Hindawi
Security and Communication Networks
Volume 2019, Article ID 8565397, 12 pages
https://doi.org/10.1155/2019/8565397

Research Article

Evaluating the Impact of Name Resolution Dependence on the DNS

Haiyan Xu, Zhaoxin Zhang, Jianen Yan, and Xin Ma

School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China

Correspondence should be addressed to Jianen Yan; yanjianen_hit@163.com

Received 21 March 2019; Revised 12 July 2019; Accepted 20 August 2019; Published 9 September 2019

Guest Editor: Sebastian Schrittwieser

In the process of resolving domain names to IP addresses, there exist complex dependence relationships between domains and name servers. This paper studies the impact of the resolution dependence on the DNS through constructing a domain name resolution network based on large-scale actual data. The core nodes of the resolution network are mined from different perspectives by means of four methods. Then, both core attacks and random attacks on the network are simulated for further vulnerability analysis. The experimental results show that when the top 1% of the core nodes in the network are attacked, 46.19% of the domain names become unresolved, and the load of the residual network increases by nearly 195%, while only 0.01% of domain names fail to be resolved and the load increases with 18% in the same attack scale of the random mode. For these key nodes, we need to take effective security measures to prevent them from being attacked. The simulation experiment also proves that the resolution network is a scale-free network, which exhibits robustness against random failure and vulnerability against intentional attacks. These findings provide new references for the configuration of the DNS.

1. Introduction

Domain name system (DNS) is one of the most important infrastructures on the Internet. When people receive services of the Internet, they usually connect to the remote host by entering a hostname instead of the IP address that is hard to be remembered by users. This design simplifies users operation but requires a powerful and distributed DNS to provide the service of mapping domain names to IP addresses. The mapping is transparent to the user, and the DNS provides the ability to automatically convert. Therefore, the security and reliability of the DNS are vital to the Internet. If the DNS has problems, the Internet applications based on it may be impossible to provide normal services for users, which may lead to significant economic losses.

As one of the largest distributed systems of the world, the DNS is unmatched in its efficiency and popularity. In order to handle the scale problem, the DNS deploys a large number of name servers organized in a hierarchical structure and distributed throughout the world, as shown in Figure 1. In this hierarchy, there are three types of name servers: root servers, top-level servers, and authoritative name servers. Root servers and top-level servers are managed by professional Internet organizations and academic institutions, so they are more stable and safer than the authoritative name servers that store the name mapping information of their own domain. The management style of the authoritative name servers, relying on the organizations themselves or entrusted to the Internet service providers, is relatively loose, which is a weak link in the DNS. There have been some attacks on the root or top-level domain servers, whose security is also the object of concerns of many researchers [1–3]. However, we may ignore the security of many authoritative servers below the top level. Whether some important authoritative servers will be attacked and a large number of domain names' resolution failures will occur is a question we want to verify from a macroperspective.

The DNS manages domains and regions through authorization and basic rules. The authoritative name servers are generally placed in their own management domain. Most of the administrators deploy more authoritative name servers to increase performance and reliability of name

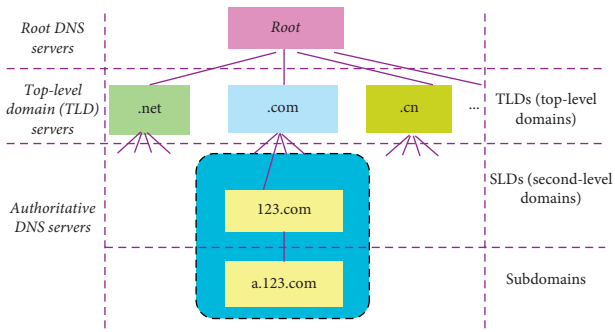

Figure 1: Hierarchical structure of DNS.

resolution. According to the DNS protocol specification [4, 5], servers can be distributed in different domain regions in order to improve the reliability of resolution; however, there is no mechanism to limit the dependencies between domains, which may result in complex name dependence. In this article, we mainly discuss the safety of the authoritative name servers (hereafter referred to as name servers).

We explored the resolving process of 1 million domain names, and the results indicate that 86.14% of the names have interdomain dependence, that is, the resolution of a domain name involves servers located in different domains. Then, in order to study the actual impact of the resolution dependence on the DNS, this paper focuses on the following three aspects:

Firstly, the resolving graph for each domain name is constructed. Thus, for each domain name, especially the one with complex interdomain dependence, its complex relationship resolution dependence is expressed through the graph.

Secondly, the resolving graphs of each domain name are combined to form a global domain name resolution network, which is a complex network of relationships between domain names and name servers. The overall characteristics of the network are also analyzed, reflecting that it is an extremely heterogeneous network, which exhibits robustness against random failures and vulnerabilities against intentional attacks. In addition, the characteristics analysis method of complex network is utilized to mine the key nodes in the network and a metrics of node load is also derived to quantify the node importance in the network.

Thirdly, the impact of critical nodes on the DNS is studied by simulating attacks on the domain name resolution network. Two strategies of the random and core attacks are simulated by node removal method to study how the name dependence impact the DNS, and then the working situation of the whole network is quantified after some name servers have failed. The experimental results show that random attacks on name servers have little impact on DNS as a whole, while attacks on a small number of core nodes in the resolution network have a great impact on DNS.

The paper is organized as follows: Section 2 introduces the concept of domain name resolution dependence, data detection, and the resolving graph of domain names. In Section 3, A global resolution network is constructed and its core nodes mining is introduced. In Section 4, an analysis method is proposed to evaluate the impact of resolution dependence on the DNS. Section 5 gives an overview of the related work in this area. Section 6 concludes our analysis.

2. Domain Name Resolution Dependence

In this section, the definition of domain name resolution dependence, the collection of actual resolution data, the construction of resolving graph for each domain name are introduced.

2.1. Definition of Domain Name Resolution Dependence. *Definition 1.* A domain name u depends on a domain name v, if and only if the resolution results of domain name v impact on the results of domain name *infrastructures*.

The existence of domain name resolution dependence is mainly due to the following three reasons:

(1) Dependent on parent domain: since the resolving process is top-down, a domain name always relies on its parent domain. If without considering the cache, the authoritative data of a domain are always returned by its parent domain. A parent domain may contain more than one subdomain.

(2) Dependent on name servers: the mapping information from hostnames to IP addresses is stored in their name servers. If a resolver wants to resolve a name, DNS queries must be initiated to their name servers.

(3) Dependent on aliases: if a domain name has an alias, the alias must be resolved. Therefore, the resolution of a domain name is also dependent on its alias.

Due to the reasons above, domain dependence is partitioned into the following three types:

(1) Intradomain dependence: if v, an authoritative name server's DNS name of domain u, is administered by domain u itself, then the dependence relationship between u and v is the intradomain dependence, such as the relationship between domain *baidu.com* and its name server *dns.baidu.com*. When the DNS resolver receives this type of resource record, it will use the IP of the name server in the additional part of the resource record to respond for the next query.

