ripe-7

Some practical DNS advice

Piet Beertema

hostmaster@NIC.EU.net

12 February 1990

0. Introduction

On the Feb '90 RIPE meeting I reported some problems I had come across in existing nameservers and promised to report them on the mailing list, together with some advice for corrections and some general advice about setting up [the zone files of] a nameserver, and in particular a top level domain nameserver, and managing a domain.

1. SOA records

A problem I've found in quite some nameservers is that the various timers have been set (far) too low. Especially for top level domain nameservers this causes unnecessary traffic over international and intercontinental links.
Unfortunately the examples given in the BIND manual, in RFC's and in some expert documents give those very short timer values, and that's most likely what people have modelled their SOA records after.

First of all a short explanation of the timers used in the SOA record:
- Refresh: The SOA record of the primary server is checked every "refresh" time by the secondary servers; if it has changed, a zone transfer is done.
- Retry: If a secondary server cannot reach the primary server, it tries it again every "retry" time.
- Expire: If for "expire" time the primary server cannot be reached, all information about the zone is invalidated on the secondary servers (i.e. they are no longer authoritative for that zone).
- Default TTL: The default TTL value for all records in the zone file; a different TTL value may be given explicitly in a record when necessary. (This timer is named "Minimum" in most examples, but that name is highly confusing).

For top level domain servers I would recommend the following values:

```
      86400 ; Refresh     24 hours
       7200 ; Retry        2 hours
    2592000 ; Expire      30 days
     345600 ; Default TTL  4 days
```

For other servers I would suggest:

```
      28800 ; Refresh     8 hours
       7200       ; Retry       2 hours
     604800 ; Expire      7 days
      86400 ; Default TTL 1 day
```

but here the frequency of changes, the required speed of propagation, the reachability of the primary server etc. play a role in optimizing the timer values.

2. Glue records

Quite often do people put unnecessary glue (A) records in their zone files. Even worse is that I've even seen *wrong* glue records for an external host in a primary zone file! Glue records need only be in a zone file if the server host is within the zone and there is no A record for that host elsewhere in the zone file.
A patch for BIND 4.8.3 recently issued by Andrew Partan of UUnet tackles the problem of wrong glue records entering secondary servers by masking out glue records that don't belong to the zone file being pulled in on a zone trasfer. This patch has proved to be very helpful. A patch to mask out bad information on outgoing zone transfers should also be applied, but though currently there is no recommended source for this.

3. "secondary server surprise"

I've seen it happen on various occasions that hosts got bombarded by nameserver requests without knowing why. On investigation it turned out then that such a host was supposed to (i.e. the information was in the root servers) run secondary for some domain (or reverse (in-addr.arpa)) domain, without that host's nameserver manager having been asked or even been told so!
Note that NIC.DDN.MIL accepts requests for in-addr.arpa servers WITHOUT checking if secondary servers have been set up, informed, or asked.

4. "MX records surprise"

In a sense similar to point 3. Sometimes nameserver managers enter MX records in their zone files that point to external hosts, without first asking or even informing the systems managers of those external hosts. This has to be fought out between the nameserver manager and the systems managers involved. Only as a last resort, if really nothing helps to get the offending records removed, can the systems manager turn to the naming authority of the domain above the offending domain to get the problem sorted out.

5. "Name extension suprise"

Sometimes one encounters weird names, which appear to be an external name extended with a local domain. This is caused by forgetting to terminate a name with a dot: names in zone files that don't end with a dot are always expanded with the name of the current zone (the domain that the zone file stands for or the last $ORIGIN).
Example: zone file for foo.xx:

```
pqr       MX 100 relay.yy.
xyz       MX 100 relay.yy          (no trailing dot!)
```

When fully written out this stands for:

```
pqr.foo.xx.      MX 100 relay.yy.
xyz.foo.xx.      MX 100 relay.yy.foo.xx.   (name extension!)
```

6. Missing secondary servers

It is required that there be a least 2 nameservers for a domain. For obvious reasons the nameservers for top level domains need to be very well reachable from all over the Internet. This implies that there must be more than just 2 of them; besides, most of the (secondary) servers should be placed at "strategic" locations, e.g. close to a point where international and/or intercontinental lines come together.
To keep things manageable, there shouldn't be too many servers for a domain either.
Important aspect in selecting the location of primary and secondary servers are reliability (network, host) and expedient contacts: in case

of problems, changes/fixes must be carried out quickly.
It should be considered logical that primary servers for European top level domains should run on a host in Europe. For each top level domain there should be 2 secondary servers in Europe and 2 in the USA (there may of course be more on either side).
EUnet has offered to run secondary server for each European top level domain on mcsun.EU.net.

7. Wildcard MX records

Wildcard MX records should be avoided where possible. They often cause confusion and errors: especially beginning nameserver managers tend to overlook the fact that a host/domain listed with ANY type of record in a zone file is NOT covered by an overall wildcard MX record in that zone; this goes not only for simple domain/host names, but also for names that cover one or more domains. Take the following example in zone foo.bar:

```
*         MX 100 mailhost
pqr       MX 100 mailhost
abc.def   MX 100 mailhost
```

This makes pqr.foo.bar, def.foo.bar and abd.def.foo.bar valid domains, but the wildcard MX record covers NONE of them, nor anything below them.
To cover everything by MX records, the required entries are:

```
*         MX 100 mailhost
pqr       MX 100 mailhost
*.pqr     MX 100 mailhost
abc.def   MX 100 mailhost
*.def     MX 100 mailhost
*.abc.def MX 100 mailhost
```

An overall wildcard MX record is almost never useful.

