

Five Basic Mistakes Not to Make in DNS

by Ron Aitchison

Here are five things you can do to make sure your DNS is in good shape and not causing problems for the rest of the Internet, which, by the way, also includes you.

DNS Is Really, Really Important

Every time we get email, access a web page, make a VoIP call, or complete many other tasks, we use the Domain Name System (DNS). That makes DNS part of the critical infrastructure of the Internet.

This article describes five things that you can do to keep you and your organization safe as well as reduce unnecessary load on the DNS infrastructure:

- * Reverse-Map Private (RFC1918) IP Addresses in Your DNS
- * Ensure That Localhost Is Forward- and Reverse-Mapped
- * Ensure That Your Domain Name Does Not Have a Lame Delegation
- * Ensure That You Are Not Running an Open Recursive Name Server
- * Ensure That Your Email Address Is Correct in the SOA RR

For each of the items discussed, the corrective actions and BIND configuration (named.conf) or zone file fragments are included.

Some DNS Background

DNS is a complex, highly distributed system that operates on the hierarchical structure of domain names, so it is worth briefly covering some background.

In tech speak, DNS resolves a name (domain name) into an IP address using a series of queries to authoritative DNS servers until the final answer, an IP address, is obtained. This process is referred to as forward mapping and is done in units called zones, which correspond to each level of the hierarchy of the domain name. To resolve the name `www.example.com`, a local DNS server will query authoritative DNS servers for each level in the name, beginning with the root-servers, which will return a referral to the authoritative `.com` gTLD servers. The gTLD server, when queried, will return a referral to the authoritative name servers for the domain `example.com`, which finally will return the IP address we require. Sometimes it also useful to be able to start with an IP address and find the name allocated to it; email systems especially use this technique as part of an antispam arsenal. DNS performs IP address-to-name translation by manipulating the IP address and using reverse-mapped zones under the reserved domain name `IN-ADDR.ARPA`.

There are two broad classes of DNS servers:

- * DNS servers that operate on behalf of a group of users, either on a local network or within a larger organization--say, an ISP--and that provide what are called recursive services. These DNS servers will issue queries and follow any referrals to ensure their clients get the required answer. Such DNS servers typically maintain a cache of answers to save them from issuing the same query repeatedly, and for that reason are frequently called caching name servers or recursive name servers, or sometimes just resolvers. DNS records are maintained in the cache for a period determined by the Time to Live value associated with each Resource Record (RR).
- * DNS servers that are either defined to be a master or a slave for the zone. The Resource Records obtained from these servers in answer to a query are marked with a special bit (AA) to indicate they come from an authoritative source. These servers are, not too surprisingly, typically called authoritative name servers.

Now, by way of illustrating the sometimes confusing nature of the DNS, some organizations (especially smaller ones) elect to run both caching and authoritative functions in the same name server. While this is not a recommended practice for reasons that we shall see later, it may be simply a pragmatic solution. Finally, each PC has what is sometimes, but erroneously, called a resolver, which typically caches results. In fact this is almost always a stub-resolver that needs the services of a caching or recursive name server to operate effectively, since it is incapable of following referrals.

The root servers are always the starting point for any new query cycle, which makes them the critical part of the critical infrastructure! There are 13 root-servers named a.root-servers.net through m.root-servers.net. Using anycast techniques, these servers are replicated across the globe. There are now well over 100 instances of the various root-servers in operational use, the majority of which now lie outside North America. The root-servers receive more than 2 billion queries per day, of which (according to some studies [6],) only 2% are legitimate queries! While the vast majority of unnecessary traffic relates to buggy software and badly configured firewalls, a significant proportion was caused by poorly configured DNS software.

So with all this information at our fingertips, let's look at five DNS issues that you should check to minimize unnecessary traffic on the DNS infrastructure, and help keep your organization running smoothly.

Reverse-Map Private (RFC 1918) IP Addresses

Up to 7% of the total traffic arriving at some root servers consists of reverse queries for private IP addresses, and a complete routing infrastructure (AS112) has been constructed just to handle this problem. Private IP addresses are any in the ranges 192.168.0.0/16, 172.16.0.0/12, and 10.0.0.0/8. While ISPs are delegated the task of reverse-mapping public IPs, they have no such responsibility for private IP addresses. If you are running your own local recursive name server, it is your responsibility to make sure these IP addresses are reverse-mapped in your DNS configuration.

