Hindawi
Security and Communication Networks
Volume 2019, Article ID 6045251, 11 pages
https://doi.org/10.1155/2019/6045251

Research Article

A Novel Device Identification Method Based on Passive Measurement

Wei Sun,[1] **Hao Zhang**,[2] **Li-jun Cai**,[2] **Ai-min Yu**,[2] **Jin-qiao Shi**,[3] **and Jian-guo Jiang**[2]

[1]*Beijing Jiaotong University, School of Computer and Information Technology, Beijing 100044, China*
[2]*Chinese Academy of Sciences, Institute of Information Engineering, Beijing 100093, China*
[3]*Beijing University of Posts and Telecommunications, School of Cyberspace Security, Beijing 100876, China*

Correspondence should be addressed to Wei Sun; 11112075@bjtu.edu.cn

Received 27 February 2019; Accepted 19 May 2019; Published 23 June 2019

Guest Editor: Fagen Li

Nowadays, with the continuous integration of production network and business network, more and more Industrial Internet of Things and Internal Office Network have been interconnected and evolved into a large-scale enterprise-level intraindustry network. Terminal devices are the basic units of internal network. Accurate identification of the type of device corresponding to the IP address and detailed description of the communication behavior of the device are of great significance for conducting network security risk assessment, hidden danger investigation, and threat warning. Traditional cyberspace surveying and mapping techniques take the form of active measurement, but they cannot be transplanted to large-scale intranet. Resources or specific targets in internal networks are often protected by firewalls, VPNs, gateways, and other technologies, so they are difficult to analyze and determine by active measurement. In this paper, a passive measurement method is proposed to identify and characterize devices in the network through real traffic data. Firstly, a new graph structure mining method is used to determine the server-like devices and host-like devices; then, the NAT-like devices are determined by quantitative analysis of traffic; finally, by qualitative analysis of the NAT-like device traffic, it is determined whether there are server-like devices behind the NAT-like device. This method will prove to be useful in identifying all kinds of devices in network data traffic, detecting unauthorized NAT-like devices and whether there are server-like devices behind the NAT-like devices.

1. Introduction

With the rapid development of information technology, the integration of production network and business network has become a reality. The interconnection of Industrial Internet of Things (IoT) and Internal Office Network has become a new networking trend, which not only greatly improves the production efficiency, but also achieves tight coupling of business work and production scheduling. A series of Stuxnet incidents that have erupted since 2010 are typical events that invade other networks from industrial IoT intrusions, which have had a huge negative impact on the global network, making losses difficult to measure. The methods of cyberattacks are more diverse, the devices targeted are more extensive, and the exploited vulnerabilities are more complicated, resulting in more serious threats. Specific security vulnerabilities often threaten a certain type of terminal device, which highlights the importance and urgency of fully understanding the distribution and attributes of cyberspace terminal devices.

Cyberspace surveying and mapping technology is an extension of network measurement, and network measurement technology is used for network mapping [1–3]. According to the measurement method, cyberspace surveying and mapping technology can be divided into active measurement and passive measurement [4–9]. For large-scale internal networks, especially those connected to industrial IoT, active measurement is not a good way. The main reasons are as follows: (1) Industrial IoT requires high real-time performance and stability and cannot tolerate the large number of data detection packages generated by active measurement. (2) Due to the limitation of the internal network transmission bandwidth, active measurement cannot be performed because it is easy to cause network congestion. (3) Active measurement methods are easily identified as attacks due to the placement of various security products in the internal network. (4) The

internal network has strict network boundaries and access control methods, so it is difficult to achieve reachability and coverage. In response to these active measurement problems, this paper presents a cyberspace surveying and mapping model guided by passive measurements. It can improve measurement results, reduce measurement complexity, and reduce network load.

In a large-scale industry internal network, hundreds of millions of traffic data are generated every day. Analyzing the data and mining their value are in line with the rules of passive measurement. The traffic data reflects the real state of the internal network and has a strong practical significance for comprehensively collecting the distribution and warning the potential threats of the devices in the cyberspace [10–13].

According to the devices corresponding to the IP addresses in the traffic data, we divide them into three categories: server-like devices, host-like devices, and NAT-like devices. Server-like devices include office application service devices, industrial production service devices, cloud computing devices, and data storage devices. Host-like devices include office computers and data collection terminals. NAT-like devices include firewalls, routers, gateways, and other address translation devices.

Normally, NAT devices [14] appearing in traffic data act as terminal devices, which makes devices hidden behind NAT-like devices arbitrarily access service resources in the internal network. Attackers may use devices hidden behind NAT-like devices to engage in illegal activities, such as launching an attack [15, 16], scanning the entire network, stealing data, maliciously spreading, or providing data services for violations, which can cause very serious problems. Therefore, it is necessary to periodically detect and filter out unauthorized NAT-like devices by analyzing data traffic.

In this paper, our goal is to identify the devices appearing in traffic data in order to achieve the surveying and mapping description of the internal network. We draw on the method of social relationship analysis and propose an unsupervised learning framework based on graph feature analysis and traffic analysis. Graph feature analysis is used to separate server-like devices and host-like devices, and traffic analysis is used to identify NAT-like devices. It should be noted that the communication relationship between the devices behind the NAT device does not appear in the traffic data, but these devices also belong to the device assets in the network. Therefore, we describe a set of validation analysis methods to identify server-like devices hidden behind NAT-like devices.

This paper makes the following research contributions:

(1) We use an unsupervised learning algorithm to classify network devices that appear in data traffic, which makes it feasible in internal networks compared to active measurements.

(2) We propose a graph feature analysis method for attribute mapping of all devices.

