Advanced Topics in IP Networks

Out of bailiwick project

Ron Aharoni - 300028065, Steven Karas - 328620984, Phillip Kaplan - 324478403

Last edited 2019-03-30

Collaboration Statement We worked alone and referred to sources cited inline and in the references section, and well as the course materials.

1 Introduction

1.1 Abstract

DNS in broad strokes works by translating human-friendly textual names into IP addresses that can be used to send traffic to a server. DNS is organized into a hierarchical tree with a single entity responsible for each branch in the tree. For example, the root of the tree is managed by ICANN, and each of the top level domains underneath are managed by various registries.
Glue records are critical to the proper functioning of the DNS system[3]. However, improper use of glue can leave a domain susceptible to various security vulnerabilities. Here we attempt to analyze the use of certain best practices by domain owners on the internet, specifically the prevalence of "out of bailiwick" glue records.

1.2 Related Work

In "The Availability and Security Implications of Glue in the Domain Name System" [6] Zheng Wang discusses the necessity of glue records to the proper functioning of DNS along with the implications of current practices in terms of both availability and security. He presents the ability of known mitigation techniques to prevent certain classes of cache poisoning attacks on DNS. The cache attacks he discusses were originally presented by Dan Kaminsky in his presentation "Blackops2008 – it's the end of the cache as we know it". Wang additionally discusses the implications of DNSSEC and attacks on it, and whether bailiwick checking is relevant if the records are secured.

1.3 Bailiwick definition

A bailiwick for DNS purposes is defined[2] as:

> In-bailiwick:
>
> (a) An adjective to describe a name server whose name is either subordinate to or (rarely) the same as the zone origin. In- bailiwick name servers require glue records in their parent zone (using the first of the definitions of "glue records" in the definition above).
>
> (b) Data for which the server is either authoritative, or else authoritative for an ancestor of the owner name. This sense of the term normally is used when discussing the relevancy of glue records in a response. For example, the server for the parent zone "example.com" might reply with glue records for "ns.child.example.com". Because the "child.example.com" zone is a descendant of the "example.com" zone, the glue records are in- bailiwick.
>
> Out-of-bailiwick: The antonym of in-bailiwick

2 Data

We downloaded the Majestic Million[4] on January 18th, 2019. This is a list of the top one million sites ranked by the number of subnet that link to them, collected and published by Majestic.com.

From these, we constructed a list of 1,000,878 domains and domain prefixes to check for out of bailiwick glue records.

2.1 Data Quality

From some randomized sampling of the domain list, some of the domains are listed because of links to subdomains, whereas others were included despite being subdomains of others in the list (e.g. plus.google.com and google.com both appear on the list). Additionally, not all of the domains returned valid records (e.g. ns1.example.com as their NS record), were valid domains (e.g. http://163), or had any records at all (NXDOMAIN).

To deal with these issues we decided that the correct approach was to do our analysis on zones rather than on domains. Domains that we did not consider to represent valid zones:

- Domains that returned "NX Domain"
- Domains that returned an SOA record (e.g. plus.google.com)

Out of the million initial domains we constructed a list of 1,000,878 domains and domain prefixes by using all of the possible subdomains. We did this with a simple script. Out of the generated domains, 964039 turned out to represent valid zones.

2.2 DNS record collection

We used a BIND server configured as a recursive resolver to query the records for the domain list. We run a small pipeline in parallel[5] to do this quickly.

2.3 Data Collection

DNS queries To perform the DNS queries, we used dig. Dig is a command line tool for querying the domain name system. We used the following command to fetch the name server records for all domains:

```
dig [domain] @127.0.0.1 NS
```

Where the localhost flag instructed dig to use the local resolver. Since we were limited largely by the round trip time of serially sending queries, to speed up the process we used the parallel utility running 20 queries at any given time.

We used a BIND server configured as a recursive resolver to query the records for the domain list. We constructed and sent our queries using dig in a parallel [5] pipeline to quickly gather the data:

```
cat majestic_all_possible_domains | parallel -j 20 -- dig {} @127.0.0.1 NS |
  gzip > dig.output.gz
```

After less than 6 hours all of the queries had been completed and the data stored in a text summary file.

2.4 Data Analysis

using Python3 to parse the zone files and gather statistics. Graphs were plotted using pandas and matplotlib. We set up a short pipeline to collate the unsorted results against the original ranked list:

```
tail -n+2 raw_results.csv | sort -t, -k1 | pv -l | cat > sorted_results.csv
tail -n+2 majestic_million.csv | sort -t, -k3 | pv -l | cat > sorted_majestic.csv
join -t, -o 2.1,0,1.2,1.3,1.4,1.5 -1 1 -2 3 sorted_results.csv sorted_majestic.csv |
  sort -t, -k1 -n | pv -l | cat <(echo $'Majestic Million Rank,Domain,
    Num NS records,Num glue records,Num out-of-bailiwick glue,Num loose-out-bailiwick glue')
      - > collated_results.csv
```

We then used pandas and matplotlib to generate plots showing the CDF of having out-of-bailiwick glue records. The code can be found at Github[1].

