DNS Session 3: Configuration of Authoritative Nameservice

Joe Abley

AfNOG 2012, Serekunda, The Gambia
Recap

- DNS is a distributed database
- Resolver asks Cache for information
- Cache traverses the DNS delegation tree to find Authoritative nameserver which has the information requested
- Bad configuration of authoritative servers can result in broken domains
DNS Replication

- For every domain, we need more than one authoritative nameserver with the same information (RFC 2182)
- Data is entered in one server (Master) and replicated to the others (Slave(s))
- Outside world cannot tell the difference between master and slave
  - NS records are returned in random order for equal load sharing
- Used to be called "primary" and "secondary"
Slaves connect to Master to retrieve copy of zone data

- The master does not "push" data to the slaves
When does replication take place?

- Slaves poll the master periodically - called the "Refresh Interval" - to check for new data
  - Originally this was the only mechanism
- With new software, master can also notify the slaves when the data changes
  - Results in quicker updates
- The notification is unreliable (e.g. network might lose a packet) so we still need checks at the Refresh Interval
Serial Numbers

- Every zone file has a Serial Number
- Slave will only copy data when this number **INCREASES**
  - Periodic UDP query to check Serial Number
  - If increased, TCP transfer of zone data
- It is your responsibility to increase the serial number after every change, otherwise slaves and master will be inconsistent
Recommended serial number format: YYYYYMMDDDNN

- YYYY = year
- MM = month (01-12)
- DD = day (01-31)
- NN = number of changes today (00-99)
  - e.g. if you change the file on 23rd April 2007, the serial number will be 2008052700. If you change it again on the same day, it will be 2008052701.
Serial Numbers: Danger 1

- If you ever decrease the serial number, the slaves will never update again until the serial number goes above its previous value.
- RFC1912 section 3.1 explains a method to fix this problem.
- At worst, you can contact all your slaves and get them to delete their copy of the zone data.
Serial Numbers: Danger 2

- Serial no. is a 32-bit unsigned number
- Range: 0 to 4,294,967,295
- Any value larger than this is silently truncated
- e.g. 200805270000 (note extra digit)
  = 4ACE48698 (hex)
  = ACE48698 (32 bits)
  = 2900657816
- If you make this mistake, then later correct it, the serial number will have decreased
Configuration of Master

- `/etc/namedb/named.conf` points to `zone file` (manually created) containing your RRs

- Choose a logical place to keep them
  - e.g. `/etc/namedb/master/tiscali.co.uk`
  - or `/etc/namedb/master/uk.co.tiscali`

```plaintext
zone "example.com" {
    type master;
    file "master/example.com";
    allow-transfer { 192.188.58.126;
                     192.188.58.2; }
};
```
Configuration of Slave

- named.conf points to IP address of master and location where zone file should be created
- Zone files are transferred automatically
- Don't touch them!

```bash
zone "example.com" {
  type slave;
  masters { 192.188.58.126; };
  file "slave/example.com";
  allow-transfer { none; };
};
```
Master and Slave

- It's perfectly OK for one server to be Master for some zones and Slave for others.
- That's why we recommend keeping the files in different directories:
  - `/etc/namedb/master/`
  - `/etc/namedb/slave/`
    - (also, the slave directory can have appropriate permissions so that the daemon can create files)
allow-transfer { ... }

- Remote machines can request a transfer of the entire zone contents
- By default, this is permitted to anyone
- Better to restrict this
- You can set a global default, and override this for each zone if required

```bash
options {
    allow-transfer { 127.0.0.1; };
};
```
Structure of a zone file

- **Global options**
  - `$TTL 1d`
  - Sets the default TTL for all other records

- **SOA RR**
  - "Start Of Authority"
  - Housekeeping information for the zone

- **NS RRs**
  - List all the nameservers for the zone, master and slaves

- **Other RRs**
  - The actual data you wish to publish
## Format of a Resource Record

<table>
<thead>
<tr>
<th>Domain</th>
<th>TTL</th>
<th>Class</th>
<th>Type</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>www</td>
<td>3600</td>
<td>IN</td>
<td>A</td>
<td>212.74.112.80</td>
</tr>
</tbody>
</table>

- One per line (except SOA can extend over several lines)
- If you omit the Domain Name, it is the same as the previous line
- TTL shortcuts: e.g. 60s, 30m, 4h, 1w2d
- If you omit the TTL, uses the $TTL default value
- If you omit the Class, it defaults to IN
- Type and Data cannot be omitted
- Comments start with SEMICOLON (;)
Shortcuts

- If the Domain Name does not end in a dot, the zone's own domain ("origin") is appended
- A Domain Name of "@" means the origin itself
- e.g. in zone file for example.com:
  - @ means example.com.
If you write this...

```
$TTL 1d
@                       SOA ( ... )
NS  ns0
NS  ns0.as9105.net.
; Main webserver
www                     A   212.74.112.80
MX  10 mail
```

... it becomes this

```
example.com.  86400 IN  SOA ( ... )
example.com.  86400 IN  NS  ns0.example.com.
example.com.  86400 IN  NS  ns0.as9105.net.
www.example.com. 86400 IN  A   212.74.112.80
www.example.com. 86400 IN  MX  10 mail.example.com.
```
Format of the SOA record

$TTL 1d

@ 1h IN SOA ns1.example.net. joe.pooh.org. ( 2004030300 ; Serial 8h ; Refresh 1h ; Retry 4w ; Expire 1h ) ; Negative

