The attacker can access the web server and the FTP server in the internal network in a normal way. The attack behavior of the attacker cannot be detected by the firewall. The attacker's attack initial point is its own host H. The attacker uses the H-based platform to exploit the vulnerabilities on the remote Web server and FTP server for penetration and privilege escalation, and then use the server as a springboard to continue to penetrate the user host or the intranet database server.

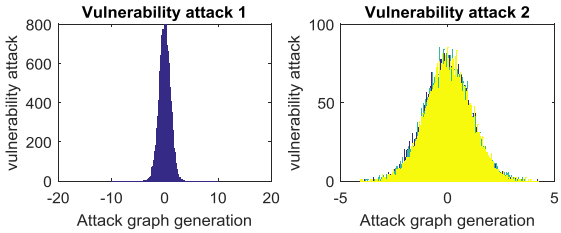


FIGURE 4. Schematic diagram of vulnerability attacks under the IoT.

As shown in Figure 4, in the attack graph generation process, the combination of forward search and backward search is used to generate an attack graph in the IoT environment. The vulnerability attack will have a peak with the attack graph generation process. The redundant nodes in the figure will have a process of reproduction and evolution under the action of the intelligent early warning algorithm as the vulnerability changes.

Assume that the host H2 is MySql database server and stores important data of the intranet. The attacker often has the ultimate goal of such a server, so the importance is the highest, and the importance value is 10. The security evaluation information of each state node is shown in Table 5.

The attacker can access the web server and the FTP server in the internal network in a normal way. The attack behavior of the attacker cannot be detected by the firewall. The attacker's attack initial point is its own host HO. The attacker uses the HO-based platform to exploit the vulnerabilities on the remote Web server and FTP server for penetration and privilege escalation, and then use the server as a springboard to continue to penetrate the user host or the intranet database server. The network attack graph is generated by using the vulnerability detection attack graph algorithm mentioned above in the Internet of Things, and the generated attack graph is shown in Figure. 5.

Take the node weight as the success probability of the node attack, and use the formula of the attack probability of the intelligent early warning vulnerability attack sequence in the IoT environment to calculate the successful attack probability of each node in the critical attack path. The calculation result is shown in Table 6.

In the Table 6, Attack sequence are S0 to Si, Attack probability is Pi, Sequence loss expectation is Pi * C.Based

TABLE 5. Evaluation information of each node of the critical attack path.

Node number	State node	Vulnerability score (S)	Host importance (C)	Loss expectation (S*C)
S1	user(CVE-2006-3747, H1)	8.0	S(medium)	40
S2	root(CVE-2007-5079, H1)	6.5	5(medium)	32.5
S3	user(CVE-2005-2558, H2)	5.2	10(higher)	52
S4	root(CVE-2007-0908, H2)	5.5	10(higher)	55

TABLE 6. Attack sequence loss expectation.

Attack sequence (S0-Si)	Attack probability (Pi)	Si loss expectation C	Sequence loss expectation (Pi*C)
S0-S1	0.237	40	9.48
S0-S2	0.056	32.5	1.82
S0-S3	0.028	52	1.456
S0-S4	0.003	55	0.165

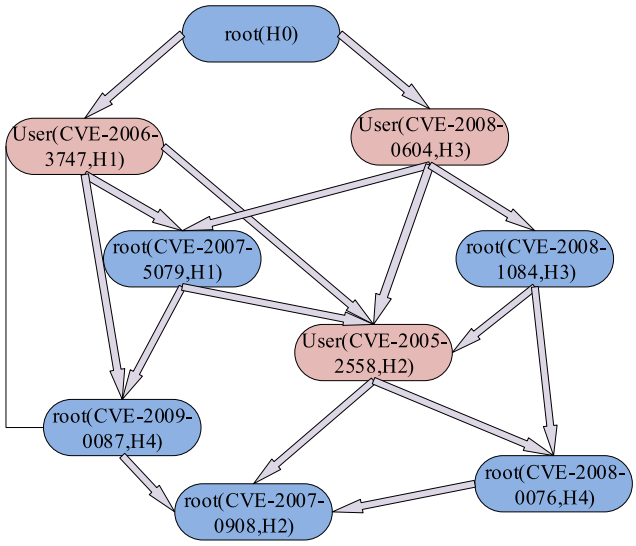


FIGURE 5. Generates an attack graph.

on the data in the Table 6, the total loss expectation of the critical attack path is calculated. The calculation results are as follows:

$$L = \sum_{i=0}^{n} (L_i, P_{0,J}) = 12.92 \tag{9}$$

The loss expectation of the key attack path of the above attack graph, that is, the critical attack path risk value is 12.92, the risk value is the metric value of the experimental network, and the value is used to reflect the overall security status of the current network, and the quantified security metric value enables different network security. The situation is referential and provides guidance to network security managers.

