IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 26, 2019, accepted September 6, 2019, date of publication September 13, 2019,
date of current version September 25, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2941006

A Prefix Hijacking Detection Model Based on the Immune Network Theory

JIAN ZHANG[1,2], DAOFENG LI[1,2], AND BOWEN ZHAO[3]
[1]School of Computer, Electrical and Information, Guangxi University, Nanning 530004, China
[2]Guangxi Colleges and Universities Key Laboratory of Multimedia Communications and Information Processing, Guangxi University, Nanning 530004, China
[3]School of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, China

Corresponding author: Daofeng Li (ldf_0123@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61662004, and in part by the National Natural Science Foundation of Guangxi under Grant 2016GXNSFAA380215.

ABSTRACT The prefix hijacking problem is an urgent security issue that need to address in the Border Gateway Protocol (BGP) security research. In order to solve the problem of prefix hijacking in BGP, we propose (a) new (p)refix (h)ijacking (d)etection model based on the immune network theory in this paper, called aPHD. To be specific, aPHD uses real BGP UPDATE messages for pre-training and has the ability to detect UPDATE messages in real time after pre-training. The aPHD (1) can effectively detect prefix hijacking attacks with high accuracy; (2)is easy to deployment; and (3) has a low false positive rate and low overhead. Extensive performance evaluation shows that our solution is secure and feasible. The aPHD improved the accuracy rate by 6.2% and reduced the false positive rate by 85.7%.

INDEX TERMS Immune network theory, prefix hijacking, BGP security, negative selection.

I. INTRODUCTION

Due to the large scale of the Internet and a large number of ISPs (Intemet Service Provider), attacks against BGP are increasing and seriously affect the use and development of the Internet [1]. Because BGP lacks a secure and reliable route authentication mechanism, the authenticity and integrity of the routing information cannot be verified. Moreover, BGP unconditionally trusts interconnected autonomous system (AS), which results in BGP being vulnerable to abnormal route advertisement attacks [2]. Since BGP is currently the only interdomain routing protocol in use [3], its security is of great significance for the reliable and stable operation of the entire Internet.

BGP routing messages include Network Layer Reachability Information (NLRI) [4] and path attributes. NLRI consists of the IP prefix and length. The IP prefix is one of the most critical information in the BGP routing information. Its primary function is to identify the network address of the reachable AS announced by the BGP routing message. At present, there are many attacks against BGP exploiting NLRI and path attributes, such as prefix hijacking [5], path forgery [6], route leak [7], and TCP protocol attacks [8]. The prefix hijacking means that a malicious AS controls the IP address block without the consent of the legal owner of the prefix. Prefix hijacking can result in the man-in-the-middle attack and even routing black holes. Prefix hijacking causes global routing fluctuations and consumes many network resources. Prefix hijacking can lead to the inability to track the source of spam [9], which also blocks the way to address spam fundamentally. Prefix hijacking often occurs on the Internet. For example, in 2018, Amazon lost control of 1,300+ Amazon Cloud Services IP address for two hours, when hackers used a BGP-hijacking to reroute traffic to rogue destinations.[1] Therefore, prefix hijacking is a crucial issue that need to address in BGP security research.

Considering that traditional intrusion detection methods have the disadvantages of high false positive rate [10] and inability to effectively identify new types of attacks, traditional intrusion detection methods cannot meet the requirements of prefix hijacking detection. In this paper, we design a prefix hijacking detection model based on immune network theory, called aPHD, which focuses on three issues: (1) event collection, where excellent detection efficiency is supported by data; (2) real-time detection, where the received UPDATE messages will be detected; (3) attack response, where the detected attack gets a quick response. In aPHD, we apply the immune network theory to construct a self-organizing,

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Maaz Rehan.

[1] https://www.cbronline.com/news/amazon-cloud-ip/amp/

self-learning, adaptive, and robust detection model that can be used to replace the traditional intrusion detection model. In order to achieve effective prefix hijacking detection, we construct a variety of detector objects for detection in different situations. Then we use detectors to identify prefix hijacking events, just like antibodies in organisms recognize antigens. By combining immune network theory with prefix hijacking detection, identify attacks and fast responses can be effectively achieved. Moreover, we demonstrate that aPHD is secure and feasible through performance evaluation. The main contributions of this paper can be summarized as follows:

- This paper systematically analyzes the commonalities between immune network theory and prefix hijacking detection and also defines a set of dynamic evolution equations.
- Compared with the related schemes that modify the content of BGP protocols, aPHD is easy to deploy and has low overhead.
- Compared with the related schemes using anomaly detection, aPHD has a high detection rate and a low false positive rate.

The rest of the paper is organized as follows. Section II presents the related work and Section III provides the preliminaries. In Section IV, we formulate the immune model and the immune network theory. In Section V, we describe the detailed construction of aPHD. In Section VI, we implement the aPHD and make a detailed performance analysis in comparison with the related work theoretically and experimentally. Finally, we conclude the paper in Section VII.

II. RELATED WORK

There are two topics related to our research: route authentication technology and prefix hijacking detection technology. They are discussed separately below.

A. ROUTE AUTHENTICATION TECHNOLOGY

The SIDR (Secure Inter-Domain Routing) working group of IETF (Internet Engineering Task Force) decomposes BGP route authentication into two problems:

- Whether an AS has a legal authorization to advertise an IP prefix.
- Whether the AS_PATH in a BGP route is consistent with the path propagated by its NLRI.

These two problems represent the authenticity and integrity of the routing information. Solving these two problems is equivalent to basically eliminating the security threat of prefix hijacking. Around the solution of these two problems, quite a lot of scientific ideas have emerged. Firstly, the most natural idea is to introduce a PKI (Public Key Infrastructure) to sign BGP routing messages digitally, such as secure BGP [11], secure origin BGP [12], interdomain routing validation [6], resource public key infrastructure and BGPsec [13], route origin verification [14]. Digital signatures have proven to be the most effective way to solve identity authentication problems after years of development. In addition, some studies mainly focus on the following aspects:

- Safe and efficient origin AS certification [15].
- Safe and efficient AS_PATH certification [16], [17].
- Simple and easy to deploy PKI system [18].
- Lightweight authentication technology [19].

In fact, route authentication technology can fundamentally solve the prefix hijacking problem. However, there are two significant shortcomings in this type of research. The defects are the need to establish and deploy a complex PKI system and the corresponding computational overhead caused by the use of asymmetric encryption and decryption techniques. Therefore, routing authentication technology cannot meet current needs due to its two significant drawbacks.

B. PREFIX HIJACKING DETECTION TECHNOLOGY

Prefix hijacking detection technology is another research hotspot in prefix hijacking. There are also many different technical solutions. In fact, most of the solutions are based on the following two essential features of prefix hijacking to study related detection techniques:

- MOAS (Multiple Origin AS) conflict [20]. It means that a prefix matches the behavior of multiple origin ASs. This is the essential feature of prefix hijacking in the routing control plane.
- IP address conflict [21]. It means that prefix hijacking directly leads to the problem that there are multiple different routing destinations for one destination IP address in the routing data plane. In other words, assuming that 128.0.0.0/16 is owned by AS 1 but is hijacked by AS 2, packets with a destination address of 128.0.0.0/16 may be returned from AS 1 and AS 2 respectively. Alternatively, the source address is 128.0.0.0/ 16 packets may have no return.