3 Evaluation and Discussion

3.1 Results

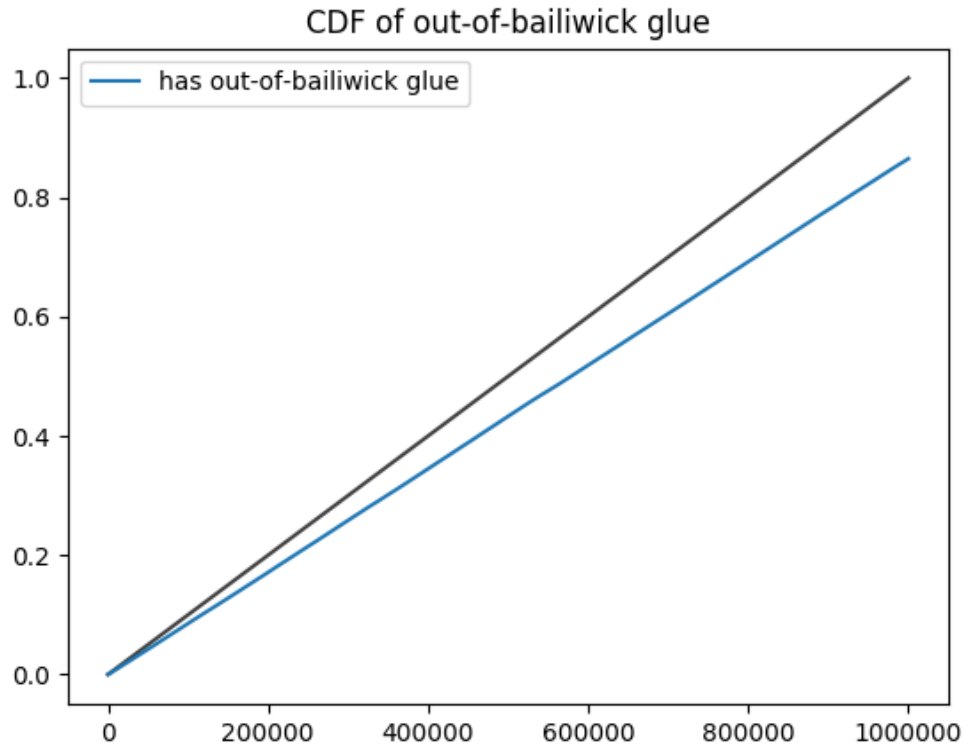
Number of domains	1,000,879
NXDOMAINs	26,678
Number of empty domains	65,708
Number of NS records	2,476,318
Number of glue records	3,156,909
Number of improper glue records	0
Out of bailiwick glue	3,078,764
Loosely out of bailiwick glue	1,600,449

3.2 Prevalence of third party providers

Provider	Domains
ns.cloudflare.com	218,747
domaincontrol.com	129,789
worldnic.com	65,400
myhostadmin.net	61,948
dnspod.net	45,038
hichina.com	43,271
dnsmadeeasy.com	40,619
dns.com	24,208
registrar-servers.com	23,821
googledomains.com	23,705

Table 1: Domain count for 10 most popular providers

3.3 CDF



3.4 Discussion

Our results clearly show that use of out of bailiwick glue is a widespread practice. This appears to be due to the prevalence of name servers being hosted by third party providers such as GoDaddy or Cloudflare. Third party name server hosting is largely a desirable situation, as requiring smaller web hosts to have their own nameservers would add significant relative complexity to their network and require additional expertise, and would therefore provide additional attack surface ripe for exploitation by a determined attacker. The better way to deal with this problem is likely to make use of DNSSEC, which is much more resistant to the types of attacks analyzed in the works we discussed.

4 Conclusion

DNS glue is a critical component of DNS, but has security implications. Certain types of DNS spoofing attacks can be mitigated by servers refusing to cache based on glue for out of bailiwick name servers. However, the vast majority of name servers on the internet are in fact out of bailiwick. This is justifiable since the overhead of hosting a private name server is high. A better way to mitigate these attacks would be widespread adoption of DNSSEC.

References

[1] Networks project. https://github.com/ron-aharoni/networks-project, 2019.

[2] Paul E. Hoffman, Andrew Sullivan, and Kazunori Fujiwara. DNS Terminology. RFC 7719, December 2015.

[3] M. Lottor. Domain administrators operations guide. Internet Requests for Comments, November 1987. http://www.rfc-editor.org/rfc/rfc1033.txt.

[4] Majestic.com. The majestic million report. https://majestic.com/reports/majestic-million, 2019.

[5] O. Tange. Gnu parallel - the command-line power tool. *;login: The USENIX Magazine*, 36(1):42–47, Feb 2011.

[6] Zheng Wang. The Availability and Security Implications of Glue in the Domain Name System. *arXiv e-prints*, page arXiv:1605.01394, May 2016.