Dynamic intelligent early warning vulnerability mining detection in the IoT environment uses C# to implement protocol-driven crawling, and uses regular expressions to match input points and output points to obtain test IoT. Calling C#-enabled open source tools when implementing event-driven crawlers, by manipulating browsers and DOM objects in the page, it simulates user operations on web pages, triggers various events, and analyzes DOM tree structure changes. The function hijacking of the verification module needs to insert the hijacking function code into the response data packet. In the experiment, the writing filter is used to filter and replace the data packet, and the core of the filter file is written.

In order to verify the effectiveness of the algorithm, this paper builds a local experiment website of intelligent communication early warning vulnerability mining detection algorithm in the IoT environment, collects the vulnerability released by the vulnerability publishing platform, and builds a similar environment for the recurrence of the vulnerability. In order to ensure the comprehensiveness of the algorithm verification, the intelligent communication early warning vulnerability mining detection under the constructed IoT environment is mainly divided into two categories: a) no filtering and purification of user input, mainly for the omission of code writers, forgetting Vulnerabilities caused by filtering user input; b) Vulnerabilities in filtering and encoding, mainly for code writers who have insufficient knowledge of XSS, and loopholes caused by logic problems in encoding filtering. Finally, the intelligent communication early warning vulnerability mining detection algorithm is used for detection. The main content of the detection is the number of detected vulnerabilities, the accuracy of detection and the recall rate, and the total time spent by the detection, which respectively reflect the accuracy and coverage of the detection method. the test results are shown in Table 7.

From the experimental results in Table 7, it can be seen that when detecting unfiltered vulnerabilities, the performance is slightly better, because the hybrid driver crawler can automatically crawl the page, which is beneficial to obtain more input and output points, and has obvious advantages in finding input points ignored by programmers. When detecting the filtered vulnerability, the detection effect is better. The preliminary analysis is more effective because the selected test cases

TABLE 7. Test results comparison.

Test content	Intelligent warning	Vulnerability mining detection
The number of unfiltered vulnerabilities detected/number	37	41
Detect accuracy of unfiltered vulnerabilities /%	39.2	87.8
Recall rate of detected unfiltered vulnerabilities /%	66	72
The number of filtered vulnerabilities detected /	32	29
Detect the accuracy of filtering vulnerabilities /%	71.9	65.5
Detecting the recovery rate of filtering vulnerabilities /%	46	18
Total time spent testing /mm	34.2	17.8

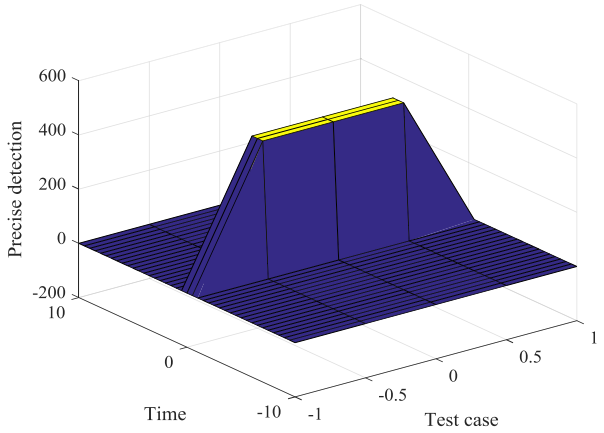


FIGURE 6. Turn on the time distribution map for accurate detection.