(2) Interdomain dependence: if v, an authoritative name server's DNS name of domain u, is not administered by domain u, then the dependence relationship between u and v is interdomain dependence, such as the relationship between the domain *edu.cn* and its name server *cuhk.edu.hk*. When the DNS resolver receives this type of resource record, it will requery the address of the name server and then use the address to make the next query. Even though the additional part of the response packet has an address for the name server, the DNS resolver will ignore it and still requery the address of the name server.

(3) Alias dependence: if a domain name v is the alias of a domain name u, then u is dependent on v, such as the relationship between http://www.baidu.com and www.a.shifen.com. According to the RFC standard [4], the DNS resolver will restart the query for the address of it once the alias record is received.

Because of the existence of interdomain dependence, some name servers may serve for multiple domain names, which can form very complicated dependence relationships among many domain names. This is verified by detecting the resolution data of a large collection of domain names.

2.2. Dataset. The data for this paper are derived from the ranking data of Top 1 million sites from http://www.alexa.com [6], a famous network navigation service provider. These sites are the most popular sites, and they are typical representatives. For each of these domain names, its recursive DNS request messages are constructed and then sent to the DNS servers. From the returned response messages, we recorded the name and address of name servers. If the relationship between a domain name and its name server belongs to the type of interdomain dependence, the recursive detection of its name server will be conducted.

We use an example of a domain name www.edu.cn to explain the method of detecting domain name resolution dependence. Resolving this domain name will be iterated from the root, top-level and *edu.cn* domains [4]. When the domain *edu.cn* is queried, the top-level server returns five authoritative servers, as shown in Figure 2. If it is a server in its own domain, such as *dns.edu.cn*, there are additional records in the DNS response package giving the name server's IP address, then the DNS resolver will issue a DNS query to this address; if the authoritative server is not in its own domain, such as *ns2.cuhk.hk,* there is no IP address of an extraterritorial server. In this case, the server's address needs to be queried iteratively from the root, top-level, and *cuhk.hk* domains. This leads to complex dependencies.

This paper assumes that the root servers and TLD servers were in a normal state, so the resolution dependence is only related to the name servers below the TLD.

In the detection process, this rule is followed: stop detection when the name servers are self-dependent. For example, if the server of *A.net* is *ns1.A.net*, which does not rely on other domains, the detection process will stop.

2.3. Data Statistics for Domain Name Resolution Dependence Measurement. After the resolution dependence detection of 1 million domain names, we find that about 86.14% of the names have interdomain dependence. Then, we further acquire statistics on these data. First, we acquire statistics on the number of name servers involved in each domain name resolution (not including the root name servers and the TLD name servers), and the cumulative distribution of it is plotted in Figure 3. Specifically, about 59.43% of the domain names depend on 2 servers, and

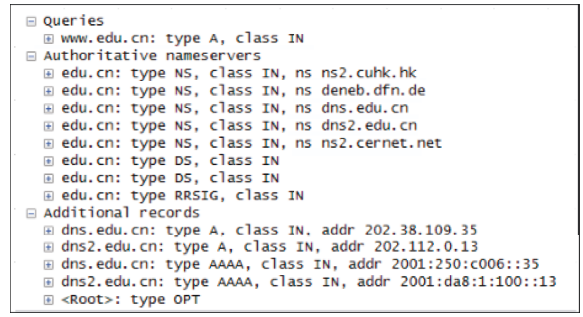

Figure 2: DNS response packets as raw data returned by a top-level server.

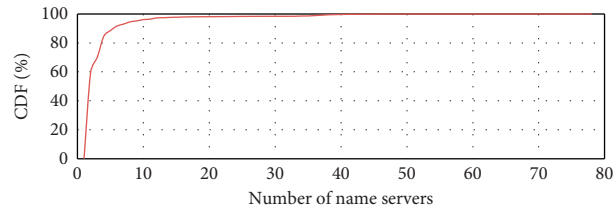

Figure 3: The cumulative distribution of the number of name servers involved in each domain name resolution.

15.38% of them depend on more than 5 servers. The number of servers is in the interval [1, 78]. From the overall distribution, about a quarter of these domains have complex dependencies. The configuration of these domains deserves high attention.

On the other side, some name servers have high degree value. They can provide resolution services for hundreds of domain names, and most of them are the DNS service providers, as shown in Table 1. Once these servers are compromised or down, it may affect a large area of domain names.

As mentioned above, we have demonstrated the overall resolution dependence of the 1 million domain names. We found that some domains have an extremely complex dependency relationship and also discovered that some name servers provide resolution services for hundreds of domain names. These domains and name servers raise our great concerns.

2.4. Construction of the Resolving Graph for Each Domain Name. According to the results of name resolution detection, we construct the resolving graph for each domain name, which reflects the relationship between the domains and their name servers. In this graph, a domain or a name server is considered as a node and the resolution dependence between them is taken as an edge.

Definition 2. Domain name resolving graph can be defined as two tuples:

$$G_d = (V_d, E_d), \tag{1}$$

where V_d is the set of nodes, and any node $v \in V_d$ in the graph represents a domain name or a name server. E_d is the edge

Table 1: Name servers of high dependence degree.

No.	Name servers	Number of domains
1	f1g1ns2.dnspod.net	376
2	f1g1ns1.dnspod.net	376
3	dns3.registrar-servers.com	187
4	dns2.registrar-servers.com	187
5	dns1.registrar-servers.com	187
6	dns4.registrar-servers.com	183
7	dns5.registrar-servers.com	181
8	ns2.bluehost.com	171
9	ns1.bluehost.com	171
10	ns0.dnsmadeeasy.com	170
11	ns1.dnsmadeeasy.com	168
12	ns2.dnsmadeeasy.com	161
13	ns11.dnsmadeeasy.com	161
14	ns3.dnsmadeeasy.com	159
15	ns10.dnsmadeeasy.com	159
16	ns12.dnsmadeeasy.com	158
17	ns2.rackspace.com	152
18	ns13.dnsmadeeasy.com	152
19	ns.rackspace.com	150
20	ns4.dnsmadeeasy.com	148
21	ns2.dreamhost.com	144
22	ns1.dreamhost.com	144
23	dns2.stabletransit.com	139
24	dns1.stabletransit.com	139
25	ns3.dreamhost.com	133
26	ns1.dns.ne.jp	132
27	ns2.dns.ne.jp	130
28	ns14.dnsmadeeasy.com	130
29	dns4.name-services.com	130
30	dns3.name-services.com	130

Number of domains represents the number of domains that depend on this name server to resolve.

set, and any edge $(u, v) \in E_d$ representing that the resolution of domain u depends on the resolution of domain v or relies on a name server.

