electronics

MDPI

Article

A Quantitative Method for the DNS Isolation Management Risk Estimation

SiYu Liang, ZhiHong Tian, XinDa Cheng, Yu Jiang *, Le Wang * and Shen Su

Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China; stevewoo23@gmail.com (S.L.); tianzhihong@gzhu.edu.cn (Z.T.); 2111906003@e.gzhu.edu.cn (X.C.); sushen@gzhu.edu.cn (S.S.)

* Correspondence: jiangyu@gzhu.edu.cn (Y.J.); wangle@gzhu.edu.cn (L.W.)

Received: 12 April 2020; Accepted: 17 May 2020; Published: 1 June 2020

Abstract: The domain name system (DNS) is an important infrastructure of the Internet, providing domain name resolution services for almost all Internet communication systems. However, the current DNS is centrally managed, leading to unfair sovereignty of the Internet among countries. A domestic DNS is unable to work normally, noted as isolated management risk (IMR), especially when the national network is isolated from the rest of the Internet. To improve understanding of the DNS isolated management risk for better DNS resource deployment, it is critical to determine how serious the IMR is among various countries. In order to quantify DNS isolated management risk, this paper proposed an effective approach to collect DNS resolution demand data from the network used by various intelligent devices and to conduct data analysis to estimate isolated management risk of certain country's domestic DNSs. Our idea is to quantify the domain name resolution demand and its relationship with the overseas resolution processes. We further used our quantitative method to compare the IMR of the USA and China and analyzed the difference between them.

Keywords: domain name system; isolated management risk; cyberspace sovereignty; DNS data collection; DNS data analysis

1. Introduction

As an important infrastructure of the Internet, the DNS (domain name system) [1] provides domain name resolution services for the majority of network communication systems. Especially with the development of intelligent communication systems in recent years, the Internet plays a more important role. As a basic support for maintaining operation of the Internet worldwide, the DNS has become more and more influential. This paper focuses on the security issues of the DNS.

However, the design and management of the DNS is centralized. In a related work [2], Professor Fang pointed out that the centralized structure of the DNS had the risk of power abuse, including disappearing management risk, blind management risk, isolated management risk, and hijacking management risk, and their definitions have been widely known in recent years. This work also mentioned that when an isolated management risk (IMR) occurred, the Internet, which relies on the international root domain name resolution system, was also inoperable within the country. This would have a significant impact on the domestic network and also illustrated the lack of cyberspace autonomy in this country. Therefore, IMR is a significant risk, and we cannot ignore it. IMR is the risk that the DNS services cannot be used normally when a country's network is isolated from the world Internet. Due to an uneven distribution of authoritative DNS servers, many countries cannot complete most of the DNS resolution requirements within the country and need to rely on servers outside the country. The occurrence of IMR causes domestic and overseas servers to be unable to communicate, which affects most of the DNS resolution requirements, making the domestic network availability

greatly reduced. The national cyberspace sovereignty is threatened. Therefore, it is very important to quantitatively evaluate IMR, so that the relevant departments of a country can find the gap between the deployment of domestic DNS servers and user needs as soon as possible and more directly understand the cost of eliminating IMR. However, no quantitative evaluation method for IMR has been proposed in previous research work, and we are the first team to carry out this work. Therefore, in order for each network management department to clearly understand the seriousness of this risk in their cyberspace and begin to carry out prevention work, it is necessary to find a method to quantitatively evaluate IMR.

Measuring the dependence on overseas domain name servers is the core idea of this evaluation method. Therefore, to quantify the evaluation of IMR, we divided it into the following two problems: "how to obtain user DNS resolution demand data" and "how to assess the dependence of these demands on overseas domain name servers". We propose solutions to these two problems in Section 4 of this article. First, we propose a method of data collection that deploys a collector to interact with the recursive server and then estimates the demand for domain name resolution. Then, we define a quantified risk index, the proportion of overseas server dependency domain (POSD), and specific evaluation methods. Through data analysis of the collected user demand data, we can see the specific severity of the IMR from the final indicator.

In order to show the feasibility of the data collection method, we performed a simulation experiment, which is described in Section 5. With reference to the logic of BIND9 software, we built a DNS scenario and simulated network traffic through a large number of random DNS resolution requests. Then, we regularly conducted repetitive measurements of the recursive server, estimated the frequency of the DNS requests, and compared them with the actual results. Finally, we found that the measurement method has a relatively stable accuracy in the scene of 0.5 query per second (QPS), which is sufficient for the IMR evaluation method.

We further use our quantitative data analysis method to compare the IMR of the United States and China, and analyze the difference between them, in Section 6. We downloaded the DNS demand dataset from an open platform and analyzed this dataset using our IMR evaluating algorithm. Through this experiment, we found that China's IMR is much higher than that of the United States where the Internet originated. We hope that with the proposed methods, we can quantitatively assess the damage of the Internet and set a cost price for the IMR, as well as make the network's potential security issues in such a communication system clearer and the service more stable.

In this paper, we present a method for quantitative analysis of IMR and POSD, and we put forward a method for measuring data collection. In addition, we conducted experiments through DNS servers in the production environment to compare the severity of IMR between China and the United States. We hope that this quantitative risk approach enables relevant defense agencies to have a clearer understanding of the impact of isolated management risk.

2. Related Work

Previous researchers have proposed many DNS defects, such as cache poisoning attacks [3], spoofing, and distributed denial of service (DDoS) attacks. The DNS risk mentioned in this paper is caused by the abuse of power due to the centralized structure design of the DNS from the perspective of national cyberspace sovereignty [4,5]. A related study recognized the current power abuse risk of the DNS. First, there was the disappearing management risk that the national top-level domain (TLD) name could be deleted by the original root service. Then, there was the blind management risk that the national recursive server resolution could be rejected by the root server, and there was also an isolated management risk that national networks could be completely isolated from the Internet. In addition, in a later work, there was a hijacking management risk that proposed overseas domain name servers could be controlled. This related study mainly carried out a qualitative analysis of the risks associated with the DNS and proposed solutions. However, it was only theoretically stated that DNS had risks, and the potential seriousness of such harm could not be seen. Therefore, the contribution of this research is mainly to propose a quantitative analysis method for isolated management risk (IMR),

with the purpose of providing a way to reveal the possible risks caused by IMR, so that relevant departments have a reason to pay attention to this problem and deploy preventive measures in a timely manner.

In fact, for the defects of the power abuse risk brought on by the DNS centralized structure, related defense studies are currently in progress, for example, the study on building a new DNS. The basic idea of this study was to separate the authorization and domain name resolution mechanisms and decentralize the resolution service. Reference [6] proposed a method of national root and alliance—i.e., through a nation network autonomy, there is cooperation between countries to balance the central power and return the Internet to equality. Similar to solving the risk of the DNS power abuse through a decentralized approach, there is also a public blockchain-based solution, i.e., Namecoin [7], which is similar to the Bitcoin system. It implements group trust through the PoW (proof of work) [8] algorithm, so that each node can reach a consensus on the DNS zone database and use this database to complete the domain name resolution work. Namecoin is a system that is already operating in a production environment. Although the recognition is not high due to its regulatory issues, it also provides new ideas for solving the problem of the DNS power abuse risk.