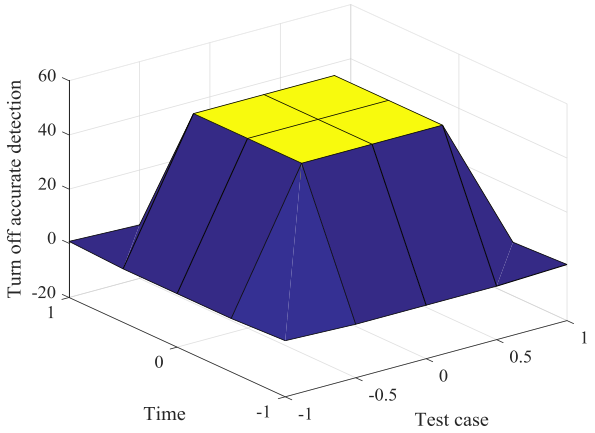


FIGURE 7. Closing the time distribution of accurate detection.

are more effective. Because the filtering and purification methods of the dynamic model application which is of intelligent communication early warning vulnerability detection are different, this diversity leads to detection. The difficulty is increased, so the recall rate of the two tools is not particularly high; the total time spent testing includes the time for tool detection and manual verification to detect the accuracy of the vulnerability, the detection efficiency is greatly improved, and the automation is solved to some extent.

B. EARLY WARNING VULNERABILITY DETECTION VERIFICATION OF IoT COUNTEREXAMPLE

Take the benchmark version of detection min version and. The k version is used as a test case set to test the detection rate and false positive rate of BVC respectively. The test results of the large version and the med version are similar to the min version. In order to test the impact of hierarchical analysis on BVC test results, we performed two tests, one for fast detection, one for accurate detection, and the other for rapid detection and accurate detection.

Figure 6 is a distribution of the time taken by the BVC for fast detection and accurate detection. Figure 7 is a distribution diagram of the time taken by the BVC when only the fast detection is performed.

In order to test the effect of different cyclic analysis methods on the detection effect of BVC, we implemented two loop analyzers, one is a loop analysis method based on reverse path, and the other is a traditional loop analyzer directly using widening/narrowing operation. The two loop analyzers are labeled LAR and LAT, respectively. Since the intermediate code converted by the front end of the LLUM compiler is in the form of SSA, there is no inverse assignment statement, so the method based on the inverse assignment does not improve the analysis precision of the traditional method.

TABLE 8. Static warning results of early warning vulnerabilities using different constraint state security check methods.

Static detector	Detection rate	False alarm rate	Confusion rate	Analysis time (ms)
BVC(CSC)	99%	0%	0%	67133
BVC(CSSC)	99%	14.5%	23.8%	57002

We tested the detection performance of CSC and CSSC separately, the results as shown in Table 8.

Table 8 is the result of their inspection. BVC does not have any false positives when using CSC, and BVC has 14.5% false positives when using CSSC, which indicates that the constraint state security detection with improved range constraint and control constraint can effectively utilize the control constraints to accurately calculate the constraint state of this paper.

First, we tested the analytical accuracy of LAR and LAT separately. Figure 8 shows the number of test cases for the

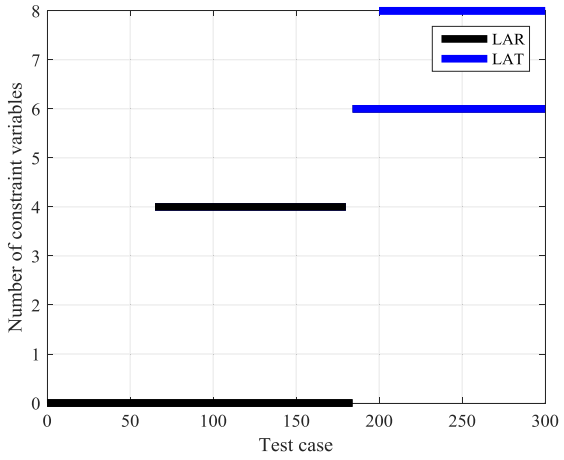


FIGURE 8. LAR results are more precise than the LAT results.