Based on the above two characteristics, one type of detection technology focuses on how to find MOAS conflict in real time and then determine whether prefix hijacking has occurred. Zhao X *et al*. [22] first proposed the MOAS detection. The core of MOAS detection technology is how to quickly and accurately detect invalid MOAS conflicts. More powerful MOAS detection techniques include MOAS List [23], prefix hijack alert system [24], and pretty good BGP [25]. Another type of detection technology uses active detection to determine whether a prefix hijacking has occurred by sending various probe packets and based on their response [26]. The prefix hijacking detection technique cannot completely solve the security problem. However, it is a lightweight solution when the complete PKI system has not been deployed yet. The prefix hijacking detection technology does not need to modify the existing protocol specification, but its disadvantage is the possibility of false positives and false negatives. This paper lists the comparison results of route authentication technology, prefix hijacking detection technology, and the aPHD on various evaluation indexes, as shown in Table 1.

TABLE 1. Comparison of related work.

Techniques	No Cryptography	Maintain protocol	Low-overhead	Real-time	Lightweight	Easy to deploy
Route authentication [11]	×	×	×	√	×	×
Prefix hijacking detection [28]	√	√	√	×	√	√
aPHD	√	√	√	√	√	√

With the advent of immune network theory, some scholars have applied immune network theory to network security, especially intrusion detection [27]. Different from traditional intrusion detection schemes, intrusion detection schemes based on immune network theory are fully capable of coping with the dynamics and complexity of network security. Farmer *et al.* [29] pioneered the dynamic model of the immune system based on immune network theory, and explored the connection between the immune system and other artificial intelligence methods, and began the research of artificial immune system. Timmis *et al.* [30]–[32] applied immune network theory to the field of pattern recognition and pointed out that the application of immune network theory in the identification of outliers is feasible and effective. Zhou and Dasgupta [33] applied the immune network theory directly to the shape recognition in two-dimensional space, and obtained a reasonable recognition rate and proposed a negative selection algorithm based on immune network theory for intrusion detection [34]. Secker *et al.* [35] applied immune network theory to the detection of spam. Ti [36] proposed a computer virus detection based on immune network theory and cosine similarity and gave the quantitative description of the model. Jamali and Fotohi [37] proposed to combat wormhole attacks by applying immune network theory. Blum *et al.* [38] proposed a network intrusion detection based on immune network theory. In these efforts, the immune network theory has adequately verified the detection of abnormal data. Compared with other anomaly detection methods, immune network theory has the advantage of low continuity of data, no need to provide abnormal signals as prior knowledge and only need standard signals as prior knowledge for training [39].

Because immune network theory is useful in the field of engineering, this paper applies it to prefix hijacking detection. This article intends to meet the evaluation indicators shown in Table 1. The above existing solutions cannot adequately meet these advantages.

III. PRELIMINARIES

In this section, we review some background of prefix hijacking and immune network theory before detailing our construction.

A. PREFIX HIJACKING

Prefix hijacking means that an AS advertises an unauthorized prefix. The so-called unauthorized is that the prefix belongs to other ASs or the address space has not been allocated. Internet address allocation follows the authorization level from IANA (Internet Assigned Numbers Authority) to RIR (Regional Internet Registries) to LIR (Local Internet Registries). The AS violates the authorization to illegally announce the illegal prefix, which will directly cause traffic hijacking. Previous studies have shown that prefix hijacking is mainly caused by administrators misconfiguring BGP routers [40]. The reason is mostly related to IGP (Interior Gateway Protocol) to BGP route redistribution. However, Pakistan Telecom made a malicious initiative to hijack YouTube's prefix in order to restrict its domestic users from accessing the YouTube site in 2008. Since then, the research community has been studying more and more of this malicious prefix hijacking behavior. In general, a malicious attacker can achieve the purpose of successfully implementing prefix hijacking by forging NLRI information and AS_PATH path.

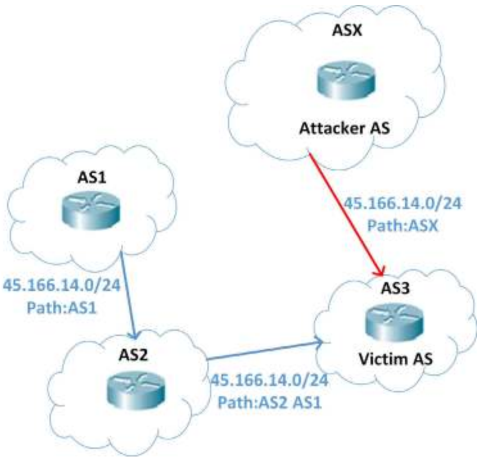


FIGURE 1. Forged NLRI prefix.

1) FORGING NLRI INFORMATION

In this case, the malicious AS falsifies the NLRI information in the BGP UPDATE message and advertises an illegal prefix [41]. As shown in Fig. 1, AS1 is the legal owner of the prefix 45.166.14.0/24, which advertises the route to the prefix. At this point, AS X malicious forgery NLRI also advertises the route to 45.166.14.0/24. The blue line represents the standard route propagation path, and the red line represents the abnormal route propagation path. According to the shortest AS_PATH principle in BGP, AS 3 will preferentially select the path through AS X to 45.166.14.0/24. Furthermore, if the attacker not only falsifies the prefix in the NLRI but also modifies it to a longer prefix length, all other ASs will choose the forged path according to the longest matching principle of BGP. ASX hijacks the AS1 prefix and can spam messages

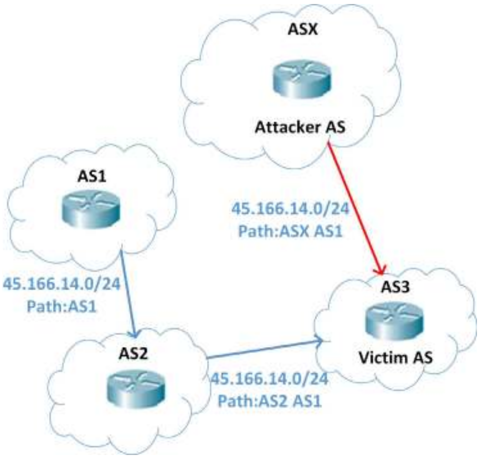


FIGURE 2. Forged AS_PATH path.

and conduct DoS attack, spam, and other attacks without worrying about divulging identity through the source IP.

