Mitigating DDoS Attacks towards Top Level Domain Name Service

Lanlan Pan, Xuebiao Yuchi, and Yong Chen
National Engineering Laboratory for Naming and Addressing,
China Internet Network Information Center, Beijing 100190, China
{panlanlan, yuchixuebiao, chenyong}@cnnic.cn

***Abstract*—As the largest country code Top Level Domain (ccTLD) name service, .CN receives billions of queries every day. Under the threat of Distributed Denial-of-Service (DDoS) attacks, effective mechanism for client classification is especially important for such busy ccTLD service. In this paper, by analyzing the query log of .CN name service, we propose a novel client classification method based on client query entropy and global recursive DNS service architecture. By checking with the query frequencies of the clients, we validate the effectiveness of the proposed method on both busy and long-tailed clients. We find that 2.32% clients can cover the most important web spiders, recursive servers, and well-known internet services, etc. The results indicate that, our method can bring significant benefits for creating the client whitelist, which is useful for mitigating DDoS attack towards Top Level Domain (TLD) name service.**

Keywords—Domain; DNS; DDoS; TLD

I. INTRODUCTION

Domain Name System (DNS) is one of the most critical Internet services. Large TLDs such as .CN and .DE receive billions of queries every day. However, DNS is extremely vulnerable to large-scale amplification and reflection DDoS attacks[1][2][3].

Normally, the ability to identify network traffic clients with high accuracy can mitigate DDoS attacks significantly. For example, the defense system doesn't need to return "TC=1" response for forging client validation if the system already knows that the client address is real but not supporting TCP. Therefore, client classification method is critical to improving the efficiency of DDoS defense mechanism towards TLD name service.

II. RELATED WORK

Various methods have been proposed to analyze the characteristics of DNS query traffic.

In [4], the authors point out that TLDs need to serve as many legitimated clients as they can, while attackers can utilize botnets, reflect with open resolvers, or spoof important source IP addresses[5]. In [6], the authors point out that poor traffic visibility and unrestricted access to DNS recursive servers could result in difficult defense on authority servers without harming the normal clients. TLDs need to make long-term analysis and identify data flow more accurately in real network environments.

In [7] [8] [9], the authors focus on learning the different DNS query structure from stub resolver to recursive server, detecting abnormal domains and infected computers. However, as TLDs serve huge numbers of clients in real time, the characteristics to figure out TLD query structure must be simple and robust enough.

In [10], Verisign provides a method for creating a whitelist of trustworthy resolvers. The characteristics to classify resolver include top-talker status, distribution of domain names queried and qtype. However, they did not do further analysis on the clients, also ignored long-tailed resolvers with small DNS query traffic. To solve this problem, we analyze the key characteristics related to TLD query traffic classification, select important clients with their type information to build up a whitelist of trustworthy clients. We also probe global IPv4 addresses, and select large covered recursive DNS from global recursive DNS dataset, add long-tailed small query recursive clients to the whitelist.

III. DNS ARCHITECTURE

As illustrated in Figure 1, domain information is stored by hierarchical authority servers and spread by global recursive servers:

1) Forwarding Recursive DNS (FRS) servers such as Q-B do iterative queries on "Root", ".CN" and "CNNIC.CN", finally get the IP addresses of "WWW.CNNIC.CN ".
2) Caching Recursive DNS (CRS) servers such as C-A and C-B receive queries from Stub DNS (U-A, U-B), then forward the queries to upper CRS (C-A to C-B) or FRS (C-B to Q-B). ".CN" TLD servers receive queries from Q-B, but not directly from Stub DNS.
3) Except for Recursive DNS, there are many other clients which send DNS queries to ".CN" too. Such as web spider clients (Q-S) from Google and Baidu, or mail service clients (Q-M) like Yahoo Email, etc.

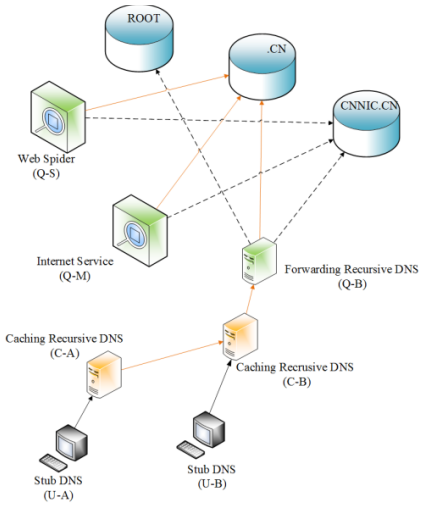

Figure 1. Overview of DNS Operation Process.

IV. Build Classifier and Create Whitelist

In this section, we describe our approach to building client classifier and creating client whitelist, which can be used for TLD DDoS attack defense.

