An Efficient Malicious Domains Detection Deployment Framework on the DNS Recursive Server*

Bin Zhang
Peng Cheng Laboratory, Shen Zhen, China
bin.zhang@pcl.ac.cn

DNS is one of the core and most important components of the Internet. Unfortunately, besides being used for obvious benign purposes, domain names are also popular for malicious use. Traditional detection system employs large-scale, passive DNS analysis techniques on DNS servers to detect domains that are involved in malicious activity. On-line detection deployment is barely impossible due to dealing huge DNS traffic. This work provides an efficient malicious domains detection deployment framework on the DNS Recursive Server, which can reduce the analyzed DNS traffic greatly, thus making the on-line detection possible.

Chinese 360 cloud service providers have served about 10% of DNS requests in China. According to our measurement of the 360 cloud DNS traffic, we find the locality feature of DNS traffic. Fig. 1 shows the hite rate of one hour DNS traffic. For the current minute DNS request, the hit rate is about 55% if its previous one minute DNS traffic is reserved, the hit rate is about 60% if its previous two minutes DNS traffic is reserved, the hit rate keeps increasing with more previous DNS traffic is reserved. The hit rate reach 88% if its previous one hour DNS traffic is reserved. That is, if we store the resolving results of one hour DNS requests in the recursive server, the recursive server just need to proceed about 12% new request from the root server.

Hence, the basic idea of our proposed scheme is to establish a whitelist on the recursive server. The whitelist stores the domain names and corresponding IP addresses of normal DNS requests resolved by the recursive server in the previous period. The vast majority of DNS requests fall into the whitelist of the local recursive server. We deploy a detection system to detect the rest of the DNS request traffic which cannot be resolved locally. The deployment framework is shown in Fig. 2.

The recursive server downstream port can process the user's DNS requests and return the resolution results. The upstream port iteratively queries the unmatched domain names in the whitelist. The concrete detection system only analyze the DNS traffic of the upstream port, greatly decreasing the DNS traffic analyzed of the traditional ways.

The DNS whitelist can be implemented by the RPZ mechanism. RPZ provides a response interception mechanism for DNS servers, matching and filtering specific domain names

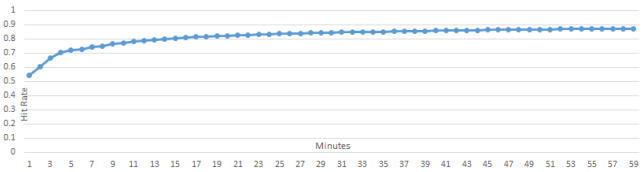

Figure 1: the hit rate of 360 DNS traffic

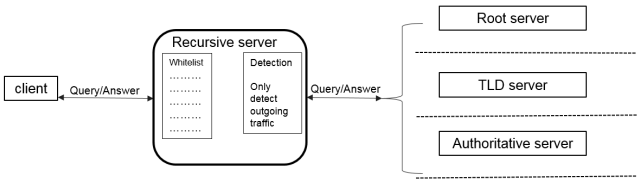

Figure 2: the DNS deployment scheme

or IP addresses for response results, and changing response results. We use RPZ in an opposite way, the whitelist domain names are directly returned to its IP address list in the RPZ zone file.

The unmatched domain names will be further queried from the root server by the upstream port. These traffic will be detected by the anomaly detection system to split the normal DNS record and malicious DNS record. The normal record will be used to update the whitelist in a timely manner.

The RPZ whitelist mechanism is similar but different from the DNS cache server. Firstly, the cache server has limited cache capacity. Secondly, the cache saves all DNS records and is hard to control, the RPZ zone only need to save A and AAAA records. Thirdly, the DNS record will be deleted in cache when the TTL value expires.

The values of our framework lies in:

1) It improves the resolving efficiency and avoids a large number of iterative DNS query requests, which reduces the DNS traffic and avoids the DNS attacks from the upper level.

2) It decreases the analyzed traffic and makes the on-line detection possible for high-speed large recursive servers.

At present, We develop a tool to generate the RPZ zone file using the historical resolving results. We will delpoy different detection methods in our framework to test the efficiency of our proposed scheme in our following work.

*This work was supported by The Major Key Project of PCL (PCL2021A02)