2) FORGING AS_PATH PATH

Forging NLRI information to implement prefix hijacking attacks can cause MOAS conflicts. MOAS conflicts are easily detected by existing BGP monitoring tools.[2,3] To avoid such monitoring, an attacker can resolve the MOAS problem by modifying both the NLRI information and the AS_PATH path. As shown in Fig. 2, AS 1 is the owner of the prefix 45.166.14.0/24. AS X is not physically connected to AS 1. However, AS X modifies AS_PATH to 5,1 so that other ASs think AS X is connected to AS 1. AS X only needs to ensure that AS 5 is the beginning and AS 1 is the end. Subsequently, AS X will hijack the traffic sent by AS 3 to AS 1 according to the shortest path principle. If no user feedback network is unreachable, the victim will not find the attack behavior. The prefix owner is the only organization that can accurately distinguish between legal changes and prefix hijacking.

B. IMMUNE NETWORK THEORY

The Biological Immune System (BIS) is a multi-level system that is highly distributed, highly parallel, and can handle complex information characteristics. BIS fights pathogens in a variety of intelligent ways, using innate and acquired immunity to accurately and specifically respond to them [42], [43]. By mutating, evolving, and learning to adapt to unfamiliar environments, BIS can respond quickly to known pathogens and unknown pathogens. In recent years, researchers have simulated the biological immune system and derived the Artificial Immune System (AIS) for solving engineering and scientific problems. AIS is widely used in computer security areas such as abnormal diagnosis and virus detection. AIS is a computational system inspired by BIS, which is characterized by dynamics, adaptability, robustness [44] and distribution [45]. Therefore, we use the immune network theory to improve detection efficiency in the prefix hijacking detection model.

The prefix hijacking detection model is very similar to the artificial immune system. First, both are made up of many independent objects that interact in multiple ways. Independent objects in AIS are multiple immune cells, and independent objects in the prefix hijacking detection model are detectors. Second, the goal of both is to ensure that the system is more secure by detecting intrusions. The primary function of the immune system is to identify and suppress malicious antigens based on the principle of ''non-I am the enemy.'' The goal of deploying a prefix hijacking detection model is to detect prefix hijacking events to ensure the security of the inter-domain routing system. Therefore, we propose an immune network theory-based model for prefix hijacked detection (aPHD). The prefix hijacking detection model uses algorithms such as immune memory and negative selection in the immune mechanism to identify normal routing messages and abnormal routing messages.

IV. PROBLEM FORMULATION

A. SYSTEM MODEL

In our aPHD, we consider a detection system based on filtered data streams. It uses real Internet attack data instead of simulated Internet attack data to train the detectors. We deploy the detection system in the data stream. The detection system filters out UPDATE messages that are subject to a prefix hijacking attack.

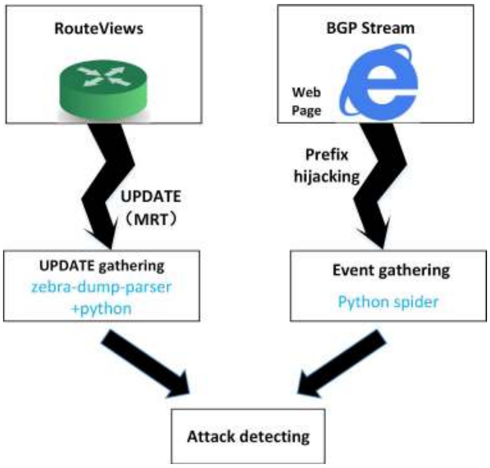


FIGURE 3. System model of our scheme.

As shown in Fig. 3, there are three parts in the system model: UPDATE gathering, event gathering, and attack detecting. The University's Route Views[4] project was originally conceived as a tool for Internet operators to obtain real-time BGP information about the global routing system from the perspectives of several different backbones and locations around the Internet. UPDATE gathering is responsible

[2] https://bgpmon.net/

[3] https://cyclops.cs.ucla.edu/

[4] http://www.routeviews.org/routeviews/

for extracting routing data from one of the global Collectors from Routeviews in a certain period of time. Due to Zebra bRIB and BGP update dumps are in the well-known MRT (Multi-Threaded Routing Toolkit) [46] format. UPDATE gathering also requires zebra-dump-parser to convert the routed data into a readable format. BGP Stream[5] is a free resource for receiving alerts about hijacks, leaks, and outages in the BGP. Event gathering is used to obtain the real prefix hijacking data on the BGP Stream. Attack detection performs system initialization and generates system parameters, and the received two sets of data are used for the training of the detectors. After a period of training, the detector will have the ability to identify prefix hijacking attacks. At this point, we can deploy attack detecting on the BGP router to detect the passed UPDATE message in real time.

B. DESIGN GOALS

According to the above model, the design goals of our aPHD are as follows:

- *Low overhead.* This property requires that aPHD should not interfere with the normal communication of the BGP router while ensuring its normal operation. This means that the amount of computation of aPHD should not exceed the upper limit of the BGP router.
- *High accuracy.* High accuracy proves that aPHD is useful and is required by most detection models. Specifically, aPHD needs to have good ability to detect prefix hijacking attacks. High accuracy rates are often accompanied by low miss rates.
- *Low error rate.* This property is to ensure that the aPHD does not recognize all UPDATE messages as prefix hijacking attacks. In general, traditional intrusion detection schemes are often accompanied by high false positive rates, called error rates. In this paper, we use the immune network theory to reduce the error rate of aPHD. In addition, the low error rate guarantees that the Internet will not be subject to large routing fluctuations. Because once a prefix hijacking is found, aPHD will notify the BGP router to discard the UPDATE message, making the path unreachable.

In addition, considering the security of UPDATE messages, the following properties should be achieved.

- *Confidentiality.* Since the aPHD will be deployed directly on the BGP router, it is equally essential to ensure the security of the routing information. The aPHD should only decode the information needed for the detection process, and should not decode the non-essential information to ensure the confidentiality of the routing information. At the same time, the aPHD should ensure that the data obtained is never leaked.
- *Integrality.* During the detection process, aPHD should ensure that routing information is not accidentally or deliberately deleted, modified, forged, out of order, replayed, or inserted.

V. OUR PROPOSED APHD SCHEME

In this section, we introduce the overview of aPHD, describe the construction in detail. All the notations in the following description can be referred in Table 2.

TABLE 2. Notations used in this paper.

Notations	Description
$\mathbb{G}$	problem domain
$Self$	normal UPDATE message set
$Nonself$	UPDATE message set with the prefix being hijacked
Ag	UPDATE message set
D	antibody detector set
I	immature detector set
M	mature detector set
E	memory detector set
d	binary sequence
p	tolerance period
age	the age of detector
$count$	number of matching Ag
$PERIOD$	threshold of p
AGE	threshold of age
$COUNT$	threshold of $count$
f_n	function called n
t	time point
$S(t)$	set called S at time t
$S_n(t)$	new set S generated at time t
$S_d(t)$	set S of death at time t
$S_s(t)$	static set S at time t
$S_t(t)$	need to tolerate set S at time t
$S_a(t)$	activated set S at time t
$Self\_Max$	the maximum number of $Self$ elements

A. OVERVIEW

In aPHD, We applied the theory of immune network to construct a real and efficient model. Specifically, aPHD acquires BGP data for detector training. Trained detectors will have the ability to detect prefix hijacking attacks. However, the pros and cons of the data directly affect the final performance of the detector. To solve this dilemma, aPHD uses real routing data and attack data. In this way, the detector can get the most out of training. To achieve excellent detection results, we established an immune model belonging to prefix hijacking detection based on immune network theory. Since the real network environment is changing at any time, we will realize the dynamic evolution of the immune model. Furthermore, we limit the number of detectors to a specific range to avoid overloading the time overhead due to a large number of detectors. To maintain the diversity of the detectors, the detectors are generated randomly and require a period of tolerance to work formally. Therefore, aPHD takes a while to initialize before it officially works. Combining immune network theory with prefix hijacking detection can effectively identify prefix hijacking attacks and ensure AS security.