In particular the zone file of a top level domain should NEVER contain only an overall wildcard MX record (*.XX). The effect of such a wildcard MX record can be that mail is unnecessarily sent across possibly expensive links, only to fail at the destination or gateway that the record points to. Top level domain zone files should explicitly list at least all the officially registered primary subdomains.

Whereas overall wildcard MX records should be avoided, wildcard MX records can be acceptable as explicit part of subdomains entries, provided they are allowed under a given subdomain (to be determined by the naming authority for that domain).
Example:

```
foo.xx.   MX 100 gateway.xx.
          MX 200 fallback.yy.
*.foo.xx. MX 100 gateway.xx.
          MX 200 fallback.yy.
```

8. Safety measures & specialties

Nameservers and resolvers aren't flawless. Bogus queries should be kept from being forwarded to the root servers, since they'll only lead to unnecessary intercontinental traffic. Known bogus queries that can easily be dealt with locally are queries for 0 and broadcast addresses. To catch such queries, every nameserver should run primary for the 0.in-addr.arpa and 255.in-addr.arpa zones; the zone files need only contain a SOA and an NS record.
Also each nameserver should run primary for 0.0.127.in-addr.arpa; that zone file should contain a SOA and NS record and an entry:

```
1   PTR    localhost
```

People maintaining zone files with the Serial number given in dotted decimal notation (e.g. when SCCS is used to maintain the files) should beware of a bug in all BIND versions: if the serial number is in Release.Version (dotted decimal) notation, then it is virtually impossible to change to a higher

release: because of the wrong way that notation is turned into an integer, it results in a serial number that is LOWER than that of the former release. It can be done, but ONLY if all secondary servers for the domain are fully restarted (nameserver killed; secondary file removed; nameserver restarted).

Very old versions of DNS resolver code also have a bug that causes queries for A records with domain names like "192.16.184.3" to go out. This happens when users type in IP addresses and the resolver code does not catch this case before sending out a DNS query. This problem has been fixed in all resolver implementations known to us but if it still pops up it is very serious because all those queries will go to the root servers looking for top level domains like "3" etc. In the nameserver code available from mcsun there is a patch #ifdef'ed YPKLUDGE which returns authoritative NXDOMAIN errors to such queries.

Running secondary nameserver off another secondary nameserver is not a good idea: it is known to have led to problems like bogus TTL values. This can be caused by older or flawed implementations, but secondary nameservers in principle should always transfer their zones from the official primary nameserver.

9. Some general points

The domain name system & nameserver is a purely technical tool, not meant in any way to exert control or impose politics. The function of a naming authority is that of a clearing house. Anyone registering a subdomain under a particular (top level) domain becomes naming authority and therewith the sole responsible for that subdomain. Requests to enter MX or NS records concerning such a subdomain therefore always MUST be honoured by the registrar of the next higher domain.
Examples of practices that are not allowed are:
- imposing specific mail routing (MX records) when registering a subdomain.
- making registration of a subdomain dependent on to the use of certain networks or services.

Of course there are obvious cases where a naming authority can refuse to register a particular subdomain and can require a proposed name to be changed in order to get it registered (think of DEC trying to register a domain IBM.XX).
There are also cases were one has to probe the authority of the person: sending in the application - not every systems manager should be able to register a domain name for a whole university.
The naming authority can impose certain extra rules as long as they don't violate of conflict with the rights and interest of the registrars of subdomains; a top level domain registrar may e.g. require that there be primary subdomain "ac" and "co" only and that subdomains be registered under those primary subdomains.
The naming authority can also interfere in exceptional cases like the one mentioned in point 4, e.g. by temporarily removing a domains's entry from the nameserver zone files; this of course should be done only with extreme care and only as a last resort.

When adding NS records for subdomains, top level domain nameserver managers should realise that the people setting up the nameserver for a subdomain often are rather inexperienced and can make mistakes that can easily lead to the subdomain becoming completely unreachable or that can cause unnecessary DNS traffic (see point 1). It is therefore highly recommended that, prior to entering such an NS record, the (top level) nameserver manager does a couple of sanity checks on the new nameserver (SOA record & timers OK?, MX records present where needed? No obvious errors made? Listed secondary servers operational?). Things that cannot be caught though by such checks are:
- resolvers set up to use external hosts as nameservers
- nameservers set up to use external hosts as forwarders

without permission from those hosts.
Care should also be taken when registering 2-letter subdomains. Although

this is allowed, an implication is that abbreviated addressing (see RFC822, par. 6.2.2) is not possible in and under that subdomain. When requested to register such a domain, one should always notify the people of this consequence. As an example take the name "cs", which is commonly used for Computer Science departments: it is also the name of the top level domain for Czecho-Slovakia, so within the domain cs.foo.bar the user@host.cs is ambiguous in that in can denote both a user on the host host.cs.foo.bar and a user on the host "host" in Czecho-Slovakia.