



DNS

Domain Name System



Basically, DNS is...

- ▶ A way/server/service to translate IP addresses to human-friendly names. For example: development.mycompany.com is much more readable and intuitive than 223.123.96.35. It also provides some clue about the role of this machine.
- ▶ A distributed system. That is, DNS data is scattered among a large number of servers. Each one manages the records of its own network. For example, the DNS of mycompany.com will have records of all the hosts using this domain (any *.mycompany.com). A DNS server will answer the requests of clients that need to resolve names of its hosts, and it will also query other DNS servers if it doesn't have this information yet.
- ▶ Called BIND (developed by ISC) in Linux systems, which is the most popular one. However, there are other Linux DNS products like NSD and Ubound. Other implementations exist from other vendors like Microsoft DNS server.



Resource Records

- These are the records making up the database of the DNS
- They are stored in *forward* and *reverse* files. The first is used to map hostnames to IP addresses while the second does the opposite.
- The forward file is named after the domain name. For example `mycompany.com`. While the reverse file is named after the IP address like `213.123.96.35.rev`
- In the forward file, each record contains:
 - Hostname
 - Record type
 - IP address (or some other value)
- If the hostname in a line is similar to the previous one, it can be omitted
- In the reverse file, the hostnames are written to be associated with their IP addresses

DNS hierarchy and delegation

- A domain address typically consists of two or more fragments separated by dots. The rightmost fragment is called the top level domain (tld). For example .com, .net, .edu, and .org are all TLD names.
- A domain address *should* always end in a period (.), which represents the root DNS: the first server to query in the tree, but it is conventionally omitted.
- Assuming that there are no cached results, if I want to query the IP address of mail.google.com for example, the following happens in sequence:
 - My machine will ask its local DNS server about this address.
 - The local DNS will ask the root domain (.), which will send a *referral* to .com server.
 - .com responds with the google's DNS server (may be NS1.GOOGLE.COM)
 - NS1.GOOGLE.COM responds with an *answer* containing the IP address, may be 209.85.139.9
 - The local DNS server caches the result and also the data for .com and google.com
- You can use `dig +trace` command to view the process as it happens



Caching



- Caching refers to DNS servers storing the results of their queries so that they don't have to repeat them when a similar resolve request is made.
- The DNS server caches data in its own database for an amount of time called time to live (TTL). This is specified by the owner of the record (for example NS1.GOOGLE.COM).
- The TTL is usually between 1 hour and 1 day, but they could be more or less depending on the configuration. The longer the TTL the less network traffic caused by DNS queries.
- If DNS query fails, for example because of network problems, unresponsive host, missing data...etc, the server makes a *negative* caching TTL. That is, it will not re-attempt to make the same request until the caching period passes.



DNS for load balancing

- You can place several records for the same host
- When the DNS is asked for a host that has several IP addresses, it responds with an randomly ordered list. For example, a heavy website may have its resource records arranged as follows:

```
www      IN  A   172.25.12.1  
          IN  A   172.25.12.2  
          IN  A   172.25.12.3
```

- The second or third IP in this may be served first in subsequent requests
- This technique is used by major websites especially search engines.



The client side

- You define the IP address of local DNS server by editing the the file `/etc/resolv.conf`
- The file is arranged as follows:
 - `search/domain domain1 domain2...`
 - `nameserver IP; name`
 - `nameserver IP; name`
 - `nameserver IP; name`
- The first line specifies which domain to append to a non-fully-qualified hostname. For example, `sales` would be interpreted as `sales.mycompany.com`. You can use `search` or `domain` keywords interchangeably.
- The `nameserver` refers to the DNS IP address, optionally followed by the hostname. For example `nameserver 172.25.22.10; NS1`. You can add up to three nameservers
- Servers are contacted in order. When a timeout occurs (about 5 seconds), the next server is used.



DNS server types

- Master (primary): it stores the zone records locally. A zone refers to the part of the domain address managed by the a given DNS.
- Slave (secondary): obtains its records from the master server. A master server must have at least one slave. A machine can act both as a master of its own zone and a slave for other zones.
- Caching-only: it is not authoritative for any zones. It uses information accumulates over time from queries made to other servers. Its main use is for networks that needs to make DNS requests but do not domain name of their own
- A master or a slave server provides an authoritative answer for their own zones.
- A caching-only server provides a non-authoritative answer to queries. A master or slave server will provide the same answer for zones they do not own.



Recursive or nonrecursive?