B. EVIDENCE COLLECTION

In this section we mainly introduce the design and deployment of the evidence collection. The evidence collection for aPHD consists of two parts: **UPDATE gathering** and **event gathering**. In our construction, evidence collection

[5] https://bgpstream.com/

is programmed and developed by python language.[6] The detailed evidence collection is described below.

1)***UPDATE gathering:*** UPDATE gathering converts MRT formatted routing data collected from Routerviews into readable ASCII (American Standard Code for Information Interchange) data. The UPDATE gathering extracts the feature information such as the IP prefix, the prefix length, and the route attributes from the routing information to form a binary string as an antigen set Ag. This process is called antigen presentation. UPDATE gathering implements antigen presentation by calling algorithm 1. Algorithm 1 needs to input the BGP data compression package downloaded from Routeviews and output the antigen string in binary format.

```
Algorithm 1 AntigenPresentation
Input: UPDATE.bz2
Output: antigen.txt
update = Decompress(UPDATE.bz2);
data_ascii = Zebra(update);
while Read(data_ascii) do
    info = Abstract(data_ascii);
    bin_str = Bin(info);
    antigen.txt = Write(bin_str);
end while
return antigen.txt;
```

2)***Event gathering:*** BGP Stream will publish detected attack events through Twitter. Event gathering uses python crawlers to get real prefix hijacking attack data. We need to find the URL of the event along with the posted twitter. Event gathering also needs to obtain the IP prefix, prefix length, and route attributes of the prefix hijacking attack and convert it to a binary string. Event gathering implements data collection by calling algorithm 2. Algorithm 2 needs to input the number of prefix hijacking events that need to be obtained and output the event string in binary format.

```
Algorithm 2 HijackingEvent
Input: N
Output: event.txt
url = "twitter.com/bgpstream";
for $i \in [1, N]$ do
    Addr[i] = Open(url);
end for
for $j \in [1, N]$ do
    info = Open(Addr[j]);
    bin_str = Bin(info);
    event.txt = Write(bin_str);
end for
return event.txt;
```

C. IMMUNE MODEL

In this section, we mainly introduce an immune model suitable for prefix hijacking detection. We use the theory of immune network as a template to find the point of convergence between immune network theory and prefix hijacking. Define problem domain $\mathbb{G} = \bigcup_{i=1}^{\infty} \{0, 1\}^i$, antibody set $Ag \subset \mathbb{G}$, self set $Self \subset Ag$ and nonself set $Nonself \subset Ag$. The relationship between $Self$ and $Nonself$ is as follows: $Self \cup Nonself = Ag$ and $Self \cap Nonself = \varnothing$ [47]. Ag represents all UPDATE messages set, where $Self$ represents normal UPDATE messages set, and $Nonself$ represents UPDATE messages set for which the prefix is hijacked. Ag, $Self$, and $Nonself$ are binary strings obtained by antigen presentation. In the antigen presentation process, we take reasonable measures to reduce the UPDATE message attributes to reduce search space and computational complexity. Define the antibody detector set $D = \{< d, p, age, count > | d \in \mathbb{G}, p \in N, age \in N, count \in N\}$, where N represents a natural number. We also use a string of binary strings d to represent the antibody detector. In the immune model, the detector is an important abstract concept. The detectors D are divided into three categories: immature detector I, mature detector M and memory detector E. So there is $I \cup M \cup E = D$ and the three have no intersection at the same time.

1) *SELF* DYNAMIC EVOLUTION

In the real network environment, the number of RIB (Routing Information Base) is so large that $Self$ is too large. Since the cost of self-tolerance of the detector is exponentially related to the size of $Self$, the computational overhead is too large due to the large $Self$. Therefore, we propose a scheme in which $Self$ dynamically changes over time to reduce computational overhead. Define the evolution equation of $Self$

$$Self(t) = Self_s(t) \quad t = 0, \tag{1}$$

$$Self(t) = Self(t-1) + Self_n(t) - Self_d(t) \quad t \geq 1. \tag{2}$$

$Self_s$ indicates a manually configured static route set. $Self_d$ indicates $Self$ that is eliminated according to the LRU (Least Recently Used) principle when the size of $Self$ exceeds the threshold $Self\_Max$.

2) *I* DYNAMIC EVOLUTION

To maintain the diversity of the detector and the correct rate of detector matching attacks, we randomly generate immature detectors. In order to prevent the detector from matching the set of self, the immature detector must undergo self-tolerance. It evolved into a mature detector through a self-tolerant detector and died without a self-tolerant immature detector. Define the evolution equation of I

$$I(t) = I(t-1) + I_n(t) - I_d(t) - I_a(t) \quad t \geq 1, \tag{3}$$

$$I_a = \{d | d \in I(t-1) \wedge d.p > PERIOD\}. \tag{4}$$

I_n denotes a randomly generated new detector set, I_a denotes a set of mature detectors evolved by immature detectors, and I_d denotes an immature detector set that is screened out during self-tolerance. When $t = 0$, there is $I = \varnothing$. Immature detectors can only evolve into mature detectors if they are not matched to $Self$ throughout the tolerance period.

[6] https://www.python.org/

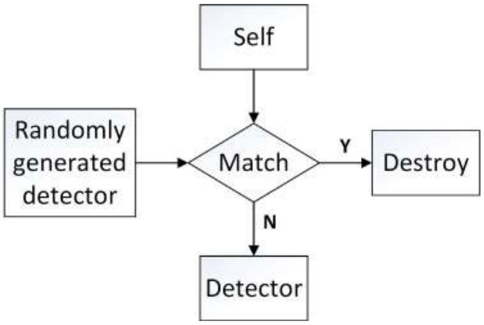


FIGURE 4. Negative selection algorithm.

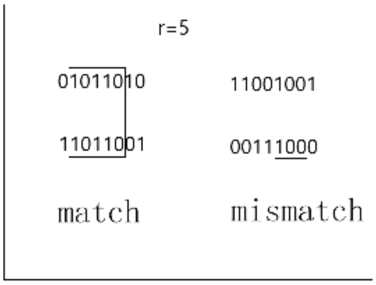


FIGURE 5. R-continual position match algorithm.