(3) We propose a description method, which is used to describe the attributes of the terminal devices.

(4) We propose a validation analysis method to determine whether there are server-like devices behind the NAT-like devices.

The model cannot only accurately identify the server-like devices and host-like devices, but also identify NAT-like devices with high accuracy. On this basis, a qualitative traffic analysis method is used to determine whether a server-like device exists behind the NAT-like device. By comparing the identified NAT-like devices with the asset list, an unauthorized NAT-like device can be found, which can eliminate network security risks, submitting devices' feature information to the security device and analyzing such information, which can improve the accuracy of the alarm.

2. Related Work

Zhao Fan et al. [17] describe the concept of cyberspace surveying and mapping. From the perspective of the Internet, it is summarized as the use of network detection, information collection, data processing, and data analysis to obtain physical resources and virtual resources. Through the positioning algorithm and the association analysis method, the physical resource is mapped to the geographic location, and the virtual resource is mapped to the simulated location. Finally, the detection results are drawn, which can intuitively reflect the state and development trend of cyberspace resources. Kohno et al. [18] propose a method for detecting network devices based on the offset value of the device clock. Fink [19] improves the method, introduces linear regression statistical method to judge the clock offset, and gives a calculation formula under certain degree of accuracy, so that the accuracy of such device recognition is controllable.

Wang Jianwei et al. [20], combining the degree of node and its neighbor node features, propose a method to evaluate the importance of the node with only local information, and the time performance is greatly improved. Kitsak et al. [21] propose Kl-shell in 2010, which is a method to determine the core of the network. The core idea of this method is to iteratively layer the nodes in the graph. The higher the number of layers, the more important the node is.

Gokcen et al. [22] generate 40 traffic attributes through the Net-Mate [23] tool and find the most effective eight attributes for identifying NAT devices using C4.5 [24] and Naive Bayes [25] machine learning algorithms. Another identification method in this paper uses payload information. However, in the internal network, the available traffic attributes are limited, and many attributes are not available due to confidentiality restrictions or encryption processing. Li Rui et al. [26] firstly express network traffic with 8-dimensional features, then filter the network traffic with an active value, and finally use the support vector machine to make two classes of NAT devices or host devices. Different servers store different resources. There may exist such a NAT-like device; the traffic generated with a particular server-like device is very high, but the level is not obvious in the total traffic analysis. Traversing server-like devices in turn, filtering out the NAT-like device connected to them by analyzing the traffic may achieve better classification results.

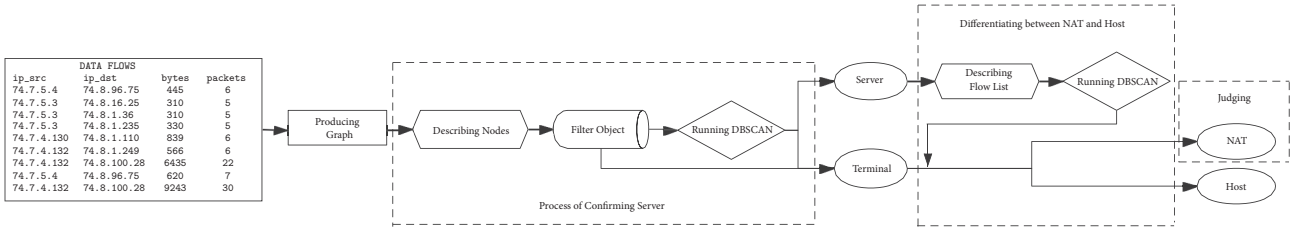


Figure 1: Model framework.

3. Recognition Method Based on Cluster Analysis

Figure 1 shows the framework of this method. The input we use is the traffic quad (source IP, destination IP, traffic size, number of packets), and each record represents a communication between the IP pair. The traffic data is converted into a graph structure; that is, the device IP address is used as a node, the communication relationship of the traffic data is used as an edge, and the information carried by the edge will be a triplet; the specific content will be described later. We will use the density-based clustering algorithm to classify devices twice. The whole process can be divided into three parts: One is to use explicit value to describe the importance of each node in the graph, then cluster analysis is used to identify server-like devices. The second is to obtain a list of terminal devices connected to the above server-like devices and use explicit numbers to describe the traffic level of the terminal device to server-like devices, thereby performing cluster analysis and separating the NAT-like devices and the host-like devices. Both of the above devices belong to the terminal device. The third is the analysis method of NAT-like device validation, to determine the existence of server-like devices behind. Next, we will discuss these three processes in detail. Finally, we briefly review the density-based clustering algorithm—DBSCAN.

3.1. Process of Confirming Server-Like Devices. Figure 2 is a simple example of node distribution, in which two central nodes represent two servers while the other nodes represent terminal devices, and the intersection part of the middle side indicates that these terminal devices communicate with both servers. We can classify nodes by characterizing node features.

Definition 1. Node degree refers to the number of edges associated with the node, also known as correlation degree.

The node degree of node i is formulized as

$$k_i = \left|\left\{e_{ij} \mid j \in V\right\}\right| \tag{1}$$

where V represents the node set of a graph and e_{ij} represents an edge between node i and node j.

Definition 2. The average degree of neighborhood refers to the average correlation degree of nodes in the neighborhood list of the node.