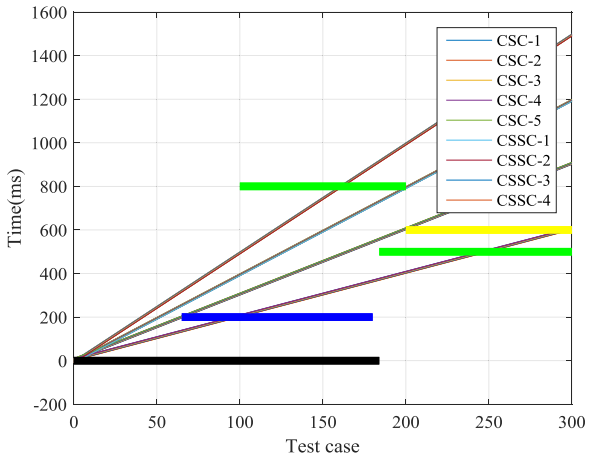


FIGURE 9. Smart alert vulnerability map using CSC.

x-axis of the two analyzers, and the number of constraint variables for the more accurate y. Statistics show that the x-axis indicates that the LAR test results define 220 constraint variables in all of the 291 test cases than the LAT test. The range of 184 constraint variables in the LAR results is more than the range in the LAT results. Accurate, that is, LAR improves the analysis accuracy of the constraint variable by more than 50%.

In order to test the detection effect of different constraint generation methods, we implement two constraint state security checkers, one is the constraint state security checker whose range constraint and control constraint are mutually improved, and the other is a simple constraint state security checker, two constraints The state security checkers are denoted as CSC and CSSC, respectively. In the following experiments, the rest of the BVC remained unchanged, except that they were tested using different constraint state safety checkers.

As shown in Figure 9, Smart Alert Vulnerability Verification with 5 different CSC and 4 different CSSC, the static detection method can improve the accuracy and efficiency of buffer warning vulnerability detection. In particular, the static detection method based on the counter-example buffer warning vulnerability can obtain high detection accuracy without user comments. The detection efficiency has reached the goal.

V. CONCLUSION

This paper studies the existing IoT security assessment methods. The traditional network security assessments are mostly the superposition of vulnerability risk quantification, and lack of correlation analysis of vulnerabilities in the whole network. This paper studies the network security assessment method based on the attack graph association analysis of the IoT environment, and analyzes the attack graph generation algorithm. The weight calculation model is introduced and the method of using the node weight to find the key attack path is proposed. Finally, the method is proposed. The critical attack path is used to measure the security status of the IoT, and a quantitative evaluation plan is given for the security status of the IoT. This paper analyzes the formation principle of intelligent communication early warning vulnerability mining detection under the IoT, and proposes a dynamic pollution propagation model based on the dynamic pollution propagation model for the intelligent communication early warning vulnerability exploitation under the IoT. Static detection method for early warning vulnerabilities based on the counterexample of IoT. As a hierarchical analysis method, this method detects possible buffer early warning vulnerabilities through stream-sensing and context-sensitive rapid detection, and then performs path-sensitive and context-sensitive accurate detection under the guidance of rapid detection results, eliminating rapid detection. The introduced false positives establish a specific path that can lead to buffer warnings. The intelligent communication early warning vulnerability mining detection algorithm finds the input point and output point page stage to use the protocol to drive the crawler. The script injection stage uses the event-driven crawler to compare the experiment with the existing detection tools in the experimental environment, and the experimentally proves the proposed algorithm. It can effectively detect the intelligent communication early warning vulnerability mining detection under the IoT. In the next step, the proposed scheme is applied to the experimental network environment. The topology design and experimental data analysis of the experimental network are given, and the security metric calculation is performed on the experimental network.

REFERENCES

[1] C. Shan, G. P. Jing, C.-Z. Hu, J.-F. Xue, and J.-Z. He, "8031 microcontroller software vulnerability detection algorithm based on vulnerability knowledge database," *Trans. Beijing Inst. Technol.*, vol. 37, no. 4, pp. 371–375, Apr. 2017.

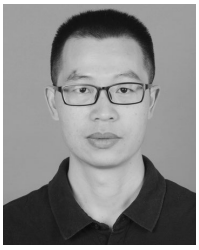
[2] X. Feng, J. Zhang, J. Chen, G. Wang, L. Zhang, and R. Li, "Design of intelligent bus positioning based on Internet of Things for smart campus," *IEEE Access*, vol. 6, pp. 60005–60015, 2018.

[3] X. Jiang and M. Diao, "A new type double-threshold signal detection algorithm for satellite communication systems based on stochastic resonance technology," *Wireless Netw.*, vol. 10, pp. 134–145, May 2019.

[4] S. Wen, Q. Meng, C. Feng, and C. Tang, "Protocol vulnerability detection based on network traffic analysis and binary reverse engineering," *PLoS ONE*, vol. 12, no. 10, Oct. 2017, Art. no. e0186188.