We use the NSA (negative selection algorithm) [48], [49] in the artificial immune algorithm as the self-tolerance algorithm. The randomly generated detector performs matching detection with *Self*. If the match is successful, the detector is destroyed, and if the match fails, the detector is retained, as shown in Fig. 4.

Define the self-tolerance function as follows

$$f_t(I) = I - \{d | d \in I \land \exists x \in Self \land f_m(d, x) = 1\}. \quad (5)$$

We use r-continual position match algorithm as the matching function f_m in f_t. $f_m = 1$ stands for match and $f_m = 0$ stands for mismatch. As shown in Fig. 5, if the number of identical consecutive characters in the corresponding position of the two strings is greater than or equal to r, then the two strings are considered to match, where r is the matching threshold.

3) *M* DYNAMIC EVOLUTION

Define the evolution equation of M

$$M(t) = M(t-1) + M_n(t) - M_d(t) - M_a(t) t \geq 1, \quad (6)$$

$$M_d = \{d | d \in M(t-1) \land d.age > AGE\}, \quad (7)$$

$$M_a = \{d | d \in M(t-1) \land d.count \geq COUNT\}. \quad (8)$$

M_a represents a set of mature detectors that are activated as memory detectors. The condition is that the number of mature detector matching prefix hijacking attacks is greater than or equal to the matching threshold $COUNT$. M_d represents a set of mature detectors that died due to a life cycle exceeding the threshold AGE. When $t = 0$, there is $M = \varnothing$. The death mechanism ensures the practicality of the detector. The death mechanism eliminates detectors that have not worked for a long time and retains the dominant detector. The mature detector is used to detect prefix hijacking attacks of unknown features, just like specific immunity in BIS.

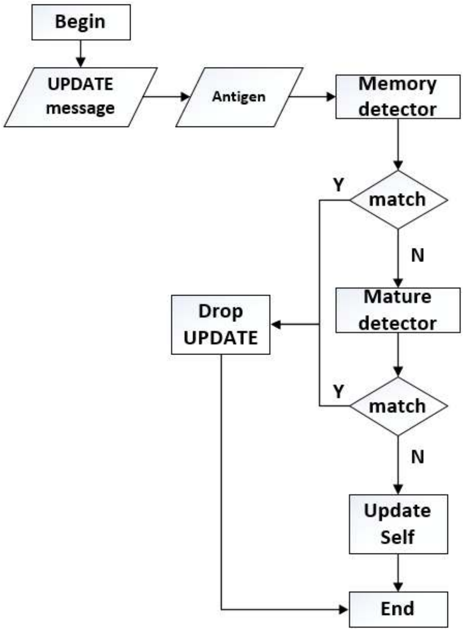


FIGURE 6. Attack detecting flowchart.

4) *E* DYNAMIC EVOLUTION

Define the evolution equation of E

$$E(t) = E(t-1) + E_n(t) - E_d(t) \quad t \geq 1. \quad (9)$$

When $t = 0$, there is $M = \varnothing$. Memory detector also has a death mechanism. Memory detectors are used to detect prefix hijacking attacks of known features, just like nonspecific immunity in BIS.

D. ATTACK DETECTION

In this section we mainly introduce the design and deployment of the attack detecting. Based on the immune model, we construct an attack detection with an evidence collection as input. The flow chart of attck detecting is shown in Fig. 6. Specific steps are as follows:

1) ***Memory detectors detecting:*** We used the memory detector to detect antigen set and discard UPDATE messages that are detected as prefix hijacking attacks. In this paper, the matching threshold r of the self-tolerance period and the matching threshold r when detecting the attack are two different parameters.

2) ***Mature detectors detecting:*** We also drop the UPDATE message detected as a prefix hijacking attack. When a mature detector matches a certain number of prefix hijacking attacks in its lifetime, it evolves into a memory detector.

3) ***Update self set:*** We add the UPDATE message with normal detection results to *Self*. At the same time, keep the

dynamic update of *Self* to ensure that the number of elements of *Self* does not exceed the threshold.

4) ***Update detector set:*** We clean out the detectors in the detector set whose life cycle exceeds the threshold to ensure the usefulness of the detectors.

The implementation of attack detection is shown in algorithm 3. Algorithm 3 needs to enter the data set UPDATE.txt and output the array of prefixes that are hijacked.

```
Algorithm 3 HijackingDetect
Input: UPDATE.txt
Output: Prefix[]
Initialization();
while Read(UPDATE.txt) do
    for $j \in [1, PERIOD]$ do
        Generate_Detector(Number);
        Nsa(detector);
    end for
    if Memory_Set! = NULL then
        Memory_Detect();
        Judge();
    else
        Mature_Detect();
        Judge();
    end if
    Check(Variable);
end while
return Prefix[];
```

VI. EXPERIMENTAL EVALUATION

In this section, we implemented the proposed aPHD to evaluate performance, and compare it to other models (S-BGP (Secure-BGP) [11] and ITMM (An immune-theory-based model for monitoring inter-domain routing system) [50]).

A. EXPERIMENTAL SETTING

In this section, we implemented a prototype to evaluate the practical performance of aPHD. Since aPHD needs to be deployed on BGP routers with limited computing power, our experiments are divided into correct rates and overheads.

We implement it on the Windows 10 Education Edition workstation with eight cores at 3.5GHz Intel Xeon E5-1620 CPU and 16GB RAM. There are two indicators for evaluating the model ability to detect prefix hijacking attacks. One indicator is the true positive rate (*TP*). The formula for calculating *TP* is

$$TP = \frac{N1}{N2} \times 100\%.$$

$N1$ represents the correct detection of the number of prefix hijacking events. $N2$ represents the total number of prefix hijacking events. Another indicator is the false positive rate (*FP*). The formula for calculating *FP* is

$$FP = \frac{N3}{N4} \times 100\%.$$

$N3$ represents the number of prefix hijacking events detected by the error. $N4$ represents the total number of Nonself.

The experimental data set comes from the MRT format real-time data stream of the University of Oregon open source project RouteViews on February 1, 2019,[7] and prefix hijacking data crawled from BGP Stream. The total number of training data events is 261,102, of which normal UPDATE messages account for 99.37%, and prefix hijacking UPDATE messages account for 0.63%.

TABLE 3. BGP daily traffic statistics.

Type	Number	Kbytes
KEEPALIVEs	3,571	68
UPDATEs	1,426	89

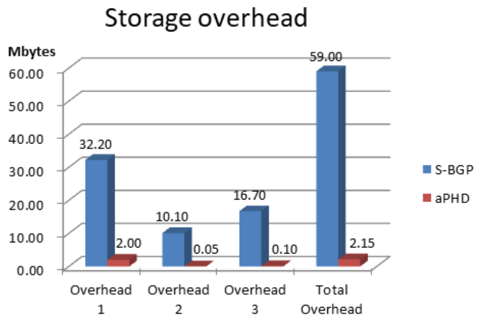


FIGURE 7. Storage overhead.

B. EVALUATION RESULT

1) OVERHEAD

Table 3 shows the nature of BGP traffic [51], they are the number and type of BGP traffic that needs to be detected. Our experiment will be based on Table 3. We simulated the H3C ER3108G router with a frequency of 1.5GHz in the workstation for the following experiment.