The above related defense works are meaningful. However, due to the existence of the risk of DNS power abuse, it is difficult for us to foretell which day our resolution services will be affected, so we need to make a strategic layout to strengthen defense. However, current work is mainly based on the qualitative analysis of risks, and the proposed solutions are only for general problems. Therefore, the quantitative analysis method of risk is helpful to promote the progress of this kind of work. In this paper, we mainly discuss how to analyze the DNS request data of a large number of users, and therefore, the key issue is how to perform data collection and data analysis. At present, there have been many studies to evaluate the risks in intelligent communication systems [9–23], mainly using data and network traffic generated in the past for analysis and modeling, so as to obtain the data that need to be measured or predicted. There is relatively little research on risk assessment for DNS, and some research on DNS data collection and DNS simulation, such as LDplayer [24], which is a DNS traffic replay tracer, proposed in 2018, which was mainly used for simulation. In the data collection section, they used several methods that collected sufficient DNS demand data for simulation, such as network behavior detection [25], reuse of malicious traffic data [26], and so on. However, these methods could not directly reflect the real domain name resolution requirements of each device and the risk indicators of the domain name system. Therefore, we also need to design effective data collection and data analysis methods to evaluate IMR.

3. Problem Statement

In an intelligent communication system, the security and stability of the network is very important, and the DNS, a critical Internet infrastructure that most user behaviors on the Internet depend on, plays this crucial role. The DNS resolution process is not complex; however, the current DNS is centrally managed and causes an unfair management situation. This situation is mainly reflected by the fact that the resolution of the secondary domain name depends on the reliable resolution of the TLD name, and the resolution of the TLD name depends on the correct reply of the root domain server. This means that if countries or organizations cannot fundamentally guarantee the reliability of the resolution to the upper-level domain name, there is a certain risk in the domain name resolution of their business. However, identifying how to reduce such risk is beyond the scope of the paper, as it involves various other factors. In this paper, we focus on how to quantitatively assess such risk and explore the cost of ignoring this risk. At present, the main problem we face is that this risk exists only in theory and cannot be displayed in actual data, as well as a lack of a reasonable cost assessment method. This article addresses how to use reasonable data collection and data analyzing methods to give a way to see how harmful this risk is.

3.1. DNS Design Flaw

As we all know, the DNS is a centralized system with a hierarchical structure. Its operation is basically dependent on the root domain name servers, or in other words, the highest level domain name server. For example, we now want to resolve the domain name "www.baidu.com", and the question is "What should we do?"

Generally, when a normal user resolves a domain name, a recursive resolution service provided by the operator or other manufacturers is used, instead of the user taking the resolution step by step. Therefore, the user sends a request to resolve a domain name through the DNS protocol to the recursive server and then waits for the recursive server to process this DNS resolution demand. The role of the recursive server is to perform iterative parsing, as shown in the left diagram in Figure 1. First, the recursive server obtains the TLD authority server IP address of "com" from the root server and then fetches the domain name server IP address of "baidu.com" from this TLD authority server, and finally gets the IP address of "www.baidu.com" from the secondary domain name server. From this process, it can be found that due to the hierarchical design of DNS, domain name resolution is heavily dependent on higher level domain name servers. For example, if the resolution fails at the root server, subsequent iterative parsing fails. If you fail to get the IP address of the "com" name server deployed in another country, you cannot resolve the "baidu.com", even if its name server is deployed in your country. This is not a reasonable phenomenon in many scenarios.

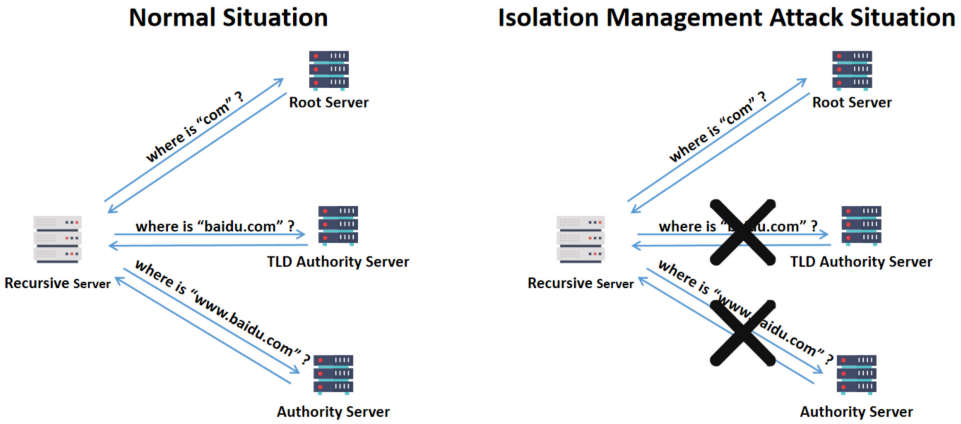

Figure 1. Isolated management attack.

The centralized design implemented by the root server can solve the credibility problem in DNS resolution. However, DNS is a system used worldwide, which means that its participants include countries and governments, and its centralized design brings management risks. When the DNS management power cannot be equally distributed to each country, it means that the independence of national cyberspace sovereignty cannot be guaranteed.

3.2. Isolation Management Risk

Users basically rely on the DNS when they use Internet services. The stability of DNS services is very important to the availability of a country's network. If a country's DNS service fails on a large scale, the profit of Internet users and service providers is ultimately affected. Is it possible to achieve that by exploiting DNS design flaws? The design of the DNS is centralized, which means that some countries cannot fundamentally guarantee the stability of DNS domestically. In Figure 2, we assume that an attacker launches an attack on a country's network outside the border. A typical attacker can have the following attack capabilities:

1. Blocking network capability means the attacker can perform DDoS attacks on the network infrastructure of the country, such as export routing service;

2. Hijacking service capability means the attacker can use cryptographic attacks or political power to hijack DNS services and send large amounts of forged data to the attack target.

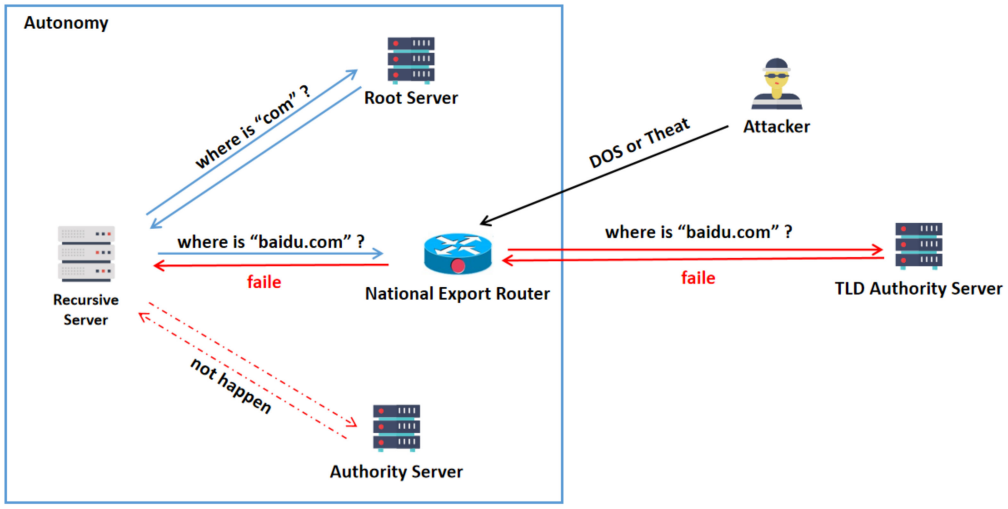

Figure 2. Isolated management attack.