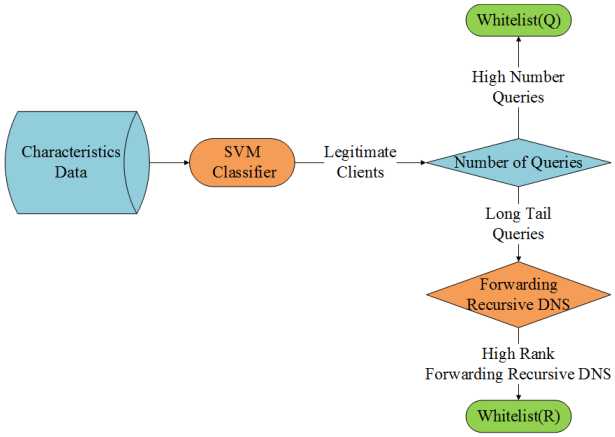

Figure 2. Whitelist Solution.

Figure 2 illustrates the main components of our solution:

1) We collect data and calculate key characteristics to build a linear SVM[11] classifier for client type decision. The classifier will learn from client's behavior to check if each client is legitimate or not.
2) We select legitimate clients with high query times, add to the whitelist. They are pre-classified as Recursive DNS servers, web spiders, or some other internet services, etc.
3) We select legitimate clients with long-tailed query times, which are high-rank FRS, add to the whitelist.

A. Collect Data

We extract *<timestamp, client IP, queried domain, qtype>* data from the query log of .CN. We first configure a wildcard IP address "218.241.111.100" for "*.BS.CNNIC.CN". Then we probe Recursive DNS in the whole IPv4 address space to figure out the map of *<CRS, FRS>*, the steps are as follows:

1) For example, we select one IP address "114.114.114.114", and generate unique temporary domain "909fbe.BS.CNNIC.CN" for it.
2) Probe node PN-1 sends a query to "114.114.114.114", the qname is "PN-1-909fbe.BS.CNNIC.CN" and qtype is "A".
3) If PN-1 can receive the correct response "218.241.111.100", then we record that IP "114.114.114.114" is a CRS.
4) We check the authority query log of "BS.CNNIC.CN", and find that "58.217.249.142" queried domain "PN-1-909fbe.BS.CNNIC.CN".
5) Now we can get a record <Probe: 1, Cache: "114.114.114.114", Forwarding: "58.217.249.142">.

Finally, we collect *<CRS, FRS>* dataset in the whole IPv4 address space.

B. Characteristics

There are two groups of characteristics for the client classification: Query-Log and Recursive-DNS.

Query-Log characteristics: These characteristics are obtained from .CN query log, associated to each client IP.

1) **Number of Query Domains**
2) **Query Times**
3) **Query Times on specific domains**: As Table 1 shows, we select three widespread domains in China: "SINA.COM.CN", "TIANYA.CN" and "360.CN". These domains have the same glue NS TTL on .CN, but different NS TTL on their own authority servers. Recursive DNS servers usually use NS TTL to overwrite glue NS TTL, and probe node clients always query TLD on a fixed interval, while web spider clients usually use glue NS TTL directly. We can figure out the differences based on hot domain query times.
4) **Number of Important Domains**: One domain is marked as important if more than 500 clients have queried it. We count the number of important domains that each client queried, and calculate the important domain cover rate of each client.
5) **Query Times on Important Domains**
6) **Average of Domain Query Entropy**: Similar with HITS [12], counting how many clients have queried the domain can give us a general estimate of this domain's prominence of the whole domain set. We calculate the average of domain query entropy to figure it:

Step1, calculate the query entropy of each domain:

$$P(s_i, d) = \frac{Q(s_i, d)}{Q(d)}$$

$$H(d) = -\sum_{i=1}^{N} P(s_i, d) * log(P(s_i, d))$$

- d : domain name
- $Q(d)$: query times of d
- S : clients which have queried d, $S = \{s_i \mid i = 1, 2, \dots, N\}$
- N : number of clients in S
- $Q(s_i, d)$: the times of client s_i have queried d
- $H(d)$: domain query entropy of d

Step 2, calculate the average of domain query entropy associated to the client:

$$P'(d_i, s) = \frac{Q'(d_i, s)}{Q(s)}$$

$$E(s) = \sum_{i=1}^{M} P'(d_i, s) * H(d_i)$$

- s : client IP
- $Q(s)$: query times of s
- D : domains which have been queried by s, $D = \{d_i \mid i = 1, 2, \dots, M\}$
- M : number of domains in D
- $Q'(d_i, s)$: the times of s has queried d_i
- $E(s)$: the average of s's domain query entropy

Table 1. NS TTL Change on Delegation.