Figure 4 shows an example of a resolving graph, from which we can see many name servers are involved for mapping a domain name to IP address. The resolution relies on the root server, the TLDs, the *edu*.cn, and its own name servers. It can be viewed as transitive trust behavior from the root servers to the name servers. In this paper, assuming that the root and TLDs are always in normal condition, we only study the impact of authoritative name servers on the DNS. The reason is that the root and the TLDs are managed by professional Internet organizations or academic institutions, while the relatively loose management of the name servers is a weak link. It is clear to see that the resolution of this name "www.edu.cn" relates to the parent domain "*edu.cn*", and it has five name servers. Three of them belong to the type of interdomain dependence. In order to obtain the IP addresses of these three servers, it needs to restart the resolving of the new domains.

When a domain name with interdomain dependence is resolved, many name servers may be involved. Further analysis of the graph can be made to explore the impact of domain name resolution on the existing domain name system at the macrolevel.

3. Name Resolution Network and Its Core Nodes

Since the above resolution dependence data in Section 2.4 is more fragmented and complex, it is not conducive to our macroscopic research on the DNS. Moreover, because of the influence of interdomain dependence, the correlation between different domains is particularly complex. A name server may be used to provide services for different domains. Therefore, we connect the dependence graphs of 400,000 domain names to form a global directed name resolution network, represented as *Net1*. For the sake of clarity, Figure 5 displays an example of a smaller resolution network that contains 100 domain names.

3.1. Characteristics of the Name Resolution Network. After *Net1* is constructed, we calculate some characteristics of it for an overall understanding. Details are shown in Table 2, from which we can see that there are 1,135,448 nodes and 1,811,565 edges in this resolution network. Specifically, a few nodes have a large number of connections, but most nodes have few. This is supported by Figure 6, which is the in-degree distribution of *Net1*. A description of the in-degree is presented in Section 3.2.1. The in-degree distribution reflects that *Net1* is an extremely heterogeneous network, namely, the scale-free network. It exhibits robustness against random failures and vulnerabilities against intentional attacks [7]. In addition, the proportion of the lonely domains (i.e., the domain names only with intradomain dependence) is 15.75% of the 40,000 names, which indicates that the resolving process of three-quarters of them is related to the other domains.

3.2. Identifying the Core Nodes of the Name Resolution Network. We use some classic node centrality measures and a method we proposed to mine the core nodes of the network. The purpose is to identify the important nodes in the network, that is, to discover the name servers providing services for a large number of domain names. Once such a name server becomes the target of DDoS (Distributed Denial of Service), it will inevitably result in the resolution failure of many sites.

3.2.1. Classic Node Centrality Measures. Node centrality measures are a classic tool in complex network analysis to determine the important nodes, and some of them are considered in this paper, such as in-degree, closeness, and Eigenvector centrality. The following is a brief review of these complex network properties, with more background and details in [8].

The in-degree of a node u in a directed network is the number of other nodes that point to u. The closeness of u reflects the proximity between u and other nodes in the network. If the shortest distance between u and the other nodes in the graph is very small, then we think that the closeness of u is high. This measure is more geometrically consistent with the concept of centrality than in-degree centrality. Because if the average shortest distance between

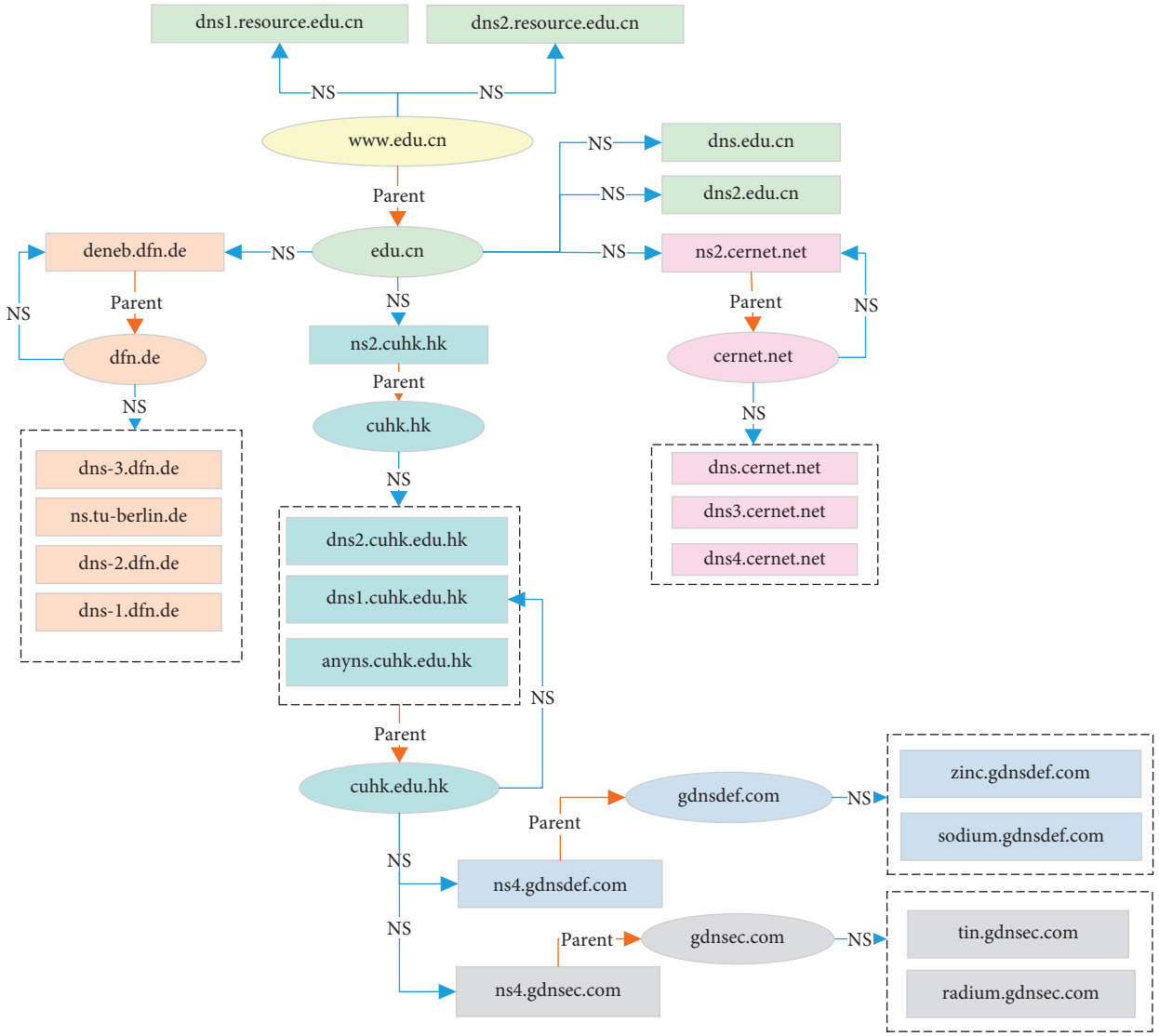

FIGURE 4: Resolving graph of a domain name, where the edge label "parent" indicates it is dependent on the parent domains, the label "NS" indicates it is dependent on the name servers. The resolution of this domain name involves 6 different domains and their name servers, which is symbolized by different colors in the graph. In addition, alias dependence does not appear in this figure. If a domain name has a "cname," alias should be marked on the edge between the domain name and the cname.

node u and other nodes is the smallest, then u is geometrically located at the center of the graph. The Eigenvector centrality of u has a relative index value, which is based on the following principle—contribution of connecting high-scoring nodes to u is more than that of connecting low-scoring nodes. It is the first of the centrality measures that considered the transitive importance of a node in a graph [9].