- A name server that does not have the answer to the query may either make a request on behalf of the requesting client (recursive server) or provide a referral to a server that *may* have the answer. The client makes a request to this referred-to server. Any server added to `/etc/resolve.conf` must be recursive; as the resolver does not understand referral responses.
- A server that follows referrals will add the information gained through its journey until it finds the answer to its cache, which speeds up subsequent requests.
- Of course TLD domain servers are and should always be nonrecursive. Imagine .com name server caching all requests made to all domains ending in .com!
- Making a publicly-accessible name server nonrecursive is a security measure that prevents “cache poisoning attacks”



Resource record types



- A zone file contains resource records that the DNS server uses to map hostnames to IP addresses and vice versa
- Each record is represented by a line. The following are the most commonly used record types:
 - A: address – it holds the IP address of a host
 - PTR: pointer – holds the hostname of an IP address
 - NS: name server – specifies a list of DNS servers that are authoritative for this zone (delegates them)
 - MX: mail exchange – holds IP address of the mail server for the zone. For example, when an e-mail is sent to somebody@mycompany.com, the DNS provides the client with the mail server responsible for dealing with this e-mail message, which might be mail.mycompany.com


LAB: install a caching server

- The caching server is used to query other DNS servers and cache the results to speed up future similar requests.
- Ensure that the server is configured to use a static IP address
- Open `/etc/named.conf` and ensure that the following lines are present and uncommented:

```
listen-on port 53 { 127.0.0.1; any; };
allow-query      { localhost; any; };
allow-query-cache { localhost; any; };
recursion yes
forwarders {
    8.8.8.8;
};
```
- Ensure that `/etc/named` is still owned by `root` and group-owned by `named` after the changes
- On the client side, add the IP of your DNS server to `/etc/resolve.conf`
- Test caching by issuing a `dig` command to any website, observe the query time and issue the same request again, the query time should decrease. For example `dig facebook.com`

LAB: install a master DNS server

- Target: install a primary DNS server for the domain `linuxadmin.dev`. The server IP would be `192.168.0.252`, the IP of the slave server would be `192.168.0.253`
- In `/etc/named.conf`, ensure that the following lines are present and uncommented:
 - `listen-on port 53 { 127.0.0.1; 192.168.0.252; };`
 - `allow-transfer { localhost; 192.168.0.253; };`
 - `recursion no;`
 - `zone "linuxadmin.dev" IN {
type master;
file "linuxadmin.dev.fwd.zone";
allow-update { none; };
};`
 - `zone "0.168.192.in-addr.arpa" IN {
type master;
file "linuxadmin.dev.rev.zone";
allow-update { none; };
};`

- 
- Create the zone file `/var/named/linuxadmin.dev.fwd.zone` to be as follows:

```
$TTL 86400
```

```
@    IN SOA  ns1.linuxadmin.dev. root.linuxadmin.dev. (
                                2016012129 ; serial
                                3600       ; refresh
                                1800       ; retry
                                604800    ; expire
                                86400)    ; minimum
```

```
@    IN NS   ns1.linxadmin.dev.
```

```
ns1  IN  A    192.168.0.252
```

```
ns2  IN  A    192.168.0.253
```

```
www  IN  A    192.168.0.253
```

```
development IN A    192.168.0.252
```

- Create the reverse zone file `/var/named/linuxadmin.dev.rev.zone` file to look as follows:

```
$TTL 86400
```

```
@      IN      SOA      ns1.linuxadmin.dev. root.linuxadmin.dev. (
                                                201601131601      ; serial
                                                3600              ; refresh
                                                1800              ; retry
                                                604800            ; expire
                                                86400             ; minimum

@      IN      A        192.168.0.252
@      IN      A        192.168.0.253
252    IN      PTR      ns1.linuxadmin.dev.
253    IN      PTR      ns2.linuxadmin.dev.
252    IN      PTR      development.linuxadmin.dev.
253    IN      PTR      www.linuxadmin.dev.
```

- Ensure that both zone files are owned by root and group-owned by named
- Check your configuration using

```
named-checkzone ns1 /var/named/linuxadmin.dev.fwd.zone
named-checkzone ns1 /var/named/linuxadmin.dev.rev.zone
```
- Restart named services
- Ensure that the DNS is working by using `dig` and `nslookup` commands

LAB: Configure DNS slave server

- This server will have the IP address 192.168.0.253 and will be the slave server for the DNS server created in the previous lab
- Install the bind packages using yum or apt-get package managers
- In the /etc/named.conf, ensure that the following lines are present and are uncommented:

```
listen-on port 53 { 127.0.0.1; 192.168.0.253};
allow-query { localhost; 192.168.0.0/24; };
recursion no;
zone "linuxadmin.dev" IN {
    type slave;
    file "slaves/linuxadmin.fwd.zone";
    masters { 192.168.0.252; }; };
zone "0.168.192.in-addr.arpa" IN {
    type slave;
    file "slaves/linuxadmin.rev.zone";
    masters { 192.168.0.252; };
};
```
- You don't need to edit the zone files as they will be automatically copied from the master server.
- After a while take a look at the contents of the zone files to ensure that they have been populated with zone data from the master server.