[5] A. V. Barabanov, A. S. Markov, and V. L. Tsirlov, "Statistics of software vulnerability detection in certification testing," *J. Phys., Conf. Ser.*, vol. 1015, May 2018, Art. no. 042033.
[6] X. Li, J. Chen, Z. Lin, L. Zhang, Z. Wang, M. Zhou, and W. Xie, "A vulnerability model construction method based on chemical abstract machine," *Wuhan Univ. J. Natural Sci.*, vol. 23, no. 2, pp. 150–162, 2018.
[7] G. Shi, Y. He, Q. Luo, B. Li, and C. Zhang, "Portable device for acetone detection based on cataluminescence sensor utilizing wireless communication technique," *Sens. Actuators B, Chem.*, vol. 257, pp. 451–459, Mar. 2018.
[8] S. M. Darwish and A. G. El-Shnawy, "An intelligent database proactive cache replacement policy for mobile communication system based on genetic programming," *Int. J. Commun. Syst.*, vol. 31, no. 2, May 2018, Art. no. e3536.
[9] H. Xu, "Intelligent modulation algorithm of WMSN communication channel," *J. Discrete Math. Sci. Cryptogr.*, vol. 20, nos. 6–7, pp. 1477–1481, 2017.
[10] A. Rahman, D. Musleh, N. Aldhafferi, A. Alqahtani, and H. Alfifi, "Adaptive communication for capacity enhancement: A hybrid intelligent approach," *J. Comput. Theor. Nanosci.*, vol. 15, no. 4, pp. 1182–1191, 2018.
[11] J. Wang, W. Gang, R. Bai, B. Li, and Y. Zhou, "Ground simulation method for arbitrary distance optical transmission of a free-space laser communication system based on an optical fiber nanoprobe," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 9, no. 12, pp. 1136–1144, Dec. 2017.
[12] L. Yang and H. Li, "Vehicle-to-vehicle communication based on a peer-to-peer network with graph theory and consensus algorithm," *IET Intell. Transp. Syst.*, vol. 13, no. 2, pp. 280–285, Feb. 2019.
[13] G. Yang, Q. Zhang, and Y.-C. Liang, "Cooperative ambient backscatter communications for green Internet-of-Things," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1116–1130, Apr. 2018.
[14] J. Haitao, Y. Guo, H. Chen, J. Guo, C. Zhou, and J. Xu, "Unauthorized access vulnerability detection method based on finite state machines for mobile applications," *J. Nanjing Univ. Sci. Technol.*, vol. 41, no. 4, pp. 434–441, 2017.
[15] Q.-Y. Zhang, S.-B. Qiao, Y.-B. Huang, and T. Zhang, "A high-performance speech perceptual hashing authentication algorithm based on discrete wavelet transform and measurement matrix," *Multimedia Tools Appl.*, vol. 77, no. 16, pp. 21653–21669, 2018.
[16] J. J. Anaya, A. Ponz, F. García, and E. Talavera, "Motorcycle detection for ADAS through camera and V2V Communication, a comparative analysis of two modern technologies," *Expert Syst. Appl.*, vol. 77, pp. 148–159, Jul. 2017.
[17] A. Pinto, W. R. Schwartz, H. Pedrini, and A. Rocha, "Using visual rhythms for detecting video-based facial spoof attacks," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 5, pp. 1025–1038, May 2015.
[18] C. Vellaithurai, A. Srivastava, S. Zonouz, and R. Berthier, "CPIndex: Cyber-physical vulnerability assessment for power-grid infrastructures," *IEEE Trans. Smart Grid*, vol. 6, no. 2, pp. 566–575, Mar. 2015.
[19] Y. He, G. J. Mendis, and J. Wei, "Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism," *IEEE Trans. Smart Grid.*, vol. 8, no. 5, pp. 2505–2516, Sep. 2017.
[20] A. Sepas-Moghaddam, F. Pereira, and P. L. Correia, "Light field-based face presentation attack detection: Reviewing, benchmarking and one step further," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 7, pp. 1696–1709, Jul. 2018.
[21] S. Bhilare, V. Kanhangad, and N. Chaudhari, "A study on vulnerability and presentation attack detection in palmprint verification system," *Pattern Anal. Appl.*, vol. 21, no. 3, pp. 769–782, 2017.
[22] T. B. Patel and H. A. Patil, "Cochlear filter and instantaneous frequency based features for spoofed speech detection," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 4, pp. 618–631, Jun. 2017.
