DNS Zones

Networks Administration

Otago Polytechnic
Dunedin, New Zealand

First, some review

Last time we turned on BIND and saw how it could serve as a recursive resolver. We didn't have to do any additional configuration for this to work. But how is it configured?

The configuration

file: /var/named/etc/named.conf

```
acl clients {
    localnets;
    ::1;
};

options {
   listen-on    { any; };
   listen-on-v6 { any; };
   allow-recursion { clients; };
};
```

DNS Zones

- Recall that last time we saw how the DNS hierarchy can be viewed as a tree.
- Each node on that tree is a *DNS Zone*.
- A zone is composed of a set of *Resource Records* of various types.
- Information about a particular zone is kept in a *Zone File*. These files conform to a standard format[1].

[1] RFC 1034 and RFC 1035

Zone declarations

In order to have BIND load a zone it must be declared.
file: /var/named/etc/named.conf

```
zone "foo.org.nz" {
    type master;
    file "master/foo.org.nz";
}
```

Slave zone declarations

Slave zones get their zone information from a master.
file: /var/named/etc/named.conf

```
zone "foo.org.nz" {
    type slave;
    masters { 10.10.1.5; };
    file "master/foo.org.nz";
}
```

Zone file names

- There is no particular requirement for how zone files are named.
- Best practice is to indicate the domain name for the zone in the filename, e.g., `example.com`.

Time To Live (TTL)

- We start a zone (in BIND 9) by specifying its *TTL*, e.g.,
  - `$TTL 3h`
  - `$TTL 1d`
  - `$TTL 1w`
- The TTL specifies the length of time for which our zone data should be cached.
- A high TTL saves load on our servers, but it means that changes will take more time to propagate.

Statement of Authority (SOA)

The SOA states that this server is an *authoritative* source of informatiion about our zone.

```
example.com. IN SOA ns1.example.com. tech.somedomain.net. (
     20140821092215 ; serial number
     3h             ; slave refresh
     1h             ; slave retry
     3d             ; slave expires
     1h )           ; negative ttl
```

Nameserver (NS) records

NS records identify the authoritative name servers for our zone

```
example.com.  IN NS ns1.example.com.
example.com.  IN NS ns2.example.com.
example.com.  IN NS ns.otherdomain.com.
```

Address (A) records

A records map host names to IP addresses.

```
fred.example.com. IN A  123.220.44.91

;a host with two addresses
barney.example.com. IN A 71.44.116.17
barney.example.com. IN A 123.211.16.100

;A records can point to the same address as other A records
ws1.example.com. IN A 71.44.116.17
ws2.example.com. IN A 123.211.16.100
```

Alias (CNAME) records

A CNAME record creates an alias for another hostname

```
dino.example.com. IN CNAME fred.example.com.
```

Mail Exchange (MX) records

MX records identify servers that receive mail for our domain

```
example.com. IN MX 10  wilma.example.com.
example.com. IN MX 20  betty.bedrock.org.
```

Text (TXT) Records

TXT records let us put public comments is a zone.

```
example.com. IN TXT "v=spf1 +mx a:colo.example.com/28 -all"
```

TXT records are used, among other things, for the Sender Policy Framework (SPF) [2]

[2] https://tools.ietf.org/html/rfc7208

Reverse zone files

Recall that we use the in-addr.arpa domain to support reverse DNS lookups. This requires another zone file
File: db.192.168.10

```
$TTL 3h
10.168.192.in-addr.arpa. IN SOA ns2.example.com. tec.sdn.net
 ... SOA stuff ... )
    10.168.192.in-addr.arpa.  IN NS ns1.example.com.
    10.168.192.in-addr.arpa.  IN NS ns2.example.com.

    1.10.168.192.in-addr.arpa.  IN PTR pebbles.example.com.
    2.10.168.192.in-addr.arpa.  IN PTR slate.example.com.
    41.10.168.192.in-addr.arpa. IN PTR bambam.rubble.com.
```