An attacker with one of the above attack capabilities poses a threat to the DNS service of a country as follows:

- Overseas network unavailable, i.e., due to the unavailability of the outbound route, network equipment in the country cannot reach the network outside the country and cannot access overseas domain name servers;
- Overseas DNS packets are not trusted, i.e., if the DNS service is hijacked, the attacker generates DNS packets from untrustworthy overseas domain name servers, resulting in a large number of DNS resolutions that are untrustworthy or even unavailable.

In extreme cases, no one can prove that a capable organization will not invalidate the DNS of a country by blocking the export routes of certain countries or controlling the root server, for instance. When a country cannot properly communicate with the outside world at the IP layer or cannot believe DNS packets from overseas, the result is that domestic users are unable to access the necessary overseas domain name server, and the operation of the DNS is affected. For example, in China, resolutions of secondary domain name with non-country-code TLD names (such as baidu.com) will be affected as a result. This poses a certain threat to the nation's cyberspace sovereignty. If a country's network operation relies heavily on overseas services, the country's ability to self-operate its cyberspace is very weak. In summary, there indeed exists an isolated management risk (IMR) in DNS, and estimating the impact of IMR is the main task of this study.

3.3. Damage of IMR

In the previous sections, we discussed the flaws that result from the DNS centralized design, and we also introduced the risk of this flaw being exploited. Now, we discuss the effects when isolated management risks occur.

When a country's network is isolated from the rest of the world, it seems that users in that country cannot access the foreign network, but their domestic network communication is normal. This is true if the design of the Internet is flat. However, because the design of the DNS is centralized, this isolated attack affects the domestic Internet, the DNS fails, and the Internet is unavailable. There are some alternative methods of root zone resolution to protect the availability of root zone resolution services in the event that a country's network is disconnected. However, for some generic TLD names,

such as "com", there is no good way to complete domain name resolution inside the disconnected area. Therefore, the main damage of IMR is that these TLD names cannot be resolved in the isolated area. The resolution requirements of such domain names cannot not be met. To assess how harmful the IMR is, it is important to actually assess how big these needs are.

It may not be adequate to discuss risk in principle, because when what we say is not closely linked to benefits, it is often not a concern. Quantitative assessment of the harm caused by IMR is actually exposing the cost of ignoring IMR, and it is necessary for organizations with power to see this potential risk more clearly.

4. Method for the Assessment of Isolated Management Risk

In the previous sections, we introduced the isolated management risk (IMR). We know that the management method of the DNS is not totally equal. This method of management brings potential dangers, and there is no good solution at present. In this section, we introduce a method for evaluating the damage of IMR.

4.1. Proportion of Overseas Server Dependency

It is not enough to know the existence of IMR. In order for others to pay attention to this risk, we need to quantify the potential harm of the risk. The purpose of this paper is to numerically assess the risk. IMR is just a potential hazard, and the occurrence of an isolated attack indicates that the hazard has already occurred. We propose an indicator, the proportion of overseas server dependency domain (POSD), to measure the degree of the isolated attack, and then assess the degree of the IMR.

We design the POSD evaluation method based on user demands. First, we need to introduce a new concept, user domain name demand list (UDDL), which represents the requirement of users for DNS resolution. When we evaluate the risk of DNS, we are actually assessing the harm caused when the risk is exploited, and therefore, we spend more time discussing the impact of the isolated attack on users. Therefore, we consider how to evaluate the impact of a DNS attack on users. We could start with the users' DNS resolution requirements. In a normal situation, Internet users can query the domain name message using the DNS. The system works fine in that it can meet the normal needs of users. When people are in an attacked environment, a certain proportion of domain names that could not be resolved normally become unresponsive. The essence of our evaluation method is to find these failed domain name requirements and calculate their weight in all user demands. This reflects the extent of the attacking damage, and in turn reflects the magnitude of the risk.

Therefore, we can split the POSD evaluation work into two goals, i.e., collecting users' DNS resolution demand and classifying domain names affected by an isolated attack.

To know what domain names the user needs to resolve, we introduce the UDDL concept, which is discussed in detail in the next section. In simple terms, the UDDL is a dataset of user demand for DNS resolution. It is an array of domain names. A UDDL would not cover all DNS resolution requests of all users due to the fact that it is impossible to collect all the resolution request data from all networks, including public free networks and private networks. Therefore, we consider a different solution, in which we only need to get the user demands that are most representative.

To distinguish whether user DNS resolution demands are affected by the isolated attack, we propose a definition, overseas server dependency domain (OSD), to mark domain names affected by an isolated attack. We introduce the details of the OSD marking method in the next section. In essence, an isolated attack is to isolate the user's network from the outside, so that the user cannot resolve some domain names.

What is the key problem for users who cannot complete a DNS resolution? Resolving a domain name depends on multilevel domain name servers, as shown in Figure 3. A low-level domain name resolution relies on results of high-level domain resolution—for example, the resolution of "www.baidu.com" relies on the resolution results of ".com" and "baidu.com" in sequence. If a user cannot access any domain name servers that manage the zone of "com", then they are not able to know

the IP address of the name servers that manage the zone of "baidu.com". Therefore, the key problem for users who cannot complete a DNS resolution is that during the recursive resolution, some level of domain name resolution fails. This situation can easily occur under an isolated attack, because the name server can be isolated outside the scope of isolation. We mark the domain name as OSD when it fails to resolve due to the isolation of the domain name server.

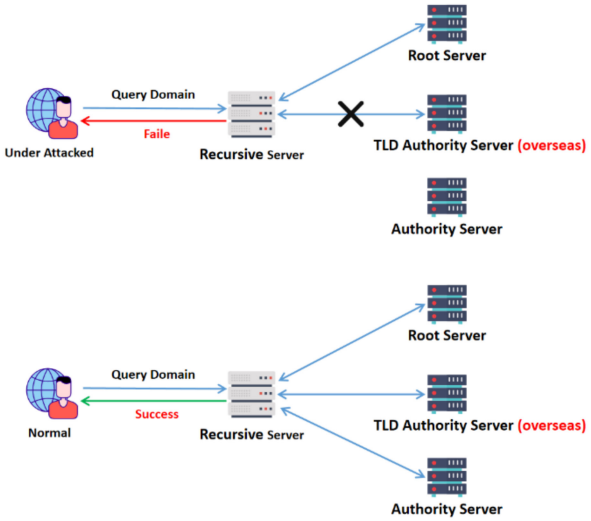

Figure 3. Effects of isolated attacks.