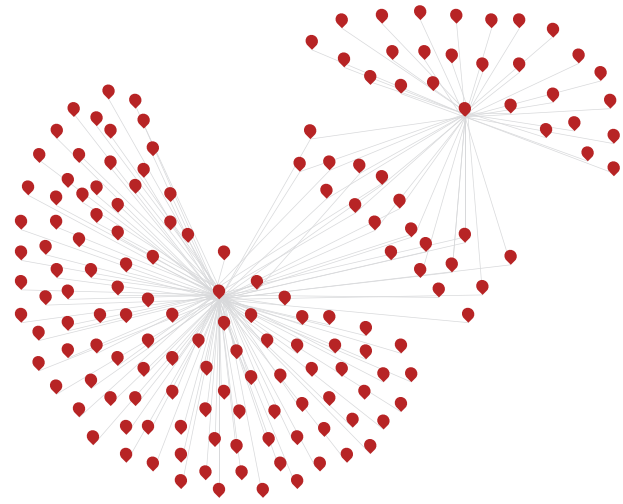


Figure 2: Graph structure example.

The average degree of neighborhood of node i is formulized as

$$k_{nn,i} = \frac{1}{|N(\mathrm{i})|} \sum_{j \in N(i)} k_j \tag{2}$$

where $N(i)$ represents the neighborhood of node i and k_j represents the correlation degree of node j which belongs to $N(i)$.

Definition 3. Traverse the nodes in the connected graph, all the paths of length 2 whose starting points are that nodes. Record the intermediate node of each path, and define the number of occurrences of each node as *the node fortress index.*

The node fortress index of node i is formulized as

$$f_i = \left|Mid_{\{path_{mn}=2 \mid m,n \in V\}}(\mathrm{i})\right| \tag{3}$$

where $\{path_{mn} = 2 \mid m, n \in V\}$ represents all paths of length 2 in the graph, and $Mid(i)$ indicates that the node i is exactly the intermediate node of the path.

Firstly, we calculate the average degree of neighborhood of each node, and then nodes are sorted from small to large. Stick out a mile; the average neighborhood of the server-like nodes is small. Secondly, there is a threshold α ($0< \alpha <1$), which means that the top α devices are selected to determine the server-like node. And the size of α depends on the experimental results. As a result, the node set which is used to

Input: G: Graph based on netflow; α: Select the front α section to filter nodes; *eps*: Neighbor radius to form a density area; *minPts*: Minimum number to form a high density area;
Output: Label of nodes: server or terminal;
$H \longleftarrow []$
$N \longleftarrow []$
for each *node* in G **do**
  $k_{node} \longleftarrow K(node)$ //calculate average neighbor degree of the node
  Put $(node, k_{node})$ in list H
$H \longleftarrow$sorting *node* according to k_{node} //ascending order
$N \longleftarrow \alpha$ of the front *node* in H //filtering nodes
$C \longleftarrow []$
for each n in N **do**
  $d_n \longleftarrow D(n)$ //calculate stronghold index of the node
  $f_n \longleftarrow F(n)$ //calculate degree of the node
  Put (d_n, f_n) in list C
Running DBSCAN with *eps* and *minPts* //with dataset C
return *outliers* //*outlier* represent Server, others represent Terminal

Algorithm 1: Process of server-like node detection.

identify the server-like nodes is reduced, and we will explain the details and prove that this step is necessary in later section. Thirdly, the two-dimensional features of all nodes in the node set are collected as the data to be clustered. As defined above, one of the characteristics is the correlation degree of each node, and the other is the fortress index of each node. Finally, we use DBSCAN algorithm for clustering analysis where outliers belong to server-like node and intracluster points belong to terminal node. Algorithm 1 shows the detailed process of identifying server-like node.

3.2. Differentiating between NAT-Like Devices and Host-Like Devices. The storage resources and uses vary between server-like devices, so the traffic levels of terminal devices for different server-like devices may also be different. At the same time, we assume that if two hosts connected to the same server-like device they may have similar traffic levels. Based on the situation we suggest above, for each server-like device, we find the list of terminal devices connected to it and then do clustering analysis. We believe that if a terminal device is a NAT-like device, it will connect with the corresponding server-like device more times, have more data packets, and generate more traffic. Therefore, we collect the three-dimensional features of all nodes in the terminal devices list as the data to be clustered. Then we use DBSCAN algorithm for clustering analysis where outliers belong to NAT-like devices and intracluster points belong to hosts. Imagine that if there are no NAT-like devices in a terminal devices list, all data will form a cluster without outliers. On the contrary, the NAT-like devices will become outliers and be marked. We repeat the above steps for each server-like node and finally intersect the results. Algorithms 2 gives the detailed process of distinguishing NAT-like devices from terminal devices.

3.3. Determining Whether Server-Like Devices Exist behind the NAT-Like Devices. According to the rules of capturing traffic data in intranet, the connection record between host and host is not included in the traffic data. If the connection record between a host-like device and a host-like device is found in the data flow, then it is certain that at least one of the communication parties is a NAT-like device, and the server-like device is hidden behind the NAT-like device.

We should also consider the connection records of server-like devices as source IP and NAT-like devices as destination IP. If the protocol used is TCP, then it is certain that the server-like device is connected to the server-like devices hidden behind the NAT-like device; if the protocol is UDP and the connection mode is query instead of answer, the server-like device sends query message to the server-like device behind the NAT-like device, proving that the server-like device is connected to the server-like device hidden behind the NAT-like device, as shown in Algorithms 3.

3.4. Density-Based Clustering Algorithm: DBSCAN. DBSCAN is an unsupervised machine learning algorithm, which assumes that classes can be determined by the compactness of the sample distribution; that is to say, the samples of the same class are closely linked.

