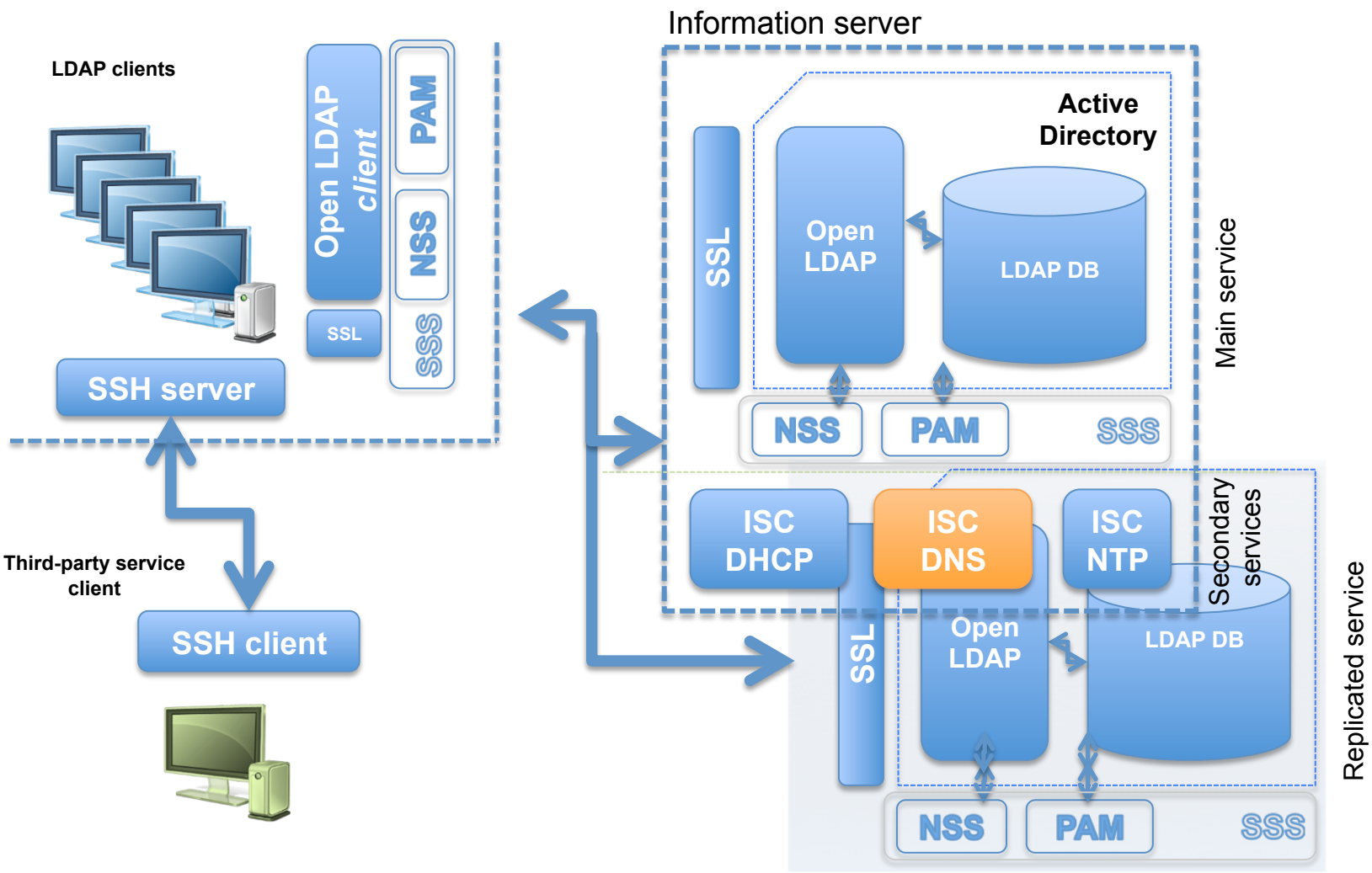
Computer System Design and Administration

Topic 5. Network naming translation service: ISC DNS

José Ángel Herrero Velasco

Department of Computer and
Electrical Engineering

Secure information service: Puzzle



Target: …server convergence

- Installation, configuration and deployment of ***third-party*** **network services** for local *networking* management on the INTRANET:
  - Dynamic configuration service (DHCP): *ISC dhcpd.*
  - Domain name service (DNS): ISC bind9:
    - **Network naming translation:**
      - **IP ←→ domain name.**
      - **…and more.**
  - Network time service (NTP): ISC ntpd.