In the following section, we provide more descriptions in detail.

4.2. UDDL Collection Method

Our purpose is to assess the isolated management risk (IMR) for a specific area. The focus of our evaluation work is to analyze user demands, not infrastructure deployment. Therefore, before the assessment work starts, we need to collect users' DNS resolution requests data (UDDL).

4.2.1. The General Ideas of the UDDL Collection

Ideally, we could collect the user DNS requests from network operators and network administrators to estimate the overall user resolution requirements, and then filter the UDDL from these log files. However, it is difficult to get an overall DNS resolution request, because collecting DNS resolution logs for all users is an unimaginable work. Therefore, we need to focus only on the domain names with the highest demand, and the problem is divided into the following two parts: (1) find out the most frequently resolved domain names, and (2) estimate their demands for resolution.

In order to collect the most frequently resolved domain names, we can analyze public DNS logs data on some platforms and extract the top N domain names with the most resolution requirements. If we do not believe such public data platforms, or they do not meet our needs, we can also get the data in other ways, such as some network traffic analysis or even through some questionnaires. Our ultimate goal is to know which domain names are most often resolved by users. We do not have the ability to collect the entire country's user DNS resolution demand data, but this would also help us to achieve the goal of assessment work. We actually get a set of UDDL with a weight of one. This can effectively reduce our measurement range in the domain name dimension.

However, this is not enough for the proportion of overseas server dependency domain (POSD) assessment. The POSD is an indicator for user demand, which means that we not only need to obtain a list of high-demand domain names, but also need to know the demand for each domain name. In other words, we need to know the weight of the UDDL. This operation aims to estimate the amount of resolution requirement for these domain names.

In order to estimate the demand for a domain name resolution, we use the public DNS recursive server and the remaining time of the domain name cache. First, each domain name has a cache time configuration, which determines how long this domain name is cached in the recursive server. Second, we get the remaining cache time of a domain name by querying the recursive server through a normal DNS request. With these methods, we obtain the time of each cache update. At this point, we say that we know the interval between user requests and cache refresh. For example, in Figure 4, we assume that "baidu.com" is a domain name with a total cache time (TCT) of 10 s. We send a DNS request for "baidu.com" to the measured recursive server and receive the response at the time point 100 s (we treat this as a time stamp, marked as "Measure" in Figure 4). Suppose that the remaining cache time (RCT) in the response is 8 s. With these data, we can calculate the specific time when the cache was last refreshed. In other words, this domain name was cached at the time point 98 s (= 100 + 8 − 10), and the cache will be flushed at the time point 108 s (= 98 + 10). Then, we send the same DNS request at some time point after the point 108 s and receive the response at the time point 116 s, and suppose that the remaining cache time in the response is 6 s at this time, which implies the domain name was re-cached at the time point 112 s (= 116 + 6 − 10). The example could be regarded as a test.

It is noteworthy that a user's domain name resolution request does not necessarily cause cache refresh. The recursive server only re-caches the domain name record that has not been cached and deletes the domain name record whose cache time has expired. Therefore, before the cache time expires, it does not refresh the cached domain names for a new request. When the recursive server receives a domain name resolution request from a user, and the domain name is not cached, the recursive server initiates an iterative resolution. This process takes a certain amount of time, which we call recursive time gap (RcG). To obtain the RcG of resolving a specific domain name for a specific recursive server, it is only necessary to trigger the recursive parsing multiple times and record the average time taken for the iterative resolution. For this example, in the measured recursive server, we can consider that there is no DNS request about "baidu.com" between the time points 108 s and 112−RcG seconds. In other words, from this test, we know that the request gap for the domain name "baidu.com" is at least 112−RcG−108 s.

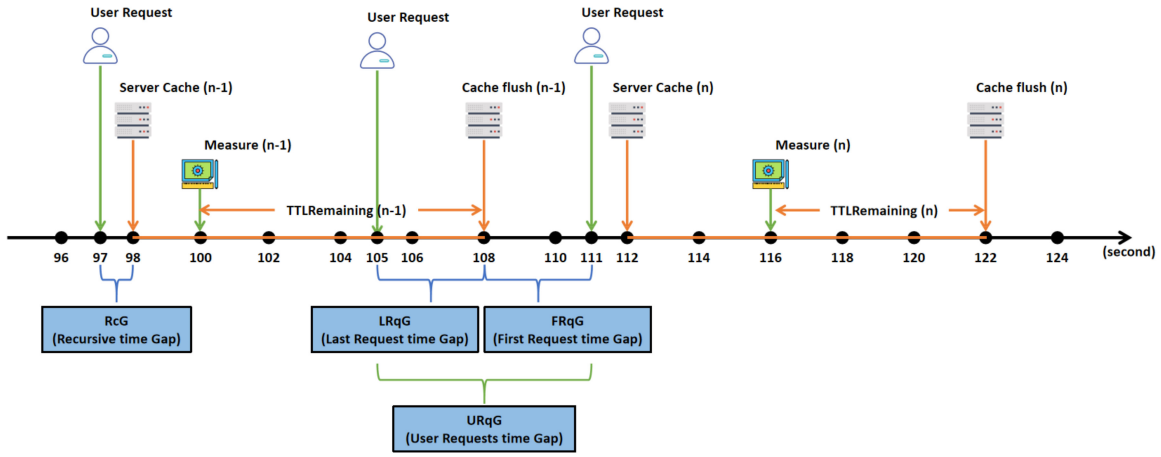

Figure 4. Measurement running example.

In order to know the time interval of user requests, however, there is a key time interval that we cannot measure directly. For convenience, the last request time gap (LRqG), which is the time gap between the time point of the last user request before a cache flushing and the time point of this cache flushing, is denoted by LRqG, and the first request time gap (FRqG), which is the time gap between the time point of the first user request after a cache flushing and the time point of this cache flushing, is denoted by FRqG. In fact, the data we ultimately want to know are the user request time gap (URqG), which is the time interval between two sequential user requests, i.e., the sum of the LRqG and an FRqG,

as shown in Formula (1). In the following section, we introduce a URqG estimation method which is based on the estimation of the LRqG.

$$URqG_{(n)} = LRqG_{(n-1)} + FRqG_{(n)} \quad (1)$$

4.2.2. URqG Estimation Method

Our goal is to know the frequency of resolution requirements for a particular domain name on a particular recursive server, and we need to obtain a large amount of URqG measurement data and then calculate the average value. We can directly measure the value of the FRqG by the remaining cache time (RCT) and the average of the RcG, but there is no way to know the LRqG, because a user's request does not trigger a cache refresh. Next, we explore a method to estimate the value of the LRqG, in order to obtain an approximate URqG.