Domain	*glue NS TTL*	*NS TTL*
SINA.COM.CN	*86400*	*86400*
TIANYA.CN	*86400*	*7200*
360.CN	*86400*	*600*

Recursive-DNS characteristics: These characteristics are obtained from the <CRS, FRS> dataset, associated to the same client which is a FRS server.

1) **Number of Caching Recursive DNS**: Big FRS cover large amounts of CRS.
2) **Number of Important Caching Recursive DNS**: Some CRS such as Google Public DNS, OpenDNS, ISP DNS are all serving many users. We collect about 600 important CRS servers in China, and count the number of important CRS servers associated to the FRS servers.
3) **User Number of Caching Recursive DNS serving**
4) **Group by Forwarding Recursive DNS's CIDR/24 address**: We calculate three same characteristics group by FRS's CIDR/24 IP prefix associated to the clients with same CIDR/24 IP prefix. Moreover, we build a directed graph based on the <CRS CIDR/24 IP prefix, FRS CIDR/24 IP prefix> links, then calculate the weighted PageRank [13] of FRS's CIDR/24 IP prefix.

V. Analysis

In this section, we show the work we have done on .CN.

A. Data

As mentioned above in Section IV, we take .CN query log from 2015-10-08 00:00 to 2015-10-08 24:00, which contains 7.07 billion queries and 1.80 million clients.

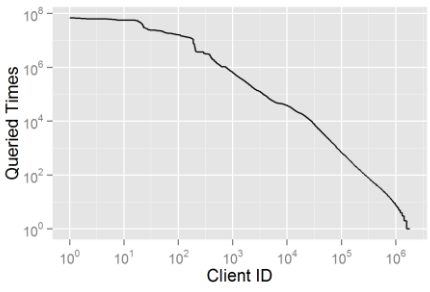

Figure 3. Client Query Times Distribution.

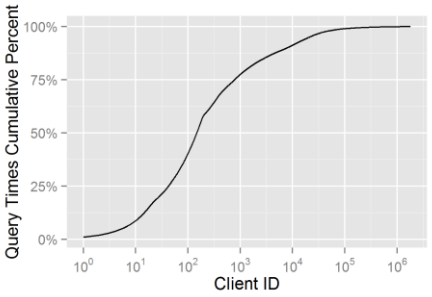

Figure 4. Client Query Times Cumulative Percent.

Clients are sorted by the number of query times that each client queried in descending order, then set the Client ID. Figure 3 and Figure 4 illustrate the long-tailed query distribution of the clients, which indicate that about 30,000 clients send more than 10,000 queries to .CN in one day (Figure 3), the top 30,000 clients totally contribute 96.58% .CN queries (Figure 4), but the tail 1.77 million clients merely contribute 3.42% .CN queries (Figure 4).

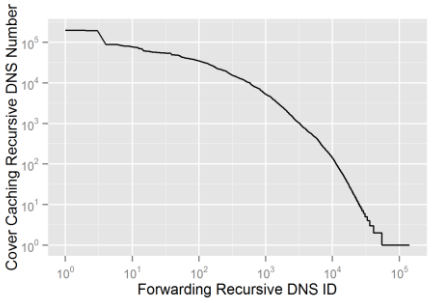

Figure 5. Forwarding Recursive DNS Cover Caching Recursive DNS Number Distribution.

We collect 25.26 million Recursive DNS records in the whole IPv4 address space with 60 probe nodes, which contains 11.42 million CRS servers and 142.8 thousand FRS servers. FRS servers are sorted by the number of CRS servers that each FRS server covered in descending order, and set the FRS ID. As Figure 5 shows, the largest number of CRS that FRS covered is about 200,000. Nearly 11,500 FRS servers cover more than 100 CRS servers.

B. Whitelist

We calculate the key characteristics of each client based on the data mentioned above. Then we manually label 13 thousand clients as training data, they are web spider clients from Google and Baidu, or well-known internet service clients like Yahoo Email, or important FRS servers from top 3 ISP of China (China Telecom, China Unicom, China Mobile) and Public DNS like GoogleDNS, or evil attack clients we have encountered, etc. We select LINEAR kernel function to build the SVM classifier, which achieves 100% accuracy when to predict the training data. We use the whole 1.8 million clients as test data.

Table 2 shows the type classification result on 1.8 million clients of .CN query log. "service_spider" indicates the clients from web spider service. "service" indicates the clients from some internet service like Skype or phishing

domain detector. “not_public_ip” indicates the clients with fake IP address which can be flush or from local area network which we pre-configured. “maybe_evil” indicates the abnormal clients which suddenly send huge number queries to small domains set. “recur”, “recur_public” and “maybe_recur” indicate the FRS servers. The accuracy to identify FRS dataset of the classifier is 99.98%. But as Figure 3 and Figure 4 shows, about 1.8 million clients send less than 10,000 queries per day. These long-tailed clients may be misclassified, we use the global recursive data to refine them.