DBSCAN algorithm has two parameters. One is the radius (Eps), which represents the circular neighborhood centered on fixed point P. The other parameter is the minimum number of points (MinPts) in the neighborhood centered on fixed point P. If there are at least MinPts in the neighborhood of Eps, the fixed point P is called the core point. If Q is located in the ϵ-neighborhood of P and this P is the core object, then Q is said to be directly density-reachable from P. For P and Q, if there is a sample sequence $p_1, p_2, \ldots, p_T$, satisfying $p_1 = P$ and $p_T = Q$, and if p_{t+1} is directly density-reachable from p_t, then P is said to be density-reachable from Q. That is to say, the density-reachable relation satisfies the transmissibility. For P and Q, if there is a core point m, so that

Input: G: Graph based on netflow; S: List of Serve Detected from Algorithm 1; T: List of Terminal Detected from Algorithm 1; *eps*: Neighbor radius to form a density area; *minPts*: Minimum number to form a high density area;
Output: Label of nodes: NAT or host;

```
N ⟵ []
for each s in S do  //ergodic per server
    L ⟵ []
    for each t in T do  //ergodic per terminal
        if from s to t exist edge in G then  //G belongs to undirected graph
            Put t in list L  //creat special terminal list about this server
    C ⟵ []
    for each l in L do
        t_l ⟵ T(l)  //time of communication between s and l
        p_l ⟵ P(l)  //package's number between s and l
        f_l ⟵ F(l)  //total flow data between sand l
        Put (t_l, p_l, f_l) in list C
    O ⟵ []
    Running DBSCAN with eps and minPts  //with dataset C
    O ⟵ outliers  //collecting outliers with list C
    N ⟵ N ∪ O  //integer O found by servers
return N  //N represent NAT, others represent Host
```

Algorithm 2: Differ NAT-like node and host-like node.

Input: S: List of Serve Detected from Algorithm 1; N: List of NAT Detected from Algorithm 2; H: List of Host Detected from Algorithm 2; R: List of Data Flow Recording;
Output: List of NATs with Servers behind them;

```
O ⟵ []
for each r in R do  //traverse through each record
    for each n in N do  //traverse through each NAT
        if n is src of r and dst is in H then  // T -> T
            Put src in list O
        if n is dst of r and src is in S then  // S -> T
            if protocol is HTTP then
                Put dst in list O
            if protocol is UDP and pattern is query then  //query or answer
                Put dst in list O
return O  //N represent NATs with Servers behind them
```

Algorithm 3: Process of judging servers behind NATs.

both P and Q can be density-reachable from the core point m, that P and Q are density-connected.

Steps to run DBSCAN:

(a) We travel through each point to find all the core points.

(b) Starting from a core point, we expand to a region with density-reachable relation and obtain a region that contains the core point and boundary points, in which any two points are density-connected.

The density-connected samples are grouped into the same class; in this way we can get the clustering results, as shown in Figure 3.

4. Evaluation

In order to test the effectiveness of the above method in device identification in the internal network, we will describe the following four aspects.

4.1. Dataset. Our dataset comes from Elasticsearch and the data collection process is shown in Figure 4. First, the traffic information is collected into the Traffic Collection Server through the mirroring interface; then, through the inspection and procedure of the PTD software, the traffic information is transmitted to the NSQ, a distributed real-time messaging platform; finally, Logstash acts as an intermediate station and the information is copied to the Elasticsearch, a Lucene-based search server. We can get traffic records for a certain

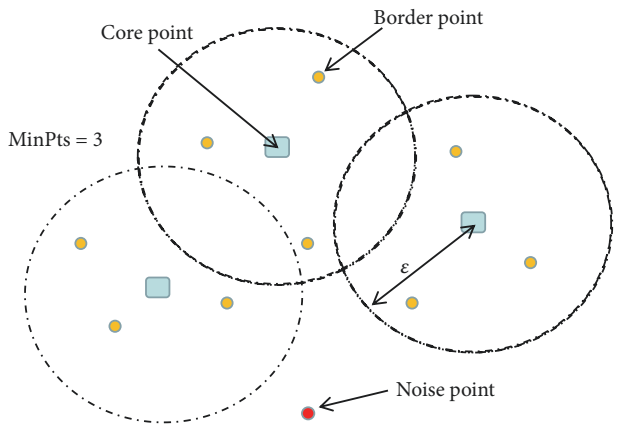


Figure 3: DBSCAN.

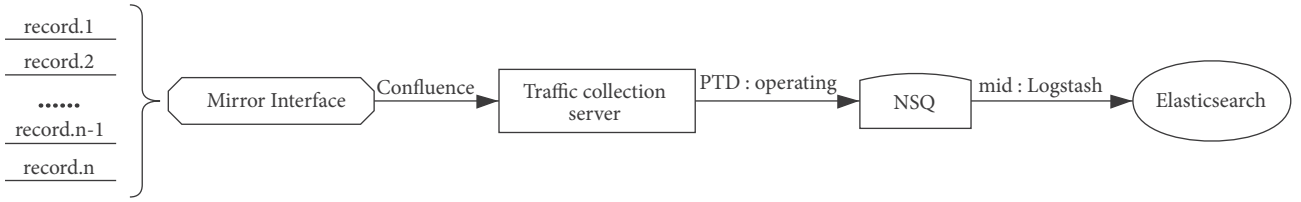


Figure 4: Process of collecting.