First, we introduce how to measure the recursive time gap (RcG). The DNS resolution is an iterative process, and the recursive server is responsible for doing this. To measure the RcG of a specific domain name for a specific recursive server, our method is to send a resolution request to the recursive server when the cache is flushed and record the resolution time. By collecting and averaging these time-consuming data in large quantities, we can get the estimated value of the RcG.

Then, we can calculate the approximate value of the first request time gap (FRqG) using the RcG, and the RCT which is returned by the recursive server. The measurement function sends requests to the recursive server upon receiving the response result and analyzing and recording relevant information, including the time. We can then calculate the time when the domain name is cached, marked as "server cache" in Figure 4. In addition, we can also analyze the cache flushing time, marked as "cache flush" in Figure 4. We calculate the value of the FRqG through Formulas (2) and (3), where TCT is the total cache time of the domain name in the recursive server and can be measured through multiple regular requests by triggering the iterative resolution. The RCT is returned in the recursive server response, meaning that the domain name cache is flushed after this time, which is precise to the second. Notice that BIND9's RCT data accuracy is in the order of seconds, resulting in inevitable errors, and thus, the accuracy is low in high QPS scenarios.

$$FRqG_{(n)} = Measure_{(n)} - (TCT - RCT_{(n)}) - avg(RcG) - CacheFlush_{(n-1)} \quad (2)$$

$$CacheFlush_{(n-1)} = Measure_{(n-1)} + RCT_{(n-1)} \quad (3)$$

After obtaining the FRqG, we need to estimate the LRqG. Since we cannot measure LRqG directly, we can only get its approximate value through other ways. Because DNS requests are discrete random independent events, a measurement method needs to obtain a large amount of such data. Therefore, under the analysis of large-scale data, we assume that there is an expected URqG value, denoted by E(URqG). In this case, the maximum value of the FRqG is assumed to be slightly greater than the E(URqG), and the minimum value of the FRqG is assumed to be slightly greater than zero, and the same is true for the LRqG, i.e., $\max(LRqG) \approx \max(FRqG) > E(URqG)$ and $\min(LRqG) \approx \min(FRqG) > 0$. By Formula (1), when the time point of cache flush is close to the time point of sending the previous request, and thus, the LRqG is small, then the FRqG is large, and vice versa. Thus, the FRqG and LRqG are related to their opposites within each URqG.

On the basis of the above inferences, we cannot yet get an accurate LRqG, but because the LRqG is oppositely related to the FRqG, we can infer the approximate value of the LRqG by dichotomy. First, we obtain a subset denoted by FRqG* from the FRqG measurement dataset, with FRqGs which are greater than the average of the FRqG, as shown in Formula (4). We choose the average of the FRqG instead of the median due to the fact that an average value takes the overall situation of FRqG

into account. Since in most practical situations, the FRqG is less than the E(URqG), we assume that E(URqG) is greater than the average of the FRqG, as shown in Formula (5).

$$\text{FRqG*} = \{x | x \text{ is an FRqG which is greater than the avg(FRqG)}\} \tag{4}$$

$$\text{E(URqG)} \geq \text{avg(FRqG)} \tag{5}$$

Therefore, we use Formula (6) to calculate the expected value of the LRqG, i.e., E(LRqG), and then combine it with Formula (1) to estimate the URqG, as shown in Formula (7). When processing measurement data, we only need to calculate the average of the FRqG* to estimate the value of the E(URqG).

$$\text{E(LRqG)} \approx \text{avg(FRqG*)} - \text{avg(FRqG)} \tag{6}$$

$$\text{E(URqG)} \approx \text{E(LRqG)} + \text{avg(FRqG)} \tag{7}$$

URqG is the time interval between two user requests, and its average represents user's request period for a specific domain name on a specific server. In other words, the inverse of URqG is the frequency of domain name resolution demand that we want to measure. This frequency can be used as the weight in the UDDL. After measuring all the domain names in the UDDL, we can proceed to the next step.

4.3. Resolution Demand Analysis

4.3.1. Tracking the Domain Name Resolution Process for Identifying OSD

Why do we track the domain name resolution process next? In the previous section, we introduced the method of UDDL collection, which means that we know what domain names the user often asks to resolve. The proportion of overseas server dependency domain (POSD) is an indicator based on user demand analysis, while tracking the domain name resolution process is an effective method to analyze user demand. With this method, we calculate the user's resolution requirement for domain name servers outside the isolated range.

Before introducing how to track a domain name resolution process, we have to emphasize the definition of OSD again. The purpose of calculating POSD is to know how much the user's needs in the UDDL require overseas servers. It was mentioned in the previous sections that a user would be unable to access the server outside the range of the isolated network when isolated attacks take effect (we named those servers overseas servers). Therefore, all domain name resolution processes that rely on overseas servers are invalidated under an isolated attack. In other words, a domain name that cannot be resolved in the cyberspace of a measuring country is marked as OSD. It is worth noting that domain name resolution is an iterative process. The recursive server needs to access the root domain name server, the top-level domain name server, the authoritative domain name server, etc. The availability of each access link is indispensable. For example, among all domain name servers in China, there is no domain name server that provides resolution for the "com" top-level domains, which means that we are unable to rely solely on DNS services within China to complete the resolution of domain names with TLD "com", although there are many secondary domain name servers for "*.com" that have been deployed in China. Before users in China resolve "*.com" (such as "baidu.com"), they must access the "com" name server in another country to obtain the IP address of the secondary domain name server. If an isolated attack occurs in China, users in the country are not able to resolve any domain name under "com", even if second-level domain name servers have been deployed in China, such as "taobao.com" and "cctv.com". This is not equality. We call these domain names overseas server dependency domains (OSDs). Notice that to determine whether a domain name could be marked as an OSD, you need to select a region as a prerequisite.

Now, we introduce the OSD identifying method. In order to know whether a domain name resolution process relies on an overseas server, we start with its resolution process. We know that the domain name resolution is performed in stages. First, resolve the TLD name, then resolve the

secondary domain name, and finally, resolve the host domain name, as shown in Figure 5. During the process, we could get an IP list of each level domain name, because there are several name servers for each level of domain name, which offer the resolution service for users.

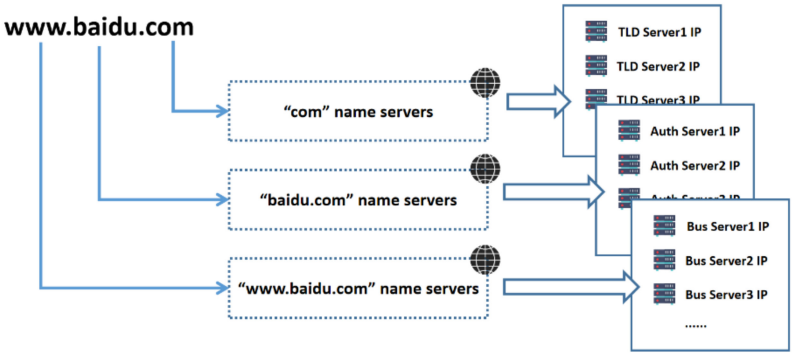

Figure 5. Running example of tracking a domain name server.