Due to the use of PKI, S-BGP increases the storage overhead of storing certificates and address attestations. APHD only needs to store the set of antigens and the detectors. Fig. 7 shows a comparison of the storage overhead of the two models. Meanwhile,S-BGP requires a lot of computing resources to verify certificates and address attestations, while aPHD does not require such a high computational overhead. Fig. 8 shows a comparison of the computational overhead of the two models.

The aPHD does not need to modify BGP, so there is no compatibility problem with BGP-4. Therefore, aPHD is easier to deploy and has low storage overhead and computational overhead than S-BGP.

[7] http://archive.routeviews.org/

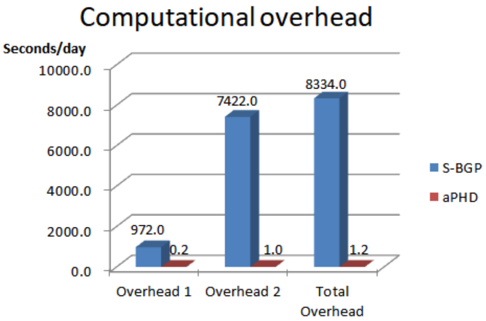


FIGURE 8. Computational overhead.

2) *TP* AND *FP*

The aPHD has multiple tunable parameters that affect *TP* and *FP*. The values of the parameters can be dynamically set according to the desired *TP* or *FP*, operating environment, and other specific conditions.

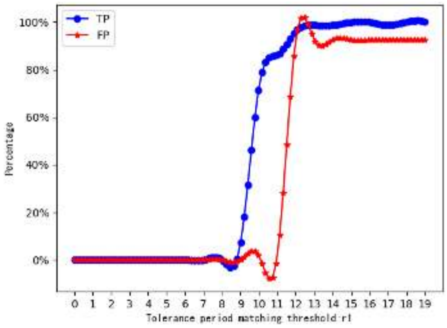


FIGURE 9. Effect of $r1$ on *TP* and *FP*.

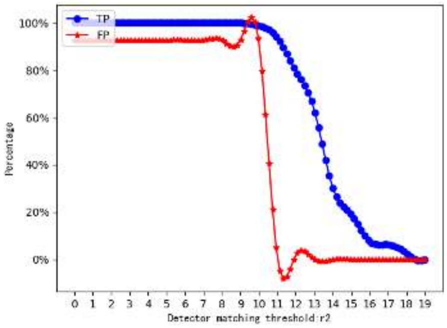


FIGURE 10. Effect of $r2$ on *TP* and *FP*.

Fig. 9 and Fig. 10 show the detection performance of the $r1$ and $r2$ influence aPHD. As $r1$ increases, both *TP* and *FP* increase gradually. The reason is that as $r1$ increases, the generation efficiency of the detectors decreases, and the detection efficiency increases. As $r2$ increases, both *TP* and *FP* decrease. The reason is that as $r2$ increases, the detection difficulty of the detector decreases. It can be seen that setting the matching threshold r of the self-tolerance period and the matching threshold r of the detector and the antigen to two different parameters can improve the model performance.

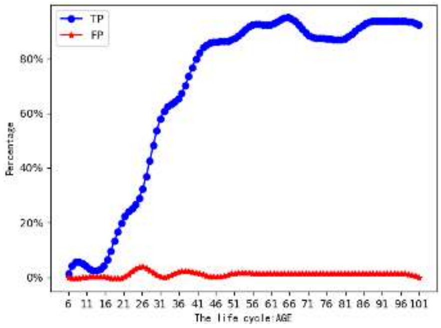


FIGURE 11. Effect of *AGE* on *TP* and *FP*.

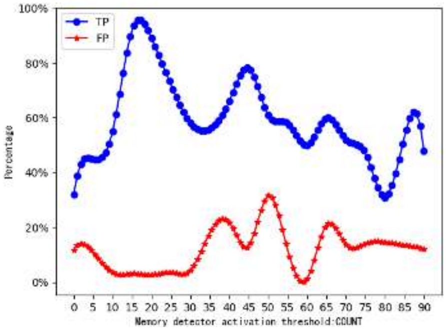


FIGURE 12. Effect of *COUNT* on *TP* and *FP*.

As shown in Fig. 11, *TP* increases as *AGE* increases. The reason is that *AGE* determines the survival of the detector. If the *AGE* is low, most of the detectors are eliminated at an early stage, resulting in a very low *TP*.

As shown in Fig. 12, suitable values can be found such that *TP* is higher and *FP* is lower. If the value of *COUNT* is small, the mature detector can easily become a memory detector. Since the memory detector does not experience multiple similar matches, the matching efficiency of the memory detector is very low. If the value of *COUNT* is large, the mature detector is difficult to become a memory detector. Due to the small number of memory detectors, aPHD did not collect features of similar attacks, resulting in similar attacks being unrecognized.

As shown in Fig. 13, the presence of a suitable *PERIOD* results in a higher *TP* and a lower *FP*. When *PERIOD* is low, the number and chance of matching the detector with the *Self* is small, which makes it easy for the generated detector to detect the *Self* as a prefix hijacking event, resulting in a lower *TP* and higher *FP* for the prefix hijacking detection model. When *PERIOD* is too high, the prefix hijacking detection model cannot provide an effective number of detectors in the early detection, which results in an unsatisfactory detection effect in the early stage.

As shown in Fig. 14, as *RANDOM_NUM* increases, *TP* gradually increases and *FP* gradually decreases. Because the greater the number of detectors, the more diverse the detectors, the better the ability to identify prefix hijacking events. The larger the value of *RANDOM_NUM*, the better, regardless of other conditions. However, the value of *RANDOM_NUM* in real-world environments is limited by

TABLE 4. Experimental parameter setting.

Parameter	Describe	Value
$r1$	Tolerance period matching threshold	11
$r2$	Detector matching threshold	12
AGE	Life cycle	96
$COUNT$	Memory detector activation threshold	6
$PERIOD$	Tolerance period	11
$RANDOM\_NUM$	Number of randomly generated detectors	98

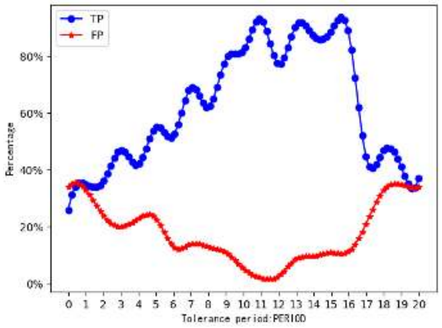


FIGURE 13. Effect of *PERIOD* on *TP* and *FP*.