To illustrate reverse-mapping a zone, let's assume that we are using a private address range, 192.168.5.0/28 (16 IP addresses only). BIND's named.conf file defines the reverse-mapped zone and should look something like this:

```
// named.conf fragment

zone "5.168.192.IN-ADDR.ARPA" IN {
    type master;
    file "192.168.5.rev";
    allow-update {"none";}
    allow-transfer {192.168.5.6;}; // ip address of zone slave
}
```

The zone filename convention above uses 192.168.5.rev to make it simpler to understand, whereas the zone name must be 5.168.192.IN-ADDR.ARPA. If you really enjoy writing reversed addresses, however, you could use "5.168.192.in-addr.arpa" as the filename. Reverse maps are standard zone files, and may use zone transfers to update the slave (secondary) name server. The allow-transfer statement just limits the source of zone transfer requests to the slave name server. The allow-update statement is precautionary, and should be removed if you are auto-updating from DHCP or another source. The reverse zone fragment would look like this:

```
; reverse map for 192.168.5.0/28
$ORIGIN 5.168.192.IN-ADDR.ARPA.
; Start of Authority (SOA) record defining the key characteristics of the zone
@      IN      SOA    ns1.example.com. hostmaster.example.com. (
                2007040800 ; serial number
                12h       ; refresh
                15m       ; retry
                3w        ; expiry
                2h        ; min = minimum
        )
; name servers Resource Records for the domain
        IN      NS     ns1.example.com.
        IN      NS     ns2.example.com.
; A RRs for name servers are not glue and therefore
; not necessary
; PTR RR maps an IPv4 address to a host name
2      IN      PTR     ns1.example.com.
3      IN      PTR     ns2.example.com.
4      IN      PTR     mail.example.com.
.....
14     IN      PTR     joe.example.com.
```

All the righthand names in this file must be Fully Qualified Domain Names (FQDNs). That is, they must terminate with a dot. If the dot at the end of mail.example.com was omitted, the \$ORIGIN substitution rule would generate a name of mail.example.com.5.168.192.IN-ADDR.ARPA, which was probably not the intended result.

Ensure That Localhost Is Forward- and Reverse-Mapped

Just over 1% of the queries at the root-servers in one of the studies were for localhost. This means that in the study case, which was a subset of the total root-server query load, there were more than 12,000 DNS configurations with no localhost zone defined. Localhost is used by many applications as shorthand when referring to the local PC, and is always mapped to the loopback address 127.0.0.1 (or ::1 for Ipv6). Apart from creating unnecessary root-server traffic, it will slow down applications considerably. To resolve localhost (and its reverse map) the BIND named.conf file should have the following zone files:

```
// named.conf fragment

// forward map zone of localhost
zone "localhost" IN {
    type master;
    file "master.localhost";
    allow-update {"none"};
    allow-transfer {192.168.5.6}; // ip address of zone slave
}
// reverse map zone of IPv4 localhost
zone "0.0.127.IN-ADDR-ARPA" IN {
    type master;
    file "localhost.rev";
    allow-update {"none"};
    allow-transfer {192.168.5.6}; // ip address of zone slave
}
```

The localhost forward-mapped zone file is normally supplied with most BIND distributions and may be called localhost or localhost.zone. The reverse-mapped zone file--again typically supplied with BIND distributions--is normally called something very meaningful like named.local, which may help explain why it was omitted in more than 12,000 configurations. The localhost zone may be treated as shown, with a master/slave configuration (in which case the allow-transfer statement is used to limit the transfer request source,) But in most cases, because the zone file is distributed with BIND, it is more normally defined as a master in all cases.

The forward-mapped zone file supplied with most BIND distributions is a masterpiece of the terse and incomprehensible, and a typical example is shown here:

```
$TTL 86400
$ORIGIN localhost.
@ 1D IN SOA @ hostmaster (
    0 ; serial
    12h ; refresh
    15m ; retry
    1w ; expiry
    3h ; minimum
)
@ 1D IN NS @ ; localhost is the name server
1D IN A 127.0.0.1 ; always returns the loop-back address
```

An alternate zone file format which is functionally identical may be more comprehensible - or then again it may not!

```
$TTL 1d ; 24 hours could have been written as 24h or 84600
$ORIGIN localhost.
localhost. IN SOA localhost. hostmaster.localhost. (
    2007040800 ; serial
    3H ; refresh
    15M ; retry
    1w ; expire
    3h ; minimum
)
localhost. IN NS localhost. ; localhost is the name server
localhost. IN A 127.0.0.1 ; the loop-back address
The reverse-mapped zone file should look like this:
```

```
$TTL 86400 ; 24 hours
$ORIGIN 0.0.127.IN-ADDR.ARPA.
@      IN      SOA      localhost. hostmaster.localhost. (
                                2007040800 ; Serial number
                                3h        ; Refresh
                                15        ; Retry
                                1w        ; Expire
                                3h )    ; Minimum
      IN      NS       localhost.
1      IN      PTR     localhost.
```