By following the process of domain name resolution, we obtain a list of classified name servers' IP addresses. Next, we should determine the server location by their IP addresses. Why should we know where the name servers are? The answer is, if we know the location of the domain name servers, we can find out if it is enough to complete the entire domain name resolution process in the isolated areas. Suppose that we are going to resolve "www.cctv.cn" in China. Since for each segment of this domain name, there is a corresponding domain name server in China, we mark "www.cctv.cn" as a normal domain name, not as an OSD. As another example, suppose we are tracking "www.baidu.com". We find that all of the domain name servers' IP addresses of "com" are not in China. Although several name servers for secondary domain and host domain are deployed in China, we mark "www.baidu.com" as an OSD, because we could not complete the resolution process without the "com" name server which is deployed in another country. In other words, we can determine whether a domain name is OSD by identifying whether the area we choose has the ability to provide a complete resolution service for this domain name. This process is described by Algorithm 1, OSD Identification.

Algorithm 1 OSD Identification (domainName, area)

Input: A domain name and the area to be measured
Output: Whether this input domain name belongs to the OSD domain name

```
subDomain = splitDomain(domainName)
isOverseasDomain = false
for each item ∈ subDomains do
    IPs = digDomain(item)
    isDomesticSubDomain = isExistDomesticServer(IPs, area)
    if isDomesticSubDomain == false then
        isOverseasDomain = true
        break
    end if
end for
return isOverseasDomain

function isExistDomesticServer(IPs, area)
    isExist = false
    for each item ∈ IPs do
        location = getIPGeolocation(item)
        locationIsBelongsToArea = isBelongsTo(location, area)
```

```
18:             if locationIsBelongsToArea == true then
19:                 isExist = true
20:                 break;
21:             end if
22:         end for
23:         return isExist
24:     end function
```

4.3.2. Batch Statistical Process

Now, after collecting the user's DNS resolution demand data, the next step is critical in our assessment work and may take a lot of time. We need to do the OSD identifying work for the domain names in UDDL.

The result of the POSD assessment shows how harmful the isolated risk is. In other words, this calculates how much damage the isolated attack will cause. Therefore, a problem is how to define the amount of damage. Under normal circumstances, users can perform domain name resolution according to their demands and get normal results. This is a manifestation of peace. If they suffer an isolated attack, then OSD domain name resolution would fail. We define the proportion of affected domain name in UDDL as the amount of damage. Therefore, we calculate the POSD via the following Formula (8):

$$\text{POSD} = |\text{OSD}|/|\text{UDDL}| \tag{8}$$

In general, the POSD is numerically the proportion of the OSD in the UDDL. We measure and obtain the UDDL through certain methods, then mark the OSD by tracking the resolution process, and finally calculate the POSD. We define the amount of damage caused by an assumed isolated attack as the size of isolated management risk (IMR).

5. Evaluation

To evaluate the effectiveness of the DNS resolution requirements estimation method, we develop an emulator, which emulates the behaviors of normal users in the domain name system. In order to simulate the operation of our estimation method, we add the measurement module in the emulation environment.

We design this program according to the regular DNS architecture, which has one DNS authority server and a recursive resolve server. In the experiment, we deploy one resolvable domain name on authority server with the total cache time (TCT) set to 300 s. Meanwhile, we simulate the behavior of normal users, to make DNS requests to the recursive server for resolving that domain name. It is worth noting that the random requests for this evaluation experiment need to conform to the Poisson distribution, and the random requests must be discrete with equal probability, due to the fact that the actual user requests usually conform to the Poisson distribution.

For this evaluation, we used the BIND9 cache rule, and the corresponding remaining cache time (RCT) is in the order of seconds; therefore, the result of URqG has an inevitable error of ±1 s. The results of this measurement method would have greater error in a scene with high frequency, and therefore, we need to observe the estimation result under a different request frequency. The QPS (query per second) configuration of normal users is required to be set dynamically in the emulator. We initialize the QPS of the normal DNS requests to be about 2.2, and decreasing for each round of experiments. We configure a fixed QPS for each round of experiments, and the frequency of randomly generated user requests basically follows this QPS. Then, we perform 100 rounds with QPS decreasing by 0.06 per round. Meanwhile, in order to obtain effective experimental data more efficiently, we set the measurement interval to about 300 + 18 s. In actual situations, this configuration can be dynamically adjusted according to the success rate of the data collection. To make experimental results easier to understand, we set the abscissa in Figure 6, and Figure 7 as the inverse of QPS which is the user request time gap (URqG).

During the simulation, we obtain the cache status from the recursive server by continuously calling the measurement method module. From the data in Figure 6, we find that when the simulated request gap is lower, and the measured user QPS is much lower than the ideal QPS; this is mainly due to the fact that the corresponding data of BIND9 re rounded down, especially in the high- frequency scene, and thus, many calculated values of FRqG are less than zero and must be discarded. When the calculated FRqG is less than zero because of the existence of the rounded error, the estimated LRqG error must be very large, and thus, the measured values are in fact meaningless. When the URqG is around 1 s, we find that its accuracy is relatively high. This is because the error brought by rounded precision is exactly ±1 s. When we configure the simulated request gap to be near 1 s, which is equal to the absolute value of the error, the measured URqG is not much different from the ideal URqG. This means that simulated request gap should not be too small, i.e., simulated QPS should not be too high.

With the continuous adjustment of the experimental configuration, the simulated request gap keeps increasing (i.e., the simulated QPS keeps decreasing). When the simulated request gap is set to be greater than 2 s, the accuracy is stable around 0.8, as shown in Figure 7.

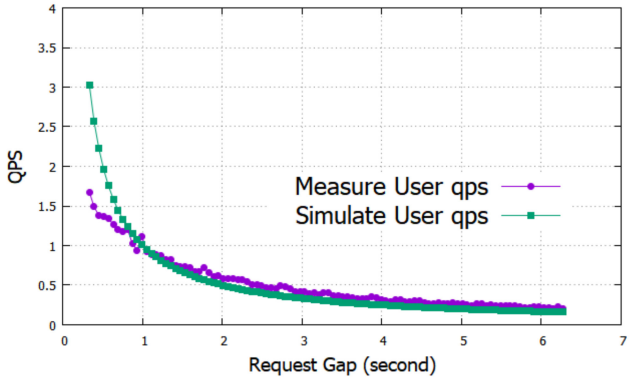

Figure 6. Comparison of measurement results with simulated data.

Figure 7. Deviation of the measurement results from the simulation data.

It is worth noting that the URqG can only be measured after the cache refresh event is triggered by a normal DNS request. When conducting an IMR assessment, we do not require 100% user domain name resolution demand data, and what is needed is just a rough ratio. Therefore, we can set a better measurement interval and obtain effective measurement data with an acceptable accuracy. Through this evaluation experiment, it can be seen that this measurement method is suitable for scenes with QPS less than 0.5, and it can basically achieve an accuracy of about 78%.