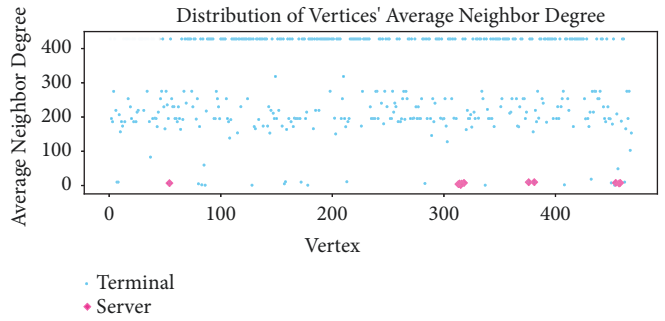


Figure 5: Line chart of the neighborhood average distribution.

day from Elasticsearch. After simple data processing, each traffic record is a four-tuple (src-ip, dst-ip, bytes, packets) that represents a communication connection between the source IP and the destination IP. In traffic data, the number of IPs is 468, and the number of records is 2,634,182. If it converted to a graph, the graph will contain 468 nodes and 1603 edges. The device asset list can display the device type corresponding to the IP address, which can be regarded as a tag set.

4.2. Selected Attributes. In this section, we discuss the two classification processes separately.

(1) Process of Confirming Server-Like Devices. There are many metrics for the importance of the nodes in graph, such as Degree Centrality, Betweenness Centrality, and Closeness Centrality. However, a single indicator only characterizes the structural features of the graph partly. In order to more accurately and comprehensively characterize the features of the network, multiple indicators are needed to reflect the features of the network. Combining theoretical research and statistical analysis, we summarize the measurement indicators of various characterizations. The neighborhood average can reduce the set of nodes to be classified. The value of the neighborhood average of the server-like nodes will always be small. On the contrary, the value of the neighborhood of the terminal nodes are almost larger, as shown in Figure 5.

We find that the association between node correlation and node fortress index is very large. The most important point is that these two indicators can be divided into two classes: server-like node and terminal node, as shown in Figure 6.

We map the above two metrics as an array into a two-dimensional space. Under the premise of adjusting the two parameters of the DBSCAN algorithm (Eps and MinPts), the two types of data points are separable. Due to the data overlap, the number of data points in the marked area in Figure 7 is 26.

(2) Differentiating between NAT-Like Devices and Host-Like Devices. As mentioned above, we believe that if a terminal

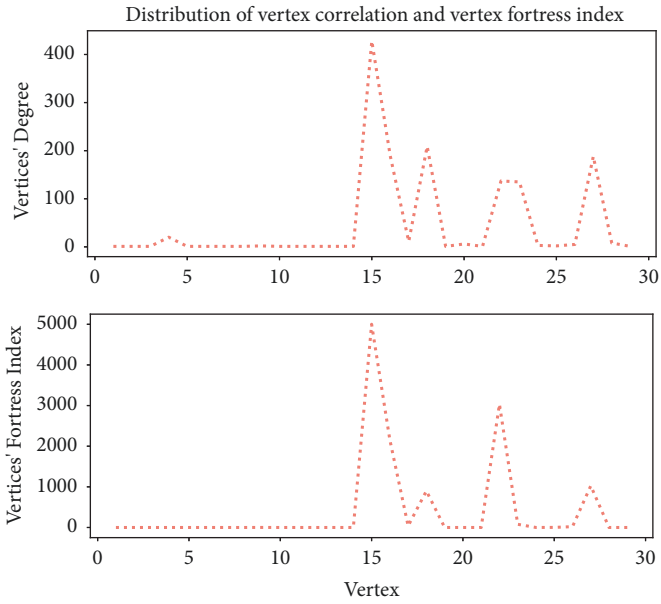


Figure 6: Distribution of node correlation and node fortress index.

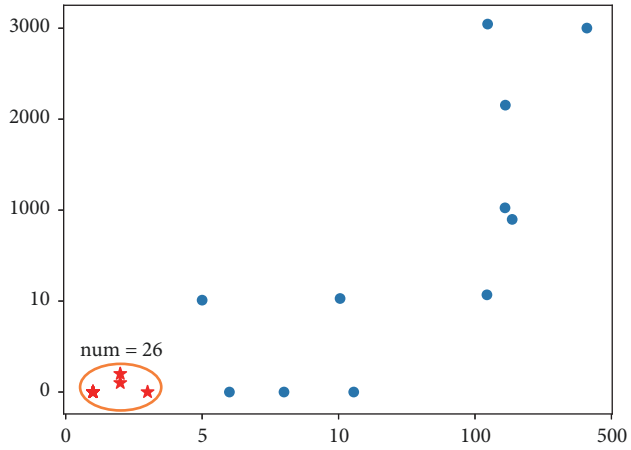


Figure 7: Distribution of data in two dimensions.

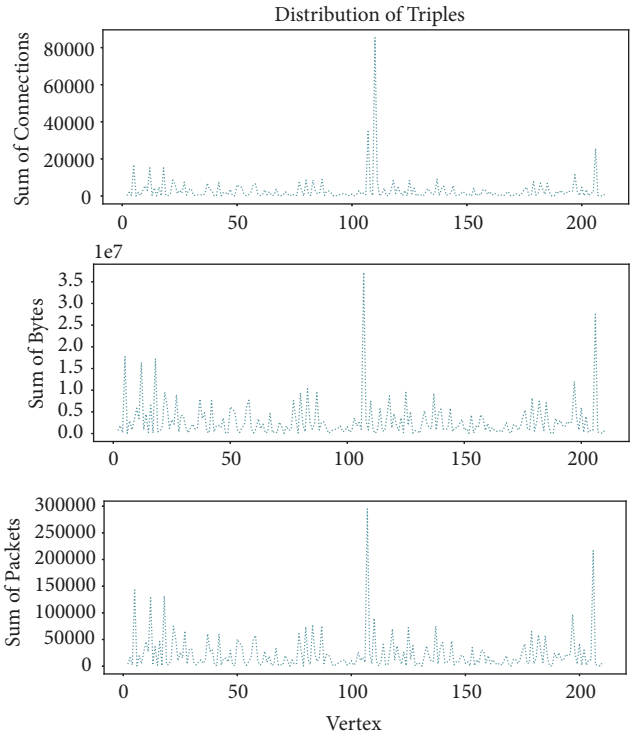


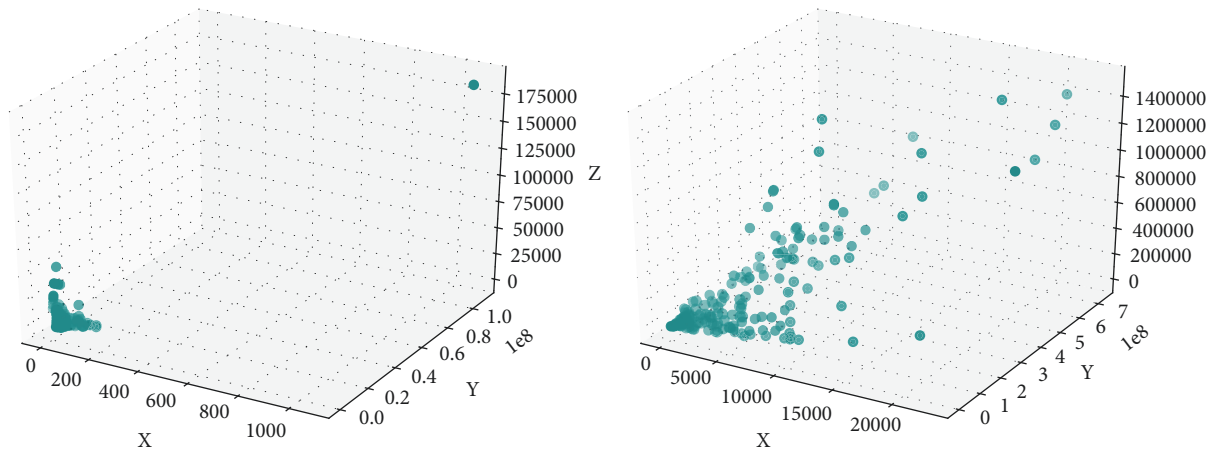
Figure 8: Distribution of triples.

device is actually a NAT-like device, it will have more connections with the corresponding server-like device. Then it will receive or send more packets as well as generating more traffic information. We have identified the server-like devices in the internal network. Then, for a server-like node, the distribution of the three indicators of the terminal devices list is shown in Figure 8, which indicates that the three indicators are roughly the same for the terminal device.

First of all, for each server-like node, we map its corresponding triplet data into 3D space. Then, we use DBSCAN algorithm for cluster analysis. In order to better demonstrate the adaptability of DBSCAN algorithm to this problem, we select two server-like nodes. One of the two does have connection with a NAT-like device, as shown in Figure 9(a). The other does not have connection with any NAT-like devices, as shown in Figure 9(b). It is obvious that there is no outlier in Figure 9(b).

4.3. Parameter Determination. In this section, we will discuss three classification processes and qualitative analysis process, respectively.

(1) Process of Confirming Server-Like Devices. In the process of confirming server-like devices, the parameter MinPts is not sensitive to clustering results, so we set the parameter MinPts value to 4 in all tests. As shown in Figure 10, with



(a) Distribution with NAT-like devices

(b) Distribution without NAT-like devices

Figure 9

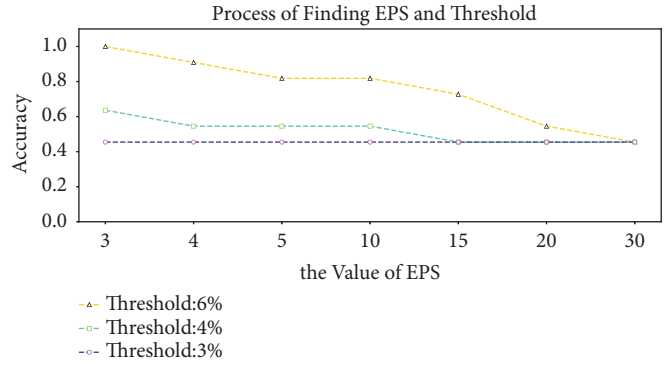


Figure 10: Accuracy of identifying server-like node.

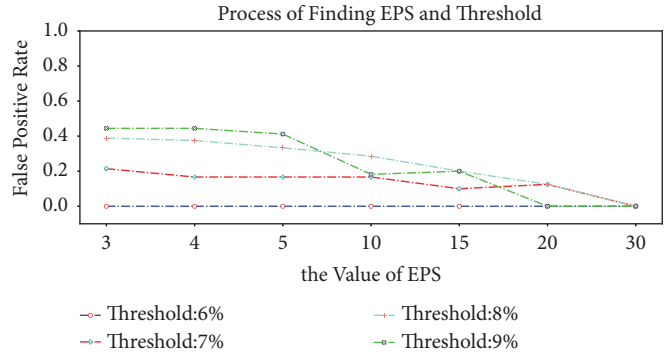


Figure 11: False rate of identifying server-like node.

the increase of Eps value, the accuracy of server-like devices recognition gradually decreases. At the same time, the minimum threshold is 0.06; otherwise, it will affect the accuracy of recognition results. When the Eps value is 3 and the threshold value is 0.06, the recognition accuracy of server-like devices is the highest. As shown in Figure 11, with the increase of Eps, the false positive rate of recognition decreases, but the recognition accuracy decreases significantly at this time. We also find that increasing the threshold will lead to greater false alarm rate. When the threshold is 0.06, the false alarm rate of the model is 0. After many tests, when Eps value is 3, MinPts value is 4, and threshold value is 0.06, the recognition effect is the best.

(2) Differentiating between NAT-Like Devices and Host-Like Devices. As shown in Figure 12, the recall rate increases with the increase of Eps value. When the Eps value is 0.7-0.9 and the MinPts value is 5, the recall rate is higher and stable. As shown in Figure 13, the accuracy decreases with the increase of Eps value. When Eps value is 0.7 and MinPts

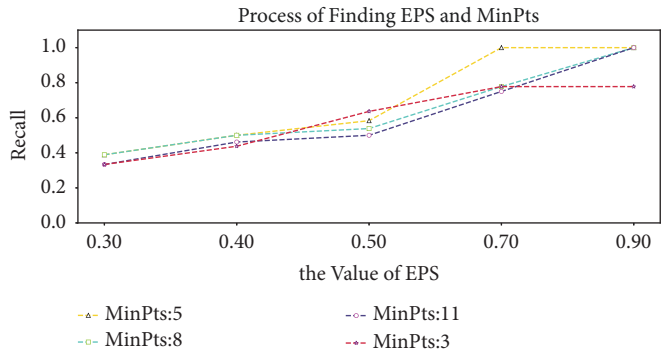


Figure 12: Recall rate of NAT-like devices.

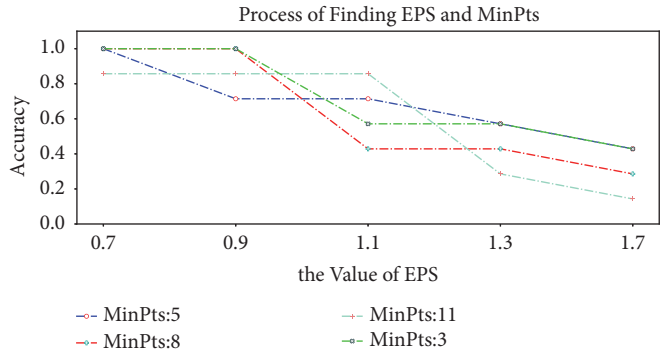


Figure 13: Accuracy of identifying NAT-like devices.

value is 3-5, the accuracy rate is higher and stable. We need to find parameter combinations with high recall and high accuracy. After many tests, when the Eps value is 0.6-0.8 and the MinPts value is 5, the recognition effect is the best.

(3) Determining Whether Server-Like Devices Exist behind a NAT-Like Device. We need to consider two situations where there are server-like devices behind NAT-like devices: one is that the server-like device only serves the devices behind the NAT-like device, so the traffic data will not have corresponding connection records; the other is that the service objects of the server-like device are the whole network devices, which can be accessed by the devices in the intranet, and then the traffic data information will have corresponding connection records. Our validation analysis process is only used for the latter situation. In the process of analysis, we find 4 such connection records, among which the source IP is a host-like device in the intranet and the destination IP is the NAT-like device. Therefore, we can conclude that there is a server-like device behind the NAT-like device.

4.4. Intranet Visualization. As mentioned above, the cyberspace surveying and mapping technology ultimately draws the detection result. We present a visual result of a dataset, as shown in Figure 14. It should be noted that the visualization part is cropped. NAT1 represents a NAT-like device with no server-like device behind it, and NAT2 represents a NAT-like device with a few server-like devices behind it.

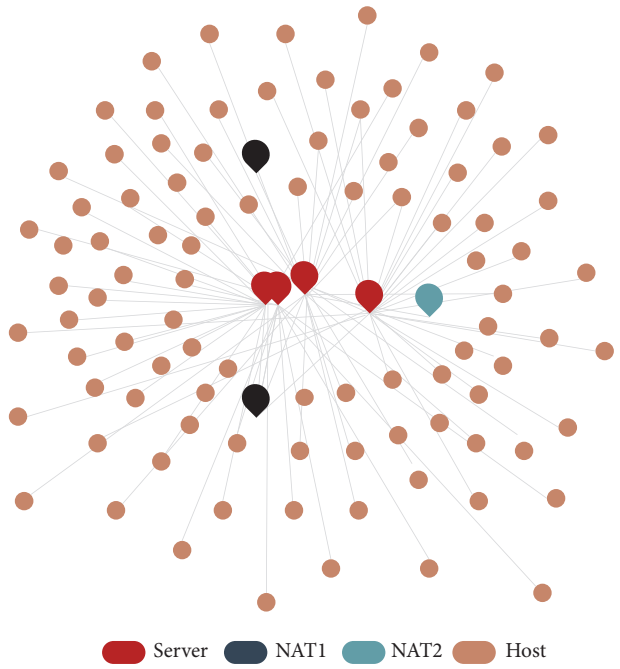


Figure 14: Visualization of the internal network.

5. Conclusion

Based on the unsupervised clustering algorithm—DBSCAN—this paper identifies the devices in the internal

network by a passive measurement. This process can be divided into two processes. One is to use the graph structure features to identify the server-like devices and terminal devices in the internal network. In this process, we use a filtering method, which can prevent the highly associated terminal devices from being misidentified as server-like devices. The second step is using traffic analysis method to divide terminal devices into NAT-like devices and host-like devices. During this process, a method of traversing the server-like devices to detect the NAT-like devices is adopted, which can make the classification result better. The framework is effective. It can identify network devices existing in the data traffic information and detect the existence of unauthorized NAT-like devices. The surveying and mapping information obtained by this framework provides important data support for improving the effectiveness and intelligence of analysis methods such as causal association, attack scene correlation, and subject-object association. In the data traffic information, only the related traffic of these three types of devices appears, and other devices such as switches and hubs do not appear in Elasticsearch. We will continue to expand our research to achieve more comprehensive cyberspace surveying and mapping.

Data Availability

The data that support the findings of this study are not publicly available due to restrictions as the data contain sensitive information about a real-world intraindustry network. Access of the dataset is restricted by the original owner. People who want to access the data should send a request to the corresponding author, Wei Sun, who will apply for permission of sharing the data from the original owner.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] R. Motamedi, R. Rejaie, and W. Willinger, "A survey of techniques for internet topology discovery," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 1044–1065, 2015.

[2] A. Dainotti, K. Benson, A. King, M. Kallitsis, and Glatz. E., "Errata for: Estimating internet address space usage through passive measurements (SIGCOMM CCR (Vol. 44, Issue 1, January, 2014)," *Acm Sigcomm Computer Communication Review*, vol. 44, no. 2, pp. 99-100, 2014.

[3] K. Levchenko, A. Dhamdhere, B. Huffaker, K. Claffy, M. Allman, and V. Paxson, "PacketLab: A universal measurement endpoint interface," in *Proceedings of the 2017 ACM Internet Measurement Conference, IMC 2017*, pp. 254–260, ACM, November 2017.

[4] W. Sun, J. Jiang, and M. Su, "A passive-measurement-guided tree network surveying and mapping model," in *Proceedings of the 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pp. 646–651, Guangzhou, China, June 2018.

[5] M. Li, Y. Sun, Y. Jiang, and Z. Tian, "Answering the min-cost quality-aware query on multi-sources in sensor-cloud systems," *Sensors*, vol. 18, no. 12, p. 4486, 2018.

[6] W. Han, Z. Tian, Z. Huang, S. Li, and Y. Jia, "Bidirectional self-adaptive resampling in internet of things big data learning," *Multimedia Tools and Applications*, 2018.

[7] Z. Wang, C. Liu, J. Qiu, Z. Tian, X. Cui, and S. Su, "Automatically traceback rdp-based targeted ransomware attacks," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 7943586, 13 pages, 2018.

[8] Z. Tian, S. Su, W. Shi, X. Du, M. Guizani, and X. Yu, "A data-driven method for future Internet route decision modeling," *Future Generation Computer Systems*, vol. 95, pp. 212–220, 2019.

[9] J. Qiu, Y. Chai, Y. Liu, Z. Gu, S. Li, and Z. Tian, "Automatic non-taxonomic relation extraction from big data in smart city," *IEEE Access*, vol. 6, pp. 74854–74864, 2018.

[10] Y. Wang, Z. Tian, H. Zhang, S. Su, and W. Shi, "A privacy preserving scheme for nearest neighbor query," *Sensors*, vol. 18, no. 8, p. 2440, 2018.

[11] Z. Tian, Y. Cui, L. An et al., "A real-time correlation of host-level events in cyber range service for smart campus," *IEEE Access*, vol. 6, pp. 35355–35364, 2018.

[12] Q. Tan, Y. Gao, J. Shi, X. Wang, B. Fang, and Z. H. Tian, "Towards a comprehensive insight into the eclipse attacks of tor hidden services," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1584–1593, 2018.

[13] J. Chen, Z. Tian, X. Cui, L. Yin, and X. Wang, "Trust architecture and reputation evaluation for internet of things," *Journal of Ambient Intelligence and Humanized Computing*, vol. 2, pp. 1–9, 2018.

[14] P. Srisuresh and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations," RFC Editor RFC2663, 1999.

[15] K. Park and H. Lee, "On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law Internets," in *Proceedings of the ACM SIGCOMM 2001- Applications, Technologies, Architectures, and Protocols for Computers Communications-*, pp. 15–26, USA, August 2001.

[16] X. Yu, Z. Tian, J. Qiu, and F. Jiang, "A data leakage prevention method based on the reduction of confidential and context terms for smart mobile devices," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 5823439, 11 pages, 2018.

[17] F. Zhao, X.-y. Luo, and F.-l. Liu, "Research on cyberspace surveying and mapping technology," *Chinese Journal of Network and Information Security*, vol. 9, no. 2, pp. 1–11, 2016.

[18] T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 2, pp. 93–108, 2005.

[19] R. Fink, "A statistical approach to remote physical device fingerprinting," in *Proceedings of the Military Communications Conference, MILCOM 2007*, USA, October 2007.

[20] J.-W. Wang, L.-L. Rong, and T.-Z. Guo, "A new measure method of network node importance based on local characteristics," *Journal of Dalian University of Technology*, vol. 50, no. 5, pp. 822–826, 2010.

[21] M. Kitsak, L. K. Gallos, S. Havlin et al., "Identification of influential spreaders in complex networks," *Nature Physics*, vol. 6, no. 11, pp. 888–893, 2010.

[22] Y. Gokcen, V. A. Foroushani, and A. N. Z. Heywood, "Can we identify NAT behavior by analyzing traffic flows?" in *Proceedings of the 2014 IEEE Computer Society's Security and Privacy Workshops, SPW 2014*, pp. 132–139, USA, May 2014.

[23] A. Dupuy, S. Sengupta, O. Wolfson, and Y. Yemini, "NETMATE: a network management environment," *IEEE Network*, vol. 5, no. 2, pp. 35–40, 1991.

[24] Quinlan J R. C4.5: programs for machine learning, 1992.

[25] J.-H. Xue and D. M. Titterington, "Comment on "on discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes"," *Neural Processing Letters*, vol. 28, no. 3, pp. 169–187, 2008.

[26] R. Li, H. Zhu, Y. Xin et al., "Remote NAT detect algorithm based on support vector machine," in *Proceedings of the International Conference on Information Engineering & Computer Science*, IEEE, 2009.