Ensure That Your Domain Name Does Not Have a Lame Delegation

Lame delegation means that a name server defined in an NS Resource Record (RR) for the zone does not respond authoritatively. That is, it does not set the AA bit in a query response for the zone. This normally happens for one of two reasons. The zone could have failed to load for some reason, in which case the problem will appear in BIND's log (or you could run the named-checkzone utility to verify the zone file). Alternatively, one or more of the name servers defined in the NS RRs for the domain is not configured with a zone clause. The fragment zone file below shows two name servers for the domain. BIND's named.conf file must have a zone clause for "example.com" at both name servers (the type may be master or slave), otherwise lame delegation will result:

```
; fragment of example.com zone file
$ORIGIN example.com.
; name servers Resource Records for the domain
; the ns1 is inside our domain, ns2.example.net
; is in another domain
; both name servers must be configure with a zone clause
; for example.com in BIND's named.conf
      IN      NS       ns1.example.com.
; out of domain name server
      IN      NS       ns2.example.net.
; normal A RR for the in domain name server
ns1    IN      A       192.168.5.2
```

It is worth a couple of points of clarification here. The master and any slave servers for the domain will respond authoritatively to queries for the domain, so it is not possible to differentiate between a master and slave answer except, possibly, by looking at the SOA RR. A caching name server, which is neither a master nor a slave server for the zone, will respond with the AA bit the first time it reads the DNS Resource Record data from an authoritative name server. If it supplies the RR from the cache, the AA bit will not be set. To state the obvious, caching name servers cannot be authoritative for a domain. Only name servers with a zone clause for the domain (the type may be master or slave) in BIND's named.conf file can be authoritative for that domain.

Lame delegations are not good for two reasons. First, they cause requery of the zone's name servers looking for an authoritative answer, which adds unnecessary network load. Secondly, BIND helpfully logs lame delegation, so you can rapidly become famous for reasons you never really wanted. Avoid lame delegations!

Ensure That You Are Not Running an Open Recursive Name Server

Running an open recursive server--which means that essentially anyone, anywhere can use your name server to perform recursive queries--is a very bad idea. First, it increases the possibility of cache poisoning (the malicious insertion of false DNS data), by issuing more recursive queries than are required. Essentially, every query is a potential source of cache poisoning, so why do more of them than you have to?

Second, and far more important, an Open DNS may be used to amplify Denial of Service attacks by being requested to query the zone under attack. Sadly, as with open mail relays, what used to be a friendly neighbor thing has become a potential source of harm to everyone else on the Internet. Either remove the ability to issue recursive queries completely, or limit the recursive queries to allowed users only. All methods are shown in the following BIND named.conf fragments:

```
// if you are running an authoritative only server
// the following statement should always be present
recursion no;
```

For a caching name server, or a mixed authoritative and caching name server (remember the recommended practice is not to mix the capabilities), use one of the two methods below:

```
// mixed authoritative and caching server
// use an appropriate local address scope statement
// to limit recursion requests to local users
allow-recursion {192.168.2.0/24;};
//
// if you run only a caching name server use this method
// use an appropriate local address scope statement
// to limit all query requests to local users
allow-query {192.168.2.0/24;};
```

Remember: Open DNS = Very Bad Idea. Treat this one as a very high priority, and fix it as soon as possible if you find you are running an Open DNS.

Ensure That Your Email Address Is Correct in the SOA RR

As the saying goes, stuff happens. You may, heaven forbid, end up with a zone that is giving people problems. For everyone's sake, and especially yours, make sure that your email address in the zone is correct and functional. The email address of a responsible person for the zone is defined in the SOA Resource Record (the so-called RNAME field), an example of which is shown below:

```
@           IN           SOA    ns1.example.com. hostmaster.example.com. (
                                2007040800 ; serial number
                                12h         ; refresh
                                15m         ; retry
                                3w          ; expiry
                                2h          ; min = minimum
                                )
```

In the above example, the email address shown is the recommended (in RFC 2142) hostmaster (specified as hostmaster.example.com), but it can be any valid email address that can ensure that you get the mail quickly. This also means you can respond equally quickly after taking any required remedial action, and earning yourself a Netizen of the Year Award.

Let's Give Ourselves Some DNS Headroom

The DNS system has worked well for more than 20 years, and it continues to work in spite of all kinds of unpleasant things thrown at it. Nevertheless, it is in our own self-interest to make sure we all play our part in trying to remove as much of the unpleasant stuff as possible. Because if real trouble arrives, as occasionally it does, the DNS infrastructure will have just a little bit more headroom to let it fix or absorb the problems and keep the Internet rolling along merrily, which is what we all want. A quiet life.

Resources

1. RFC 4697 Observed DNS Resolution Misbehavior (www.ietf.org/rfc.html)
2. ISC OARC (oarc.isc.org)
3. IEPG (www.iepg.org)
4. Cooperative Association for Internet Data Analysis (www.caida.org)
5. Root-server web site (www.root-servers.org)
6. Wow that's a lot of Packets (dns.measurement-factory.com/writings/wessels-pam2003-paper.pdf)

Ron Aitchison is the author of Pro DNS and BIND (Apress, 2005), the first book on DNS to describe the DNSSEC security updates--and in mind-numbing detail.

This article is reprinted by kind permission of O'Reilly Media, Inc.<http://www.oreillynet.com>