[23] K. Satpathi, Y. M. Yeap, A. Ukil, and N. Geddada, "Short-time Fourier transform based transient analysis of VSC interfaced point-to-point DC system," *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 4080–4091, May 2018.
[24] X. Chen, L. Wang, T. Wang, Y. Liu, and H. Yang, "A general framework for hardware Trojan detection in digital circuits by statistical learning algorithms," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 36, no. 10, pp. 1633–1646, Oct. 2017.
[25] A. Anwar, A. N. Mahmood, and Z. Tari, "Ensuring data integrity of OPF module and energy database by detecting changes in power flow patterns in smart grids," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3299–3311, Dec. 2017.
[26] Y. Li, S. Hu, and A. Y. Zomaya, "The hierarchical smart home cyberattack detection considering power overloading and frequency disturbance," *IEEE Trans. Ind. Informat.*, vol. 12, no. 5, pp. 1973–1983, Oct. 2017.
[27] S. Peng, P. Liu, and J. Han, "A Python security analysis framework in integrity verification and vulnerability detection," *Wuhan Univ. J. Natural Sci.*, vol. 24, no. 2, pp. 141–148, Apr. 2019.
[28] W. Xu and J. Leng, "Simulation of potential vulnerability spillover detection under network active protection," *Comput. Simul.*, vol. 183, no. 993, pp. 190–202, Mar. 2018.
[29] M. Yasinzadeh and M. Akhbari, "Detection of PMU spoofing in power grid based on phasor measurement analysis," *IET Gener. Transmiss. Distrib.*, vol. 12, no. 9, pp. 1980–1987, May 2018.
[30] M. Kumar and A. Sharma, "An integrated framework for software vulnerability detection, analysis and mitigation: An autonomic system," *Sādhanā*, vol. 42, no. 9, pp. 1481–1493, 2017.
[31] M.-C. Chen, S.-Q. Lu, and Q.-L. Liu, "Global regularity for a 2D model of electro-kinetic fluid in a bounded domain," *Acta Mathematicae Applicatae Sinica-English*, vol. 34, no. 2, pp. 398–403, 2018.
[32] W. Wei, J. Su, H. Song, H. Wang, and X. Fan, "CDMA-based anti-collision algorithm for EPC global C1 Gen2 systems," *Telecommun. Syst.*, vol. 67, no. 1, pp. 63–71, 2017.
[33] Y. Sun, H. Qiang, J. Xu, and G. Lin, "IoT-based online condition monitor and improved adaptive fuzzy control for a medium-low-speed maglev train system," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/TII.2019.2938145.
[34] W. Wei, X. Xu, W. Marcin, F. Xunli, D. Robertas, and L. Ye, "Multi-sink distributed power control algorithm for cyber-physical-systems in coal mine tunnels," *Comput. Netw.*, vol. 161, pp. 210–219, Oct. 2019.

MAO YI was born in Chongqing, China, in 1982. He received the M.S. degree in control theory and control engineering from Chongqing University, Chongqing, in 2007. He is currently a Vice Professor of IoT technology with the School of Electronics and IoT, Chongqing College of Electronic Engineering, Chongqing. His research interests include control theory and control engineering, the IoT application technology, and information engineering.



XIAOHUI XU was born in Sanming, Fujian, China, in 1979. He received the M.S. degree in computer software and theory from Chongqing University, Chongqing, China, in 2005, and the Ph.D. degree in computer science and technology from Shanghai Jiao Tong University, Shanghai, China, in 2011. He is currently a Vice Professor and a Senior Engineer of IoT technology with the School of Electronics and IoT, Chongqing College of Electronic Engineering, Chongqing. His research interests include urban computing, service oriented computing, and information fusion.

LEI XU was born in Chongqing, China, in 1981. He received the B.Sc. degree in automation and the Ph.D. degree in control theory and control engineering from Chongqing University, Chongqing, in 2004 and 2009, respectively. He is currently a Professor of sensor network technology with the Chongqing College of Electronic Engineering, Chongqing. His research interests include intelligent information, and control and advanced perception technology.