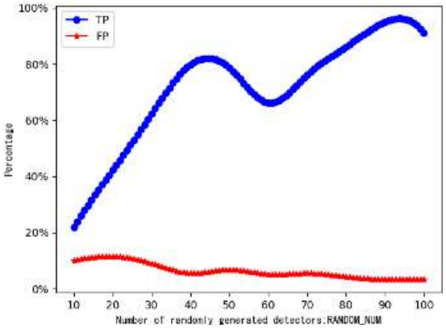


FIGURE 14. Effect of *RANDOM_NUM* on *TP* and *FP*.

TABLE 5. Comparative experiment.

Model	aPHD	ITMM
# of events	167	167
TP(%)	94.9%	89.4%
FP(%)	1.3%	9.1%

factors such as computation time, storage space, and computing power. A large number of detectors are accompanied by a large number of self-tolerance matches, which consume a lot of computational overhead and time overhead.

To get the best performance of the model, we should choose the best parameters. After repeated experiments to compare *TP* and *FP*, we get the appropriate parameters, as shown in Table 4. At this point, the model correctly identified 97.5% of the prefix hijacking attacks with a false positive rate of only 1.3%.

Under the same experimental conditions, compare the ability of the model in ITMM with aPHD to detect prefix hijacking attack. Since aPHD is a model specifically for prefix hijacking attacks, it is slightly better than ITMM in terms of detection capability, as shown in Table 5.

VII. CONCLUSION

In this paper, we studied the prefix hijacking issue in BGP security and proposed a prefix hijacking detection model, aPHD, with high accuracy, low false positive rate, and low overhead. In aPHD, we use the idea of antibody-binding antigen in immune network theory to detect attacks, which ensures the detection efficiency and reduces the false positive rate. We also added a dynamic evolution mechanism to ensure the adaptability and robustness of aPHD. Finally, we analyzed the performance of the proposed scheme from both theoretical and experimental aspects. The detailed performance evaluation demonstrates the rationality and feasibility of our proposed scheme for the practical use. The limitation of aPHD is that it requires repeated experiments to determine the parameters to achieve maximum detection efficiency in deployment. Once the environmental factors such as the number of messages in the BGP router change, aPHD needs to recalculate the appropriate parameters.

REFERENCES

[1] R. Hiran, N. Carlsson, and N. Shahmehri, "Collaborative framework for protection against attacks targeting BGP and edge networks," *Comput. Netw.*, vol. 122, pp. 120–137, Jul. 2017.

[2] P.-A. Vervier, O. Thonnard, and M. Dacier, "Mind your blocks: On the stealthiness of malicious BGP hijacks," in *Proc. NDSS*, 2015, pp. 1–15.

[3] Y. Rekhter, T. Li, and S. Hares, *A Border Gateway Protocol 4 (BGP-4)*, document RFC 4271, Internet Eng. Task Force, Fremont, CA, USA, Jan. 2006. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4271.txt

[4] J.-K. Yun, B. Hong, and Y. Kim, "The policy-based AS_PATH verification to monitor AS path hijacking," in *Proc. SECURWARE*, 2014, p. 31.

[5] P. Mohapatra, J. Scudder, D. Ward, R. Bush, and R. Austein, *BGP Prefix Origin Validation*, document IETF RFC 6811, 2013.

[6] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. D. McDaniel, and A. D. Rubin, "Working around BGP: An incremental approach to improving security and accuracy in interdomain routing," in *Proc. NDSS*, vol. 23, 2003, p. 156.

[7] B. Dickson, *Route Leaks-Definitions*, document draft-dickson-sidr-route-leak-def-01, The Internet Engineering Task Force Trust, SIDR Internet Draft, Mar. 2012, p. 9.

[8] A. Mitseva, A. Panchenko, and T. Engel, "The state of affairs in BGP security: A survey of attacks and defenses," *Comput. Commun.*, vol. 124, pp. 45–60, Jun. 2018.

[9] Z. Qian, Z. M. Mao, Y. Xie, and F. Yu, "On network-level clusters for spam detection," in *Proc. NDSS*, 2010, pp. 1–17.

[10] G. C. Tjhai, M. Papadaki, S. M. Furnell, and N. L. Clarke, "The problem of false alarms: Evaluation with snort and DARPA 1999 dataset," in *Proc. Int. Conf. Trust, Privacy Secur. Digit. Bus.* Berlin, Germany: Springer, 2008, pp. 139–150.

[11] S. Kent, C. Lynn, and K. Seo, "Secure border gateway protocol (S-BGP)," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 4, pp. 582–592, Apr. 2000.

[12] R. White, "Securing BGP through secure origin BGP (soBGP)," *Bus. Commun. Rev.*, vol. 33, no. 5, p. 47, 2003.

[13] M. Lepinski and S. Kent. (2012). *An Infrastructure to Support Secure Internet Routing*. [Online]. Available: http://tools.ietf.org/html/rfc6480

[14] J. Gersch and D. Massey, "ROVER: Route origin verification using DNS," in *Proc. 22nd Int. Conf. Comput. Commun. Netw.*, Jul./Aug. 2013, pp. 1–9.

[15] Z. Le, N. Xiong, B. Yang, and Y. Zhou, "SC-OA: A secure and efficient scheme for origin authentication of interdomain routing in cloud computing networks," in *Proc. IEEE Int. Parallel Distrib. Process. Symp.*, May 2011, pp. 243–254.

[16] Y.-C. Hu, A. Perrig, and M. Sirbu, "SPV: Secure path vector routing for securing BGP," in *Proc. ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 179–192, Oct. 2004.

[17] K. Butler, P. McDaniel, and W. Aiello, "Optimizing BGP security by exploiting path stability," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, Oct./Nov. 2006, pp. 298–310.

[18] P. C. Van Oorschot, T. Wan, and E. Kranakis, "On interdomain routing security and pretty secure BGP (psBGP)," *ACM Trans. Inf. Syst. Secur.*, vol. 10, no. 3, p. 11, Jul. 2007.

[19] Q. Li, M. Xu, J. Wu, X. Zhang, P. P. C. Lee, and K. Xu, "Enhancing the trust of Internet routing with lightweight route attestation," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 691–703, Apr. 2011.

[20] S. Cho, R. Fontugne, K. Cho, A. Dainotti, and P. Gill, "BGP hijacking classification," in *Proc. Netw. Traffic Meas. Anal. Conf.*, Jun. 2019, pp. 25–32.

[21] W. Xu, D. Chang, and X. Li, "On the classification and false alarm of invalid prefixes in RPKI based BGP route origin validation," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage.*, Apr. 2019, pp. 654–658.

[22] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang, "An analysis of BGP multiple origin AS (MOAS) conflicts," in *Proc. 1st ACM SIGCOMM Workshop Internet Meas.*, Nov. 2001, pp. 31–35.

[23] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang, "Detection of invalid routing announcement in the Internet," in *Proc. Int. Conf. Dependable Syst. Netw.*, Jun. 2002, pp. 59–68.

[24] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang, "PHAS: A prefix hijack alert system," in *Proc. USENIX Secur. Symp.*, 2006, vol. 1, no. 2, p. 3.