Table 2. Client Type Classification.

Client_Type	*Client_Number*	*Query_Times*	*Query_Percent*
service_spider	*1112*	*4.49 billion*	*63.54%*
recur	*1.8 million*	*2.02 billion*	*28.59%*
service	*112*	*0.25 billion*	*3.51%*
recur_public	*395*	*0.16 billion*	*2.29%*
maybe_recur	*4324*	*0.13 billion*	*1.82%*
not_public_ip	*53*	*14.31 million*	*0.20%*
maybe_evil	*14*	*3.6 million*	*0.05%*

As detailed in Section IV, we select legitimate clients with a high number of queries and add them to whitelist (Q) with their type information. The whitelist (Q) contains 12,371 clients, covers 92.21% of .CN queries. And we select long-tailed legitimate clients which are high-rank FRS servers and add them to whitelist (R). The whitelist (R) contains 29,564 clients, covers 1.90% of .CN queries. The whole whitelist contains 41,935 clients, 2.32% of total 1.8 million clients, covers 94.10% of .CN queries and 96.19% of total 11.42 million CRS servers we have probed. Table 3 shows the type of the whole client whitelist, and setup different protection levels in massive DDoS attack.

Table 3. Whitelist Client Information.

Client_Type	*Client_Number*	*Query_Times*	*Query_Percent*	*White List*	*Protection_Level*
service_spider	*1108*	*4.49 billion*	*63.54%*	*(Q)*	*High*
service	*93*	*0.25 billion*	*3.50%*	*(Q)*	*Middle*
recur_public	*209*	*0.16 billion*	*2.23%*	*(Q)*	*High*
recur	*9651*	*1.54 billion*	*21.84%*	*(Q)*	*High*
maybe_recur	*1310*	*0.08 billion*	*1.09%*	*(Q)*	*Middle*
recur	*29564*	*0.13 billion*	*1.90%*	*(R)*	*High*

Note that compared with a simple client whitelist, we can defense the DDoS attack more flexible with the client type information and protection level. Our method can reduce the time cost of client identification by the whitelist, and the DDoS attack defense system gains more benefits.

VI. Conclusion

In this paper, we propose a method to build up a client whitelist with type information and protection level, which can be used on .CN. The goal of our work is a deeper understanding of general TLD query traffic, and source client characteristics of DDoS attack.

We use .CN query log and global recursive data to implement our method, and build up the defense whitelist. Our analysis shows that the whitelist covers the most important clients of .CN, which contains both top clients with large query times and long-tailed recursive clients with small query times. Our future work will be concerned with some temporary clients and temporary domains from Recursive DNS log.

References

[1] Pras, Aiko, et al. "DDoS 3.0-How terrorists bring down the internet." International GI/ITG Conference on Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance. Springer International Publishing, 2016.

[2] Marrison, Chris. "Understanding the threats to DNS and how to secure it." Network Security 2015.10 (2015): 8-10.

[3] Zargar, Saman Taghavi, Jyoti Joshi, and David Tipper. "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks." Communications Surveys & Tutorials, IEEE 15.4 (2013): 2046-2069.

[4] ICANN SSAC, "SAC065 - Advisory on DDoS Attacks Leveraging DNS Infrastructure", https://www.icann.org/en/system/files/files/sac-065-en.pdf, 2014.

[5] Ferguson, Paul. "Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing." (2000).

[6] Hudaib, Adam Ali Zare. "DNS Advanced Attacks and Analysis." International Journal of Computer Science and Security (IJCSS) 8.2 (2014): 63.

[7] Shi, Hongbo, and Kazuhiko Iwasaki. "Classification of DNS queries for Anomaly Detection." 2013 IEEE 19th Pacific Rim International Symposium on Dependable Computing (PRDC). IEEE, 2013.

[8] Choi, Hyunsang, and Heejo Lee. "Identifying botnets by capturing group activities in DNS traffic." Computer Networks 56.1 (2012): 20-33.

[9] Yu, Bin, Les Smith, and Mark Threefoot. "Semi-supervised time series modeling for real-time flux domain detection on passive DNS traffic." Machine Learning and Data Mining in Pattern Recognition. Springer International Publishing, 2014. 258-271.

[10] Osterweil, Eric, and Danny Mcpherson. "White listing DNS top-talkers." U.S. Patent No. 8,935,744. 13 Jan. 2015.

[11] Steinwart, Ingo, and Andreas Christmann. Support vector machines. Springer Science & Business Media, 2008.

[12] Kleinberg, Jon M. "Hubs, authorities, and communities." ACM Computing Surveys (CSUR) 31.4es (1999): 5.

[13] Xing, Wenpu, and Ali Ghorbani. "Weighted pagerank algorithm." Communication Networks and Services Research, 2004. Proceedings. Second Annual Conference on. IEEE, 2004.