6. Experiment

In previous sections, we introduced the isolated management risk (IMR) and proposed a quantitative analysis method for IMR. The proposed POSD method runs in one machine which

is configured with an 8 GB DDR3 memory, an Intel i5-7500 CPU of 3.4 GHZ, and four core per CPU. All the codes of the program are implemented by Python3.6. The operating system of the machine is Ubuntu Linux. All the datasets in this experiment are downloaded from RAPID [27]. The datasets used in the experiment were essentially collected from a different DNS forwarding server around the world, which can be used to represent the user's demand for DNS resolution. In this study, we used the dataset named "2019-10-27-1572199582-fdns_cname.json".

Before running the measurement program, we did some data cleaning for the dataset, because there can be a lot of DNS request data on that RAPID platform that we do not need, such as a large number of meaningless domain names. We mainly cleaned up two types of data. The first type is a domain name that is repeatedly reported failure when it is resolved using a public recursive resolution server more than five times. The second category is a domain name such as "adsl-pool.sx.cn", "host.vortexnode.com", "rene-liu.com", "packetexchange.net", or "ptr.anycast.net", which has subdomains which are randomized for temporary services or generated by exhaustive scanning and which seems like a temporary domain name.

The POSD evaluation algorithm needs more DNS resolution requirements actually generated by users. For the actual evaluation effect, these misleading data should be deleted as much as possible. Readers who need to use other datasets for experiments can also clean up data according to this principle. Because this experiment only uses public datasets for evaluation, and we assume that the data in the dataset is the user's needs regarding DNS resolution (i.e., UDDL), the results of the experiment are heavily dependent on the dataset.

By comparing the differences of various geographic information databases [28], we chose three geographic information databases obtained on ipplus360 [29], Taobao [30], and GeoLite2 [31] for the experiment. In the GeoLite2 platform, we used Lite2-Country_20200421, a database with relatively insufficient data volume; therefore, the results had some deviations, but we found that this has little effect on the results.

The experiment result, shown in Figure 8, shows that in China, about 91% of the domain names in the dataset we evaluated were marked as the OSD, which means that the vast majority of user demand could be influenced by an isolated attack. However, for the same dataset, the proportion of OSD in the United States is one-third less than it is in China, which represents a relatively low IMR in the United States. Figures 9 and 10 show that most of these domain names under "com" are marked as the OSD, because these kinds of domain names are mainly used by companies from different countries that deploy their domain name servers in their nation. From the analysis of these data, we can see that the demand for domain name resolution in China's cyberspace is highly dependent on overseas domain name servers, which refutes the suggestion that the original intention of DNS designs is equality, relying on domestic network equipment as much as possible to meet users' DNS resolution needs. The result reflects an embodiment of independence in cyberspace.

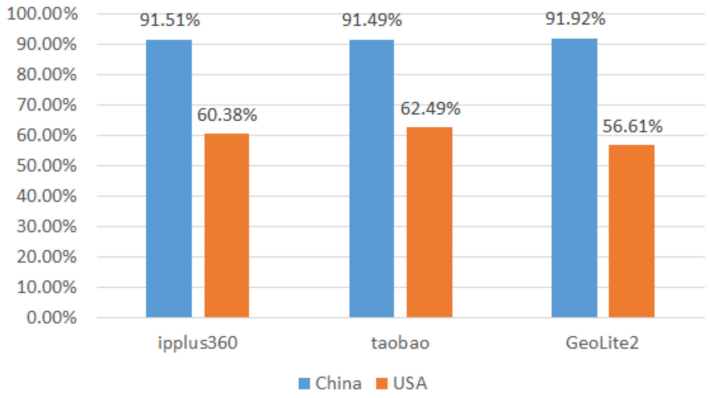

Figure 8. Proportion of overseas server dependency domain (POSD) of China and the USA.

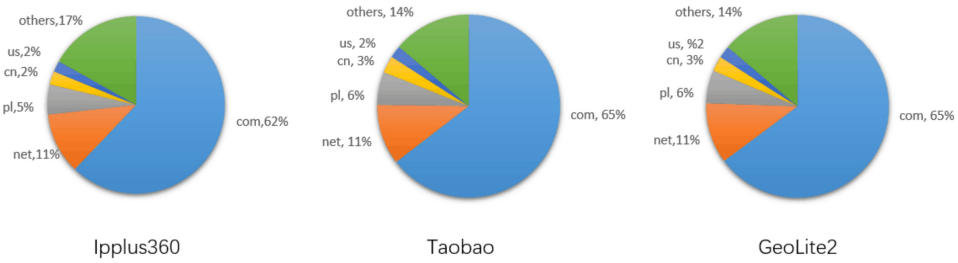

Figure 9. Top 5 top-level domains (TLD) of China overseas server dependency domain (OSD).

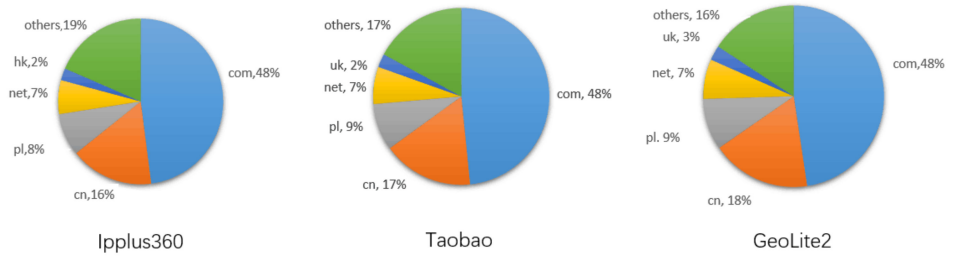

Figure 10. Top 5 TLD of USA overseas server dependency domain (OSD).

7. Conclusions

With the increasing scale of the Internet, the influence of the DNS is growing. The risks of the DNS due to centralized management are worthy of attention. In this paper, we introduce the potential risk in the network used by most communication systems, i.e., isolated management risk (IMR), and propose a quantitative analysis method, including the OSD identification algorithm, to effectively assess the degree of potential IMR in a certain area/country. We also provide a method for estimating user request gaps (URqG) via a simulation experiment. Finally, we conducted an IMR assessment experiment using public DNS datasets and IP geographic location information data. Experiment results showed that the IMR of the Internet used by communication services cannot be neglected, and relevant departments should take measures to improve the stability of the intelligent communication system and the autonomy of the national network system.

Author Contributions: Conceptualization, S.L., S.S., and Y.J.; Data curation, S.L. and X.C.; Formal analysis, S.L., X.C., and S.S.; Funding acquisition, Z.T., Y.J., and S.S.; Investigation, S.L. and X.C.; Methodology, S.L., X.C., and S.S.; Project administration, S.L. and S.S.; Resources, S.L., X.C., and S.S.; Software, S.L. and X.C.; Supervision, Z.T., Y.J., L.W., and S.S.; Validation, S.L. and X.C.; Visualization, S.L. and X.C.; Writing—Original draft, S.L.; Writing—Review and editing, S.L., S.S., and Y.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work is funded in part by the Guangdong Province Key Research and Development Plan (grant no. 2019B010137004), the National Key research and Development Plan (grant no. 2018YFB0803504), the National Natural Science Foundation of China (grant nos. U1636215, 61902083, 61976064), and the Guangdong Province Universities and Colleges Pearl River Scholar Funded Scheme (2019).