IN NS ns1.example.net.
IN NS ns2.example.net.
IN NS ns1.othernetwork.com.
Format of the SOA record

- ns1.example.net.
  - hostname of master nameserver
- joe.pooh.org.
  - E-mail address of responsible person, with "@" changed to dot, and trailing dot

- Serial number
- Refresh interval
  - How often Slave checks serial number on Master
- Retry interval
  - How often Slave checks serial number if the Master did not respond
Format of the SOA record (cont)

- Expiry time
  - If the slave is unable to contact the master for this period of time, it will delete its copy of the zone data

- Negative / Minimum
  - Old software used this as a minimum value of the TTL
  - Now it is used for negative caching: indicates how long a cache may store the non-existence of a RR

- RIPE-203 has recommended values
  - http://www.ripe.net/ripe/docs/dns-soa.html
Format of NS records

- List all authoritative nameservers for the zone - master and slave(s)
- Must point to HOSTNAME not IP address

```
$TTL 1d

@ 1h IN SOA ns1.example.net. joe.pooh.org. ( 2004030300 ; Serial
8h ; Refresh
1h ; Retry
4w ; Expire
1h ) ; Negative

IN NS ns1.example.net.
IN NS ns2.example.net.
IN NS ns1.othernetwork.com.
```
Format of other RRs

- **IN A**  1.2.3.4
- **IN MX**  10 mailhost.example.com.
  - The number is a "preference value". Mail is delivered to the lowest-number MX first
  - Must point to HOSTNAME not IP address
- **IN CNAME**  host.example.com.
- **IN PTR**  host.example.com.
- **IN TXT**  "any text you like"
When you have added or changed a zone file:

- Remember to increase the serial number!
- `named-checkzone example.com \ /etc/namedb/master/example.com`
  - bind 9 feature
  - reports zone file syntax errors; correct them!
- `named-checkconf`
  - reports errors in named.conf
- `rndc reload`
  - or: `rndc reload example.com`
- `tail /var/log/messages`
These checks are ESSENTIAL

- If you have an error in named.conf or a zone file, named may continue to run but will not be authoritative for the bad zone(s)
- You will be lame for the zone without realising it
- Slaves will not be able to contact the master
- Eventually (e.g. 4 weeks later) the slaves will expire the zone
- Your domain will stop working
Other checks you can do

- **dig +norec @x.x.x.x example.com. soa**
  - Check the AA flag
  - Repeat for the master and all the slaves
  - Check the serial numbers match

- **dig @x.x.x.x example.com. axfr**
  - "Authority Transfer"
  - Requests a full copy of the zone contents over TCP, as slaves do to master
  - This will only work from IP addresses listed in the allow-transfer {...} section
So now you have working authoritative nameservers!

- But none of this will work until you have delegation from the domain above

- That is, they put in NS records for your domain, pointing at your nameservers

- You have also put NS records within the zone file

- The two sets should match
Any questions?
TOP TEN ERRORS in authoritative nameservers

- All operators of auth nameservers should read RFC 1912
  - Common DNS Operational and Configuration Errors
- And also RFC 2182
  - Selection and Operation of Secondary DNS servers
1. Serial number errors

- Forgot to increment serial number
- Incremented serial number, then decremented it
- Used serial number greater than $2^{32}$

Impact:
- Slaves do not update
- Master and slaves have inconsistent data
- Caches will sometimes get the new data and sometimes old - intermittent problem
2. Comments in zone files starting ' #' instead of ';' 

- Syntax error in zone file
- Master is no longer authoritative for the zone
- Slaves cannot check SOA
- Slaves eventually expire the zone, and your domain stops working entirely
- Use "named-checkzone"
- Use "tail /var/log/messages"
3. Other syntax errors in zone files

- e.g. omitting the preference value from MX records
- Same impact
4. Missing the trailing dot

; zone example.com.
@ IN MX 10 mailhost.example.com

becomes

@ IN MX 10 mailhost.example.com.example.com.

; zone 2.0.192.in-addr.arpa.
1 IN PTR host.example.com

becomes

1 IN PTR host.example.com.2.0.192.in-addr.arpa.
5. NS or MX records pointing to IP addresses

- They must point to hostnames, not IP addresses
- Unfortunately, a few mail servers do accept IP addresses in MX records, so you may not see a problem with all remote sites
6. Slave cannot transfer zone from master

- Access restricted by allow-transfer {...} and slave not listed
- Or IP filters not configured correctly
- Slave will be lame (non-authoritative)
7. Lame delegation

- You cannot just list any nameserver in NS records for your domain
- You must get agreement from the nameserver operator, and they must configure it as a slave for your zone
- At best: slower DNS resolution and lack of resilience
- At worst: intermittent failures to resolve your domain
8. No delegation at all

- You can configure "example.com" on your nameservers but the outside world will not send requests to them until you have delegation.

- The problem is hidden if your nameserver is acting both as your cache and as authoritative nameserver.

- Your own clients can resolve www.example.com, but the rest of the world cannot.
9. Out-of-date glue records

- See later
10. Not managing TTL correctly during changes

- e.g. if you have a 24 hour TTL, and you swing www.example.com to point to a new server, then there will be an extended period when some users hit one machine and some hit the other

- Follow the procedure:
  - Reduce TTL to 10 minutes
  - Wait at least 24 hours
  - Make the change
  - Put the TTL back to 24 hours
Practical

- Create a new domain
- Set up master and slave nameservice
- Obtain delegation from the domain above
- Test it