[25] J. Karlin, S. Forrest, and J. Rexford, "Pretty good BGP: Improving BGP by cautiously adopting routes," in *Proc. IEEE Int. Conf. Netw. Protocols*, Nov. 2006, pp. 290–299.

[26] C. Zheng, L. Ji, P. Dan, W. Jia, and P. Francis, "A light-weight distributed scheme for detecting ip prefix hijacks in real-time," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, Aug. 2007, pp. 277–288.

[27] V. Alaparthy and S. D. Morgera, "Modeling an intrusion detection system based on adaptive immunology," *Int. J. Interdiscipl. Telecommun. Netw.*, vol. 11, no. 2, pp. 42–55, 2019.

[28] X. Shi, X. Yang, Z. Wang, Y. Xia, and J. Wu, "Detecting prefix hijackings in the Internet with argus," in *Proc. Internet Meas. Conf.*, Nov. 2012, pp. 15–18.

[29] J. D. Farmer, N. H. Packard, and A. S. Perelson, "The immune system, adaptation, and machine learning," *Phys. D, Nonlinear Phenomena*, vol. 22, nos. 1–3, pp. 187–204, Oct./Nov. 1986.

[30] J. Timmis and M. Neal, "A resource limited artificial immune system for data analysis," in *Research and Development in Intelligent Systems XVII*. London, U.K.: Springer, 2001, pp. 19–32.

[31] J. Timmis and M. Neal, "Investigating the evolution and stability of a resource limited artificial immune system," in *Proc. Special Workshop Artif. Immune Syst., Gentic Evolutionay Comput. Conf.*, 2000, pp. 40–41.

[32] L. De Castro and J. Timmis, "Artificial immune systems: A novel paradigm to pattern recognition," in *Artificial Neural Networks in Pattern Recognition*, J. M. Corchado, L. Alonso, and C. Fyfe, Eds. London, U.K.: Univ. Paisley, 2002, pp. 67–84.

[33] Z. Ji and D. Dasgupta, "Real-valued negative selection algorithm with variable-sized detectors," in *Proc. Genetic Evol. Comput. Conf.* Berlin, Germany: Springer, 2004, pp. 287–298.

[34] M. Anaya, D. A. Tibaduiza, and F. Pozo, "A bioinspired methodology based on an artificial immune system for damage detection in structural health monitoring," *Shock Vibrat.*, vol. 2015, May 2015, Art. no. 648097.

[35] A. Secker, A. A. Freitas, and J. Timmis, "AISEC: An artificial immune system for e-mail classification," in *Proc. Congr. Evol. Comput.*, vol. 1, Dec. 2003, pp. 131–138.

[36] T. Li, "Dynamic detection for computer virus based on immune system," *Sci. China F, Inf. Sci.*, vol. 51, no. 10, pp. 1475–1486, Oct. 2008.

[37] S. Jamali and R. Fotohi, "DAWA: Defending against wormhole attack in MANETs by using fuzzy logic and artificial immune system," *J. Supercomput.*, vol. 73, no. 12, pp. 5173–5196, 2017.

[38] C. Blum, J. A. Lozano, and P. P. Davidson, "An artificial bioindicator system for network intrusion detection," *Artif. Life*, vol. 21, no. 2, pp. 93–118, 2015.

[39] R. Pasti and L. N. de Castro, "An immune and a gradient-based method to train multi-layer perceptron neural networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jul. 2006, pp. 2075–2082.

[40] R. Mahajan, D. Wetherall, and T. Anderson, "Understanding BGP misconfiguration," in *Proc. ACM SIGCOMM Comput. Commun. Rev.*, Oct. 2002, vol. 32, no. 4, pp. 3–16.

[41] N. Elamathi, S. Jayashri, and R. Pitchai, "Enhanced secure communication over inter-domain routing in heterogeneous wireless networks based on analysis of BGP anomalies using soft computing techniques," *Soft Comput.*, vol. 23, no. 8, pp. 2735–2746, Apr. 2019.

[42] T. Okamoto and M. Tarao, "An artificial immunity-enhancing module for Internet servers against cyberattacks," *Artif. Life Robot.*, vol. 23, no. 3, pp. 292–297, Sep. 2018.

[43] J. M. Vidal, A. L. S. Orozco, and L. J. G. Villalba, "Adaptive artificial immune networks for mitigating DoS flooding attacks," *Swarm Evol. Comput.*, vol. 38, pp. 94–108, Feb. 2018.

[44] F. Zhang and Y. Ma, "Using IRP with a novel artificial immune algorithm for windows malicious executables detection," in *Proc. Int. Conf. Prog. Inform. Comput.*, Dec. 2016, pp. 610–616.

[45] M. Tabatabaefar, M. Miriestahbanati, and J.-C. Grégoire, "Network intrusion detection through artificial immune system," in *Proc. Annu. IEEE Int. Syst. Conf.*, Apr. 2017, pp. 1–6.

[46] L. Blunk, M. Karir, and C. Labovitz, "Multi-threaded routing toolkit (MRT) routing information export format," document RFC 6396, Internet Engineering Task Force, Fremont, CA, USA 2011.

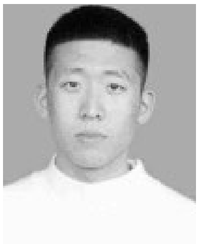
[47] R. Medzhitov and C. A. Janeway, "Decoding the patterns of self and nonself by the innate immune system," *Science*, vol. 296, no. 5566, pp. 298–300, 2002.

[48] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, "Self-nonself discrimination in a computer," in *Proc. IEEE Comput. Soc. Symp. Res. Secur. Privacy*, May 1994, pp. 202–212.

[49] D. Dasgupta and S. Forrest, "Novelty detection in time series data using ideas from immunology," in *Proc. Int. Conf. Intell. Syst.*, 1996, pp. 82–87.

[50] Y. Guo and Z. Wang, "An immune-theory-based model for monitoring inter-domain routing system," *Sci. China Inf. Sci.*, vol. 55, no. 10, pp. 2358–2368, Oct. 2012.

[51] S. T. Kent, C. Lynn, J. Mikkelson, and K. Seo, "Secure border gateway protocol (S-BGP)-real world performance and deployment issues," in *Proc. NDSS*, 2000, pp. 1–14.



JIAN ZHANG received the B.S. degree in information security from Guangxi University, China, in 2017, where he is currently pursuing the M.S. degree. His research interests include information security and BGP security.

DAOFENG LI received the B.S. and M.S. degrees in applied mathematics from Guangxi University for Nationalities and Chengdu University of Technology, China, in 2001 and 2005, respectively, and the Ph.D. degree in cryptography from the Beijing University of Posts and Telecommunications, in 2008. He has been an Associate Professor with the School of Computer, Electrical and Information, Guangxi University, since 2008. He has authored or coauthored over 20 technical articles in journals and conference proceedings. His current research interests include cryptography and information security.

BOWEN ZHAO received the B.S. and M.S. degrees in information security from the Hunan University of Science and Technology and Guangxi University, China, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree with the South China University of Technology. His research interests include information security and privacy preserving in cloud computing, and big data.