Acknowledgments: Our deepest gratitude is extended to our "shepherd", B.X. Fang.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Domain Name System. Available online: https://en.wikipedia.org/wiki/Domain_Name_System (accessed on 2 February 2020).
2. Fang, B.X. Country autonomous root domain name resolution architecture from the perfective of country cyber sovereignty. *Inf. Secur. Commun. Priv.* **2014**, *12*, 35–38.
3. Liu, Y.; Yue, M.; Tang, J.; Miao, L. Security analysis of internet domain names system. *Netinfo Secur.* **2010**, *12*, 14–16.

4. Fang, B.X. A hierarchy model on the research fields of cyberspace security technology. *Chin. J. Netw. Inf. Secur.* **2016**, *1*, 2–7.
5. Li, J.H.; Qiu, W.D.; Meng, K.; Wu, J. Discipline construction and talents training of cyberspace security. *J. Inf. Secur. Res.* **2015**, *1*, 149–154.
6. Zhang, Y.; Xia, Z.; Fang, B.; Zhang, H. An autonomous open root resolution architecture for domain name system in the internet. *J. Cyber Secur. Oct.* **2014**, *2*. [CrossRef]
7. Namecoin. Available online: http://namecoin.info (accessed on 2 February 2020).
8. Nakamoto, S.; Bitcoin, A. A Peer-to-Peer Electronic Cash System. Bitcoin. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 2 February 2020).
9. Li, M.; Sun, Y.; Su, S.; Tian, Z.; Wang, Y.; Wang, X. DPIF: A framework for distinguishing unintentional quality problems from potential shilling attacks. *Comput. Mater. Contin.* **2019**, *59*, 331–344. [CrossRef]
10. Wang, B.; Kong, W.; Li, W.; Xiong, N.N. A dual-chaining watermark scheme for data integrity protection in internet of things. *Comput. Mater. Contin.* **2019**, *58*, 679–695. [CrossRef]
11. Tan, T.; Wang, B.; Tang, Y.; Zhou, X.; Han, J. A method for vulnerability database quantitative evaluation. *Comput. Mater. Contin.* **2019**, *61*, 1129–1144. [CrossRef]
12. Cheng, J.; Xu, R.; Tang, X.; Sheng, V.S.; Cai, C. An abnormal network flow feature sequence prediction approach for ddos attacks detection in big data environment. *Comput. Mater. Contin.* **2018**, *55*, 95–119.
13. Wang, B.; Gu, X.; Yan, S. STCS: A practical solar radiation based temperature correction scheme in meteorological WSN. *Int. J. Sens. Netw.* **2018**, *28*, 22–33. [CrossRef]
14. Li, M.; Sun, Y.; Lu, H.; Maharjan, S.; Tian, Z. Deep reinforcement learning for partially observable data poisoning attack in crowdsensing systems. *IEEE Internet Things J.* **2020**. [CrossRef]
15. Tian, Z.; Luo, C.; Qiu, J.; Du, X.; Guizani, M. A distributed deep learning system for web attack detection on edge devices. *IEEE Trans. Ind. Inform.* **2019**. [CrossRef]
16. Wang, B.; Gu, X.; Zhou, A. E2S2: A code dissemination approach to energy efficiency and status surveillance for wireless sensor networks. *J. Internet Technol.* **2017**, *8*, 877–885. [CrossRef]
17. Tian, Z.; Gao, X.; Su, S.; Qiu, J. Vcash: A novel reputation framework for identifying denial of traffic service in internet of connected vehicles. *IEEE Internet Things J.* **2020**. [CrossRef]
18. Tian, Z.; Shi, W.; Wang, Y.; Zhu, C.; Du, X.; Su, S.; Sun, Y.; Guizani, N. Real time lateral movement detection based on evidence reasoning network for edge computing environment. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4285–4294. [CrossRef]
19. Wang, B.; Gu, X.; Ma, L.; Yan, S. Temperature error correction based on BP neural network in meteorological WSN. *Int. J. Sens. Netw.* **2017**, *23*, 265–278. [CrossRef]
20. Yi, L.; Luo, X.; Zhu, C.; Wang, L.; Xu, Z.; Lu, H. ConnSpoiler: Disrupting C&C communication of IoT-based botnet through fast detection of anomalous domain queries. *IEEE Trans. Ind. Inform.* **2020**, *16*. [CrossRef]
21. Qiu, J.; Tian, Z.; Du, C.; Zuo, Q.; Su, S.; Fang, B. A survey on access control in the age of internet of things. *IEEE Internet Things J.* **2020**. [CrossRef]
22. Qiu, J.; Du, L.; Zhang, D.; Su, S.; Tian, Z. Nei-TTE: Intelligent traffic time estimation based on fine-grained time derivation of road segments for smart city. *IEEE Trans. Ind. Inform.* **2020**, *16*, 2659–2666. [CrossRef]
23. Wang, B.; Kong, W.; Guan, H.; Xiong, N.N. Air quality forcasting based on gated recurrent long short term memory model in internet of things. *IEEE Access* **2019**, *7*, 69524–69534. [CrossRef]
24. Zhu, L.; Heidemann, J. LDplayer: DNS experimentation at scale. In Proceedings of the Internet Measurement Conference 2018, Boston, MA, USA, 31 October–2 November 2018; pp. 119–132.
25. Bermudez, I.N.; Mellia, M.; Muna, M.M.; Keralapura, R.; Nucci, A. DNS to the rescue: Discerning content and services in a tangled web. In Proceedings of the 2012 Internet Measurement Conference, Boston, MA, USA, 14–16 November 2012; pp. 413–426.
26. Kara, A.M.; Binsalleeh, H.; Mannan, M.; Youssef, A.; Debbabi, M. Detection of malicious payload distribution channels in DNS. In Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, Australia, 10–14 June 2014; pp. 853–858.
27. RAPID. Available online: https://opendata.rapid7.com/sonar.fdns_v2/ (accessed on 2 February 2020).
28. Xu, W.; Tao, Y.; Guan, X. Experimental Comparison of Free IP Geolocation Services. In *Security with Intelligent Computing and Big-data Services*; SICBS 2018. Advances in Intelligent Systems and Computing; Springer: Cham, Switzerland, 2020; Volume 895.
29. ipplus360. Available online: https://www.ipplus360.com/ (accessed on 2 February 2020).

30. TaoBao IP. Available online: http://ip.taobao.com/service/getIpInfo.php?ip=myip (accessed on 2 February 2020).
31. ip2location. Available online: https://lite.ip2location.com/ (accessed on 2 February 2020).