These classical node centrality measures can identify the important nodes of complex networks from different perspectives. The measures applicable to our resolution network and their specific experimental results are presented in the subsequent analysis in Section 4.4.

3.2.2. Node Load Centrality Measures. Based on the resolution network and combining the actual name resolving process, an algorithm is proposed to quantify the load of each node. For a domain name u, the load of u reflects the sum of the dependence of all other nodes on u in their domain name resolution. The algorithm starts with domain name nodes and traverses each other node in the resolving graph in turn and calculates its value of load. The initial value of the load for all domain name nodes is set as 1 and that for the remaining nodes is set as 0. The calculation of the load is an iterative cumulative process.

As shown in the construction of the resolving graph in Section 2.4, the difference between the types of nodes leads to diverse relations between adjacent nodes, such as relations between (1) child domain and parent domain, (2) domain and name server, and (3) domain name and its alias. Therefore, the quantitative rules for the three categories are defined as follows.

(1) Relations between Child Domain and Parent Domain. As shown in Figure 4, a domain name's resolution is always

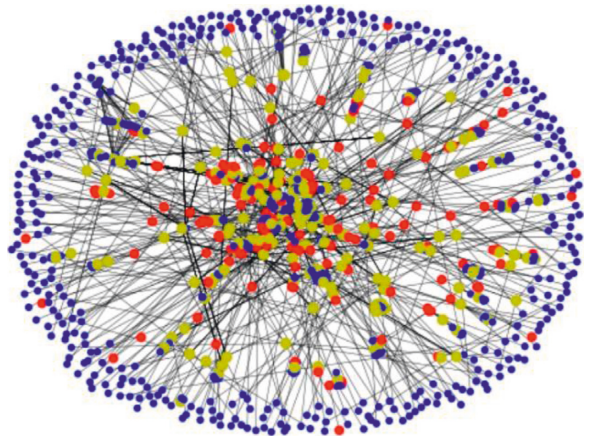

Figure 5: A global dependency graph that contains 100 domain names. The nodes of the domain name and its alias are colored by red, the domain nodes are marked yellow, and the name server nodes are marked blue.

Table 2: Characteristic parameters of *Net1*.

Metric	Value
Number of nodes	1,135,448
Number of edges	1,811,565
Maximum of degree	6,962
Nodes ratio of the degree 1	35.19%
Average value of degree	3.1909
Average value of clustering	0.00006
Proportion of lonely domains	15.75%

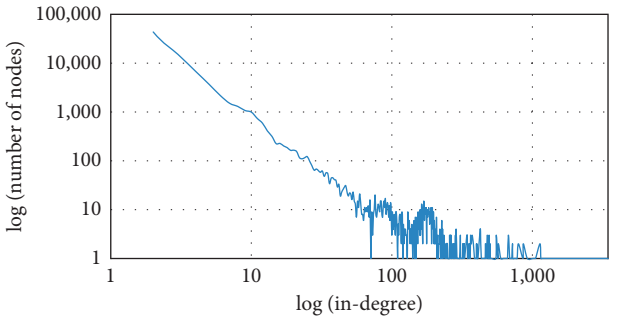

Figure 6: In-degree distribution of *Net1*.

dependent on its parent domain. When a directed edge points from a node to its parent domain node, the load of the parent domain node accumulates the load of its child node. For a parent domain pu, let u_i denote a child domain that depends on pu, and Load_u denote the load of u. Then, the load of parent domain pu in the resolution network is

$$\text{Load}_{\text{pu}} = \sum_{i \in N} \text{Load}_{u_i}, \tag{2}$$

where N is the number of child domains that depend on the parent domain pu.

(2) Relations between Domain and Name Server. A domain name can deploy multiple authoritative name servers. As long as one name server provides normal service, the domain name can be successfully resolved. In this paper, assuming that each name server provides service for its domain with the same probability, the load of a name server node is the load of the domain it is deployed divided by the number of all name servers in that domain. If this name server is set to an authoritative server by more than one domain, we accumulate these values as the load of this name server. Therefore, the load of each name server node in the resolution network is

$$\text{Load}_{\text{ns}} = \sum_{i \in N_{\text{domains}}} \left(\frac{\text{Load}_{\text{un}_i}}{N_i} \right), \tag{3}$$

where ns is a name server node, un_i indicates a domain where *ns* is deployed, N_i is the number of all name servers in domain un_i, and N_{domains} is the number of all domains where ns is deployed.

(3) Relations between Domain Name and its Alias. If a domain name has a resource record of CNAME type, that is, an alias, the alias will inherit its load value. Because at this point the resolution of the domain name has shifted to the resolution of its alias [5]. Generally, an alias corresponds to only one domain name, so there is no cumulative calculation. The load of alias node in the resolution network is

$$\text{Load}_{\text{alias}} = \text{Load}_u, \tag{4}$$

where alias denotes a domain name u's CNAME.

Following the above rules, traverse the nodes of the network and calculate the load for each node. Nodes with larger load are considered to be the center of the network. The results of quantifying the nodes in Figure 4 according to the above method are shown in Figure 7, which only shows a single domain name's dependency relationship, and the nodes in the graph are marked with load values. Since a name server can be used to provide services for different domains, the load of the name server nodes and their parent domain nodes in the resolution network may be accumulated multiple times. Consequently, by comparing the load values, some important nodes in the network can be identified.

Figure 8 presents the top 20 nodes in the network using the centrality measures listed above. The more effective measures in mining the core nodes of the network are discussed in Section 4.4.

4. Evaluating the Impact of Resolution Dependence on the DNS

The global resolution graph is proved to be a scale-free network, which has strong fault tolerance, but its antiattack ability is rather poor for the selective attack based on the key nodes [7]. The presence of the nodes of high-connectivity greatly weakens the robustness of the network. A malicious attacker only needs to select a few nodes of the network to make the network instantly paralyzed. In the DNS, due to the interdomain dependence, the failure of an important node may cause multiple domain names to be unresolved or to

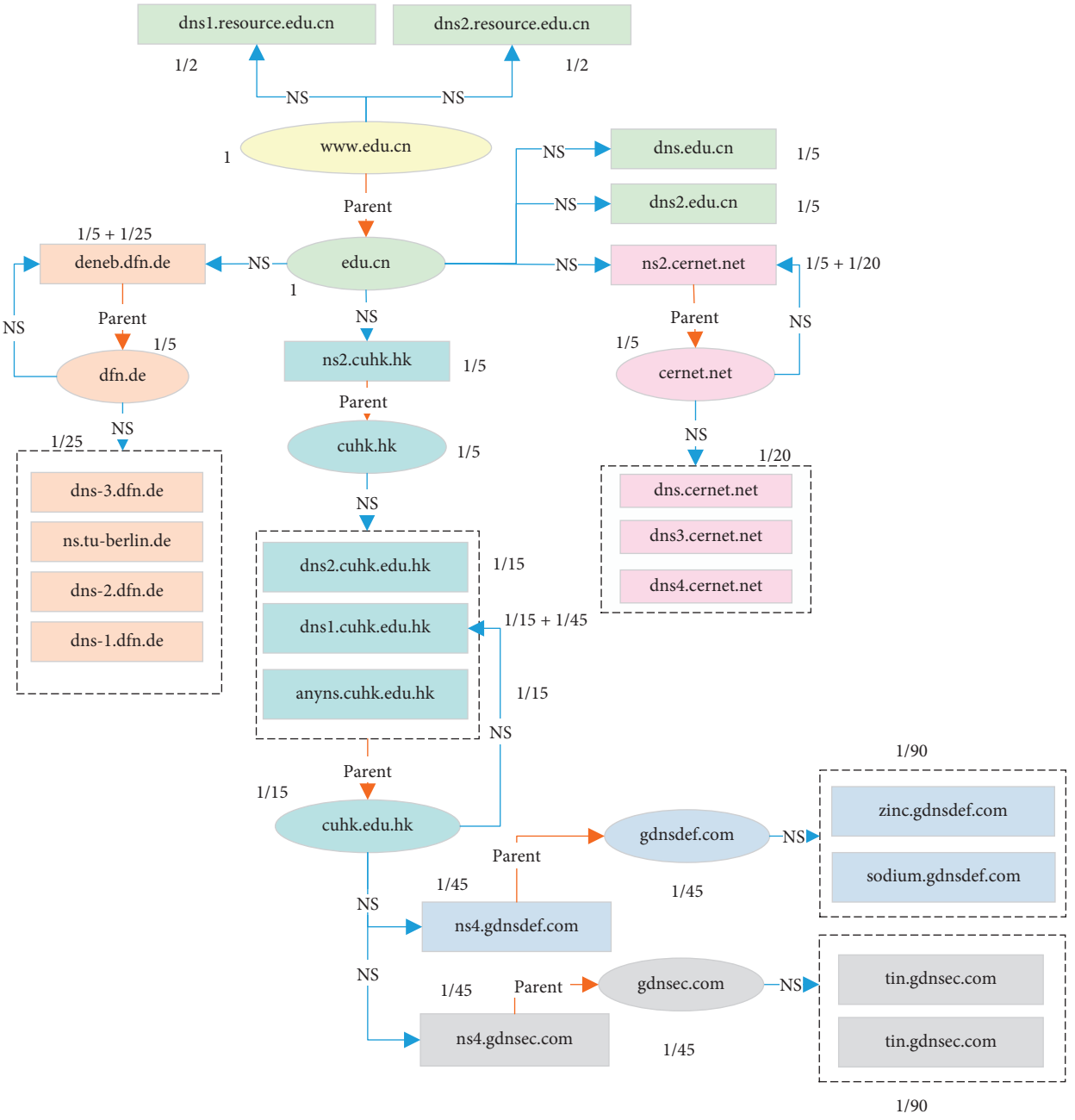

Figure 7: Resolving graph with load value corresponding to Figure 4. The fractional number next to each node represents its load value.

transfer the node's traffic load to its equivalent other servers, which is a cascading failure response in this network. In this paper, we choose to simulate attacks by removing some nodes in the resolution network to evaluate the impact of resolution dependence on the DNS.

4.1. Simulation of Attacks on Name Resolution Networks. On the basis of the name resolution network *Net1* constructed in Section 3, we simulate the DNS attacks by removing some nodes from the network, which is relatively simple compared to simulating the DNS of real resolving process of many domain names. However, it helps us to verify whether the failure of some core name servers will affect the resolution of a large number of domain names at the macrolevel.

In light of different ways of removing nodes, we classify the attacks into two categories: random attacks and core attacks. The random attacks are to select random nodes to remove from *Net1*; the core attacks is to remove the core nodes according to the metrics obtained from the node centrality measures, including the in-degree, closeness, eigenvector, and node load centrality.

After removing the nodes in the network according to certain rules, we evaluate the network status before and after attacks. One of the intuitive and effective indicators used for the evaluation is the name resolution failure rate, which is detailed in Section 4.2. Another indicator is the delay of DNS

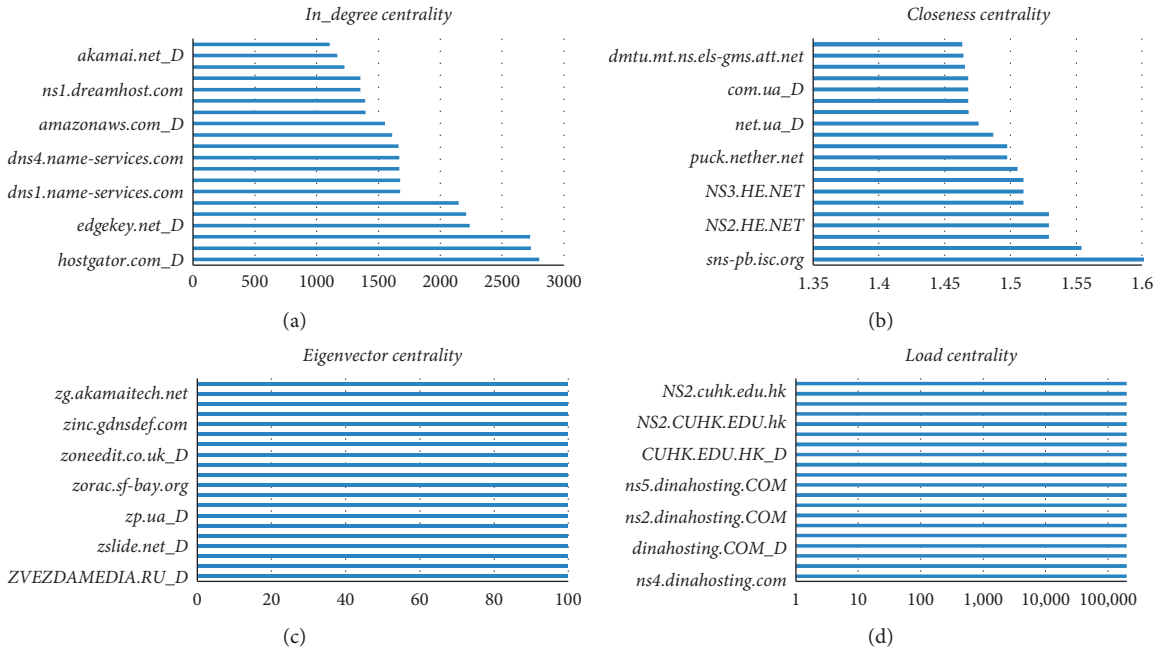

Figure 8: The top 20 nodes of the domain name resolution network, which is mined by measure of in-degree, closeness, eigenvector, and node load centrality.

name resolution, however, which is not adopted in our evaluations. Because *Net1* is a global resolution dependency graph of 400,000 domain names and the actual domain name resolving process does not been emulated, this indicator cannot be obtained from our simulation. In addition, the failure of some server nodes not only causes the related domain name to become unresolvable but also transfers the resolution workload to their equivalent other name servers, which is the cascading failure in this network. Furthermore, we propose an evaluation method of load transfer based on the idea of cascading failure of complex networks, which is described in Section 4.3.

4.2. Resolution Failure Rate Assessment. As the role of the DNS is to provide users with resolution services, the resolution failure rate can be the most intuitive measure of service quality. Consequently, we propose a method to compute the resolution failure rate under attacks from the perspective of static influence. More specifically, after having removed some nodes from the resolution network, we calculate the number of the domain names which become unresolvable and then compute the resolution failure rate. The quantization method is shown in the following formula:

$$E1_{\text{Destruction\_Set}} = \frac{N_{\text{Failure\_Name}}}{N_{\text{All}}}, \tag{5}$$

where Destruction_Set is the set of nodes to be removed from the network, $E1_{\text{Destruction\_Set}}$ is the resolution failure rate caused by the Destruction_Set removed from the network, $N_{\text{Failure\_Name}}$ is the number of domain names that fail to be resolved due to the attack, and N_{All} is the number of all the domain names in the network to be assessed.

4.3. Load Transfer Assessment. In many real-world networks, one or several nodes' failures can cause other nodes to fail through the coupling relationship between nodes, which is called cascading failure. In the name resolution network, there are many combinations of name servers served for resolving a name; that is, the failure of some name servers may not affect the normal resolution of the domain name, but it can aggravate the load of the remaining name servers of the domain name. Accordingly, it is necessary to study the dynamic influence assessment for cascading failure. Here, the changes of load of the overall network before and after attacks are monitored. The calculation of the load has been described in detail in Section 3.2.2. The dynamic impact based on load transfer is as follows:

(1) For each domain name, calculate the average load of all name server nodes that each domain name depends on

$$\text{Load}_{\text{domain}} = \frac{1}{N_{\text{server}}} \sum_{i \in N_{\text{server}}} \text{Load}_{\text{server}_i}, \tag{6}$$

where $\text{Load}_{\text{domain}}$ represents the average load of a domain name, N_{server} represents the number of servers owned by a domain name, and $\text{Load}_{\text{server}_i}$ represents the load of a name server i.

Then, compute the average load for all domain names of the network. The load of the network is shown in the following formula:

$$\text{Load}_{\text{net}} = \frac{1}{N_{\text{domain}}} \sum_{j \in N_{\text{domain}}} \text{Load}_{\text{domain}_j}, \quad (7)$$

where Load_{net} represents the average load of a resolution network, N_{domain} represents the number of domain names in the network, and $\text{Load}_{\text{domian}_j}$ represents the load of a domain name j.

(2) Remove some nodes based on rules, so some domain names become unresolvable and then recalculate the average load in the residual network.

(3) For each attack, compute the change rate of load for the overall network before and after attacks:

$$E2_{\text{Destruction\_Set}} = \frac{\text{Load}_{\text{residual\_net}}}{\text{Load}_{\text{net}}}, \quad (8)$$

where $E2_{\text{Destruction\_Set}}$ is the change rate of load caused by the Destruction_Set removed from the network, Load_{net} represents the load of a resolution network before attacks, and $\text{Load}_{\text{residual\_net}}$ represents the load of a resolution network after attacks.

Thus, we get the specific value of the load change, which is used to represent the degree of cascading effect.

4.4. Experiment and Results. Our experiment is based on the domain name resolution network *Net1* established in Section 3. The DNS attacks are emulated by removing certain nodes from the network. Two strategies of the random and core attacks are simulated to study how the name dependence impacts the DNS. The proportion of nodes in the network being attacked is set to 1%, 5%, 10%, 15%, 20%, etc. The statistical results and the analysis of them are shown below.

4.4.1. Analysis of the Resolution Failure Rate Assessment. The statistical results of the resolution failure rate assessment are displayed in Figure 9, which shows the resolution failure rate of five attacks modes in certain attack proportions. The ordinate represents the resolution failure rate. The abscissa shows the proportion of nodes in the network being attacked. The following conclusions can be drawn from the results:

(1) In the random mode, only 0.01% of domain names fail to be resolved when the attack scale is 1%, and 3.31% fail to be resolved when the scale is 20%. The failure rate is significantly lower than that of other core attacks, indicating that most domain names have multiple resolution paths. Therefore, random small-scale failure does not have a major impact on the DNS.

(2) In the four centrality measures of the core attacks, the effect of in-degree is the best. When the attack scale is 1% by removing the nodes with high in-degree value, 46.19% of the domain names become unresolved. This shows that if a small number of these core nodes are attacked, there will remain a significant impact on the DNS. In addition, the centrality measure of node load proposed in our paper can also better identify the important nodes in the network when the attack scale is above 10%, but in the smaller scale node attacks, it is not as effective as in-degree. Moreover, the closeness and eigenvector centrality are less effective in mining the core nodes of the network than the previous two types.

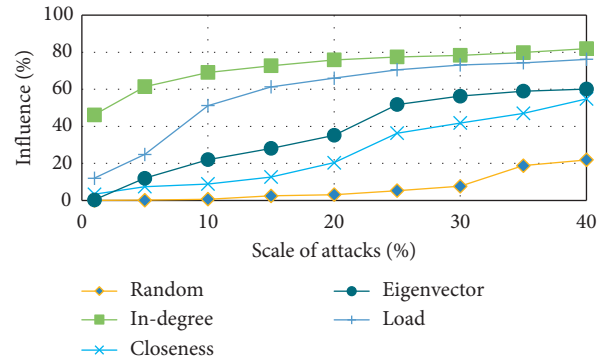

Figure 9: Statistical results of the resolution rate assessment in *Net1*.

4.4.2. Analysis of the Load Transfer Assessment. Figure 10 shows the statistical results of the load transfer assessment. The ordinate indicates the change rate of load after attacks, and the abscissa shows the proportion of nodes in the network being attacked. The following conclusions can be drawn from the analysis of the results:

(1) The results show that when the top 1% nodes (with metric of in-degree centrality) are removed, the load of the network increases by nearly 195%, while the load of the random mode increases by only 18% under the same attack scale. It can be seen that attacks on core nodes will not only cause a large number of domain names to be unresolved but also produce excessive load to other name servers. For these key nodes, effective security measures should be taken to protect them; on the other hand, the workload can be appropriately dispersed to other nodes to avoid the single point failure problems.

(2) In the simulation of core attacks, the effect of in-degree centrality is the best, followed by the load centrality. This is similar to the test results of the resolution failure rate assessment. So, these two measures can be used to identify the core of the resolution network. However, the closeness centrality does not work well in mining core nodes of the name resolution network.

4.4.3. Further Validation. Since all these conclusions need further validation, another set of 200,000 domain names were selected from the 1 million domain names surveyed to form a resolution network, represented as *Net2*. The domain name sets of net1 and net2 are independent, respectively, and have no intersection. The comparison of characteristic

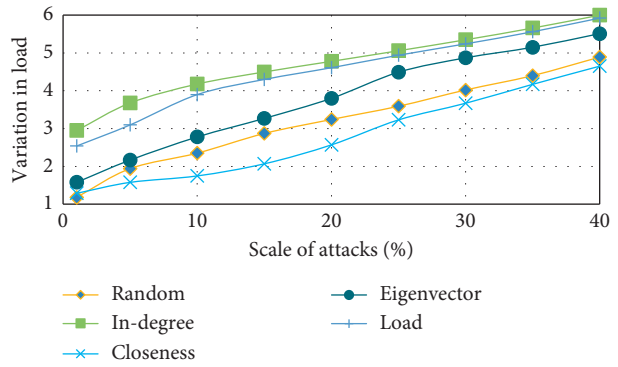

Figure 10: Statistical results of the load transfer assessment in *Net1*.

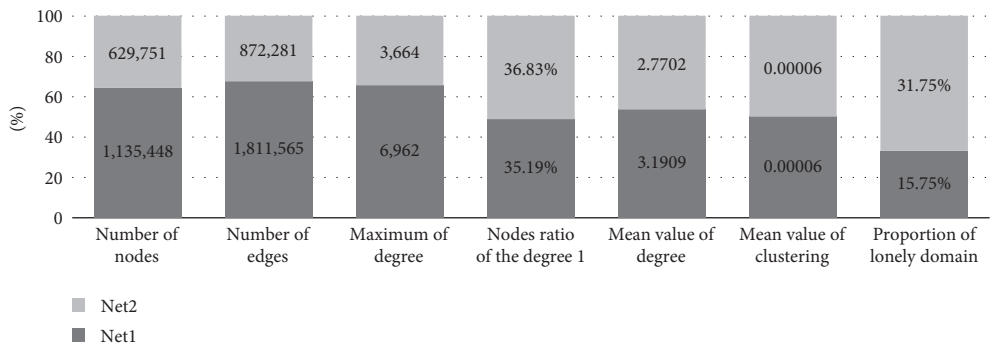

Figure 11: Comparison of characteristic parameters of *Net1* and *Net2*.

parameters of *Net1* and *Net2* is shown in Figure 11, and the in-degree distribution of *Net2* is displayed in Figure 12, showing a few nodes have a large number of connections. This reflects that *Net2* is also a scale-free network, which exhibits robustness against random failures and vulnerabilities against intentional attacks. We use the in-degree centrality and the load centrality measures that present good performance to mine the core nodes, simulate random attacks and core attacks, and utilize the resolution failure rate to evaluate the impact on the network.

The specific results are shown in Figures 13 and 14, from which we can see that the experimental results are consistent with the conclusions of *Net1*. The detailed results of resolution failure rate assessment are as follows: (1) In the random mode, only 0.01% of domain names fail to be resolved when the attack scale is 1% and 4.44% fail to be resolved when the scale is 20%. (2) In the core mode, when the attack scale is 1% by removing the nodes with high in-degree value, 62.93% of the domain names become unresolved and 8 5.38% fail to be resolved when the scale is 20%. This shows that if core nodes are attacked, there will remain a significant impact on the network, but random small-scale failure does not have a significant impact. The results of the load transfer assessment in *Net2* also verified the conclusion.

5. Related Work

The security and availability of the DNS have become a common concern. The current studies in this field mainly

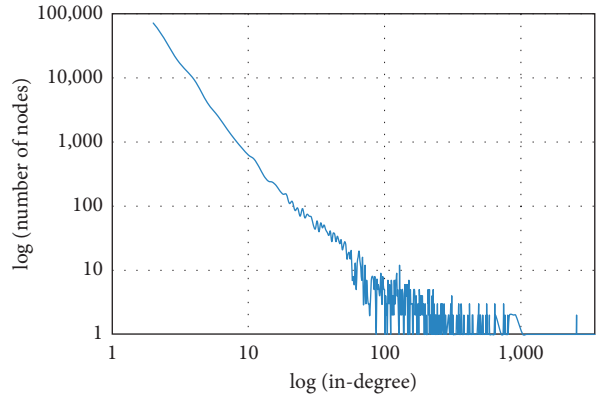

Figure 12: In-degree distribution of *Net2*.

focused on data flow anomaly detection [10–12], DNS amplification attacks [13, 14], DNS servers availability measurement [15, 16], cache poisoning detection [18], DNSSEC security protocols [17–20], and Botnet tracking combined with DNS [21, 22]. Ramasubramanian and Sirer [23] first proposed the concept of DNS dependence, which can lead to a highly insecure naming system. Casey Deccio [24, 25] proposed a DNS dependence model, which featured a probabilistic method to quantify the influence of the domain name in the trusted computing base and used a numerical value from 0 to 1 to quantify the influence degree. Fujiwara et al. [26] took the lead to measure DNS traffic increases due to interdomain dependence, with a result of 60% of DNS traffic involved out-bailiwick name servers,

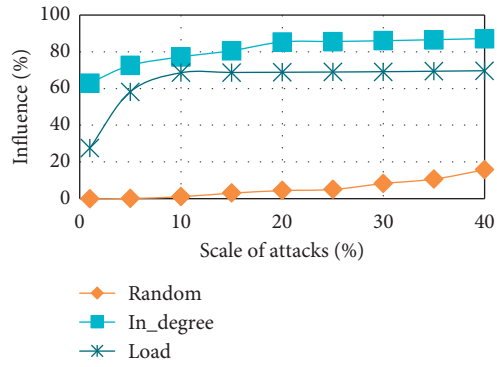

Figure 13: Statistical results of the resolution failure rate assessment in *Net2*.

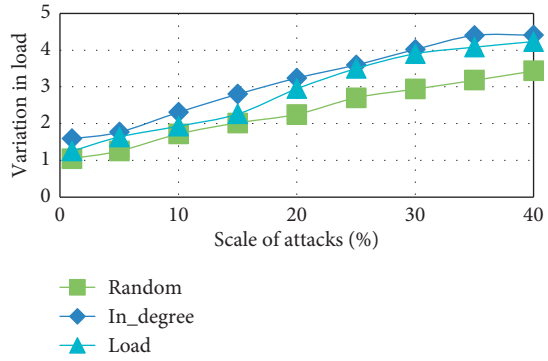

Figure 14: Statistical results of the load transfer assessment in *Net2*.

which increased the resolution delay by about 47% in the actual measurement of the DNS traffic and delay. In summary, DNS interdependent behavior has drawn the attention of researchers in recent years. However, less research has focused on the extent to which name dependence affects DNS.

6. Conclusion

In this paper, we have analyzed the actual impact of the resolution dependence on the DNS, investigated the dependence relationship of 1 million domain names, and found that 86.14% of the domain names are dependent on name servers which are not in their own authorization domain (we do not consider the case of top-level domain and the root domain here). Then, a resolution dependency graph of each domain name has been constructed based on the data we explored. Due to the influence of interdomain dependence, there may be correlations between these graphs. Therefore, based on the graphs of 400,000 domains, a global domain name resolution network has been established to analyze the problem of vulnerability. From an overall perspective, this network is a scale-free network, which exhibits robustness against random failures and vulnerabilities against intentional attacks. This is verified by our simulation of random attacks and core attacks. The resolution failure rate assessment has also been utilized to compute the resolution failure rate, and the load transfer assessment has been employed to calculate the change rate of the load after attacks. The experimental results show that when the key nodes of the first 1% are removed, 46.19% of the domain names become unresolved, and the average load of the residual nodes will increase by nearly 195%, while only 0.01% of domain names fail to be resolved and about 18% of load increase on the same attack scale of the random mode. Moreover, another set of 200,000 domain names were selected from the 1 million domain names to do the same experiment and further evidence the conclusion.

In addition, the classic node centrality measures of the complex network have been introduced and a new method has been proposed to mine the core nodes of the resolution network. In the experiment of simulating the core attack, the effect of these measures is also tested.

These findings provide new references for the configuration of the DNS. DNS administrators should take effective security measures to prevent the core nodes from being attacked. From the macro, it can help to find out weakness and problems in the design of the current domain name system.

Data Availability

The domain name resolution data used to support the findings of this study are currently under embargo, while the research findings are commercialized. Requests for data, 6 months after publication of this article, will be considered by the corresponding author.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Science and Technology Major Project under Grant no. 2017YFB0803001 and the National Natural Science Foundation of China under nos. 61370215, 61370211, and 61571144.

References

[1] G. C. M. Moura, R. D. O. Schmidt, J. Heidemann et al., "Anycast vs. DDoS: evaluating the november 2015 root DNS event," in *Proceedings of the 2016 ACM on Internet Measurement Conference*, Santa Monica, CA, USA, November 2016.

[2] Z. Liu, B. Huffaker, M. Fomenkov, N. Brownlee, and K. Claffy, "Two days in the life of the DNS anycast root servers," in *Proceedings of the International Conference on Passive and Active Network Measurement*, Louvain-la-Neuve, Belgium, April 2007.

[3] Y. Xuebiao et al., "DNS measurements at the .CN TLD servers," in *Proceedings of the Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, IEEE, Tianjin, China, August 2009.

[4] P. Mockapetris, Domain Names—Concepts and Facilities, RFC 1034, November 1987.

[5] P. Mockapetris, Domain Names—Implementation and Specification, RFC 1035, November 1987.

[6] Alexa.com[EB/OL], http://www.alexa.com/.

[7] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–382, 2000.

[8] L. Lü, D. Chen, X.-L. Ren, Q.-M. Zhang, Y.-C. Zhang, and T. Zhou, "Vital nodes identification in complex networks," *Physics Reports*, vol. 650, pp. 1–63, 2016.

[9] P. Bonacich, "Power and centrality: a family of measures," *American Journal of Sociology*, vol. 92, no. 5, pp. 1170–1182, 1987.

[10] C. Saint-Pierre, F. Cifuentes, and J. Bustos-Jimenez, "Detecting anomalies in DNS protocol traces via passive testing and process mining," in *Proceedings of the IEEE Conference on Communications and Network Security*, pp. 520-521, San Francisco, CA, USA, October 2014.

[11] D. Dagon, "Large-scale DNS data analysis," in *Proceedings of the 2012 ACM conference on Computer and communications security–CCS'12*, pp. 1054-1055, Raleigh, NC, USA, October 2012.

[12] A. Berger, A. D'Alconzo, W. N. Gansterer, and A. Pescapé, "Mining agile DNS traffic using graph analysis for cybercrime detection," *Computer Networks*, vol. 100, pp. 28–44, 2016.

[13] D. C. Macfarland, C. A. Shue, and A. J. Kalafut, "The best bang for the byte: characterizing the potential of DNS amplification attacks," *Computer Networks*, vol. 116, pp. 12–21, 2017.

[14] S. Kim, S. Lee, G. Cho, M. E. Ahmed, J. Jeong, and H. Kim, "Preventing DNS amplification attacks using the history of DNS queries with SDN," in *Proceedings of the European Symposium on Research in Computer Security*, pp. 135–152, Oslo, Norway, September 2017.

[15] E. Casalicchio, M. Caselli, and A. Coletta, "Measuring the global domain name system," *IEEE Network*, vol. 27, no. 1, pp. 25–31, 2013.

[16] H. Gao, V. Yegneswaran, J. Jiang et al., "Reexamining DNS from a global recursive resolver perspective," *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 43–57, 2014.

[17] N. Alexiou, S. Basagiannis, P. Katsaros, T. Dashpande, and S. A. Smolka, "Formal analysis of the kaminsky DNS cache-poisoning attack using probabilistic model checking," in *Proceedings of the 12th International Symposium on High-Assurance Systems Engineering*, pp. 94–103, IEEE, Boca Raton, FL, USA, July 2011.

[18] C. Deccio, J. Sedayao, K. Kant, and P. Mohapatra, "Quantifying and improving DNSSEC availability," in *Proceedings of the 20th International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–7, Maui, Hawaii, July 2011.

[19] R. V. Rijswijk-Deij, A. Sperotto, and A. Pras, "Making the case for elliptic curves in DNSSEC," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 5, pp. 13–19, 2015.

[20] T. Chung, R. van Rijswijk-Deij, B. Chandrasekaran et al., "A longitudinal, end-to-end view of the DNSSEC ecosystem," in *Proceedings of the 26th USENIX Security Symposium*, pp. 1307–1322, Vancouver, BC, Canada, August 2017.

[21] H. Choi and H. Lee, "Identifying botnets by capturing group activities in DNS traffic," *Computer Networks*, vol. 56, no. 1, pp. 20–33, 2012.

[22] J. Kwon, J. Lee, H. Lee, and A. Perrig, "PsyBoG: a scalable botnet detection method for large-scale DNS traffic," *Computer Networks*, vol. 97, pp. 48–73, 2016.

[23] V. Ramasubramanian and E. G. Sirer, "Perils of transitive trust in the domain name system," in *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, pp. 379–384, Berkeley, CA, USA, October 2005.

[24] C. Deccio, J. Sedayao, K. Kant, and P. Mohapatra, "Quantifying DNS namespace influence," *Computer Networks*, vol. 56, no. 2, pp. 780–794, 2012.

[25] C. Deccio, *Quantifying and Improving Dns Availability*, University of California, Davis, CA, USA, 2010.

[26] K. Fujiwara, A. Sato, and K. Yoshida, "DNS traffic analysis: issues of IPv6 and CDN," in *Proceedings of the 12th International Symposium on Applications and the Internet*, pp. 129–137, Izmir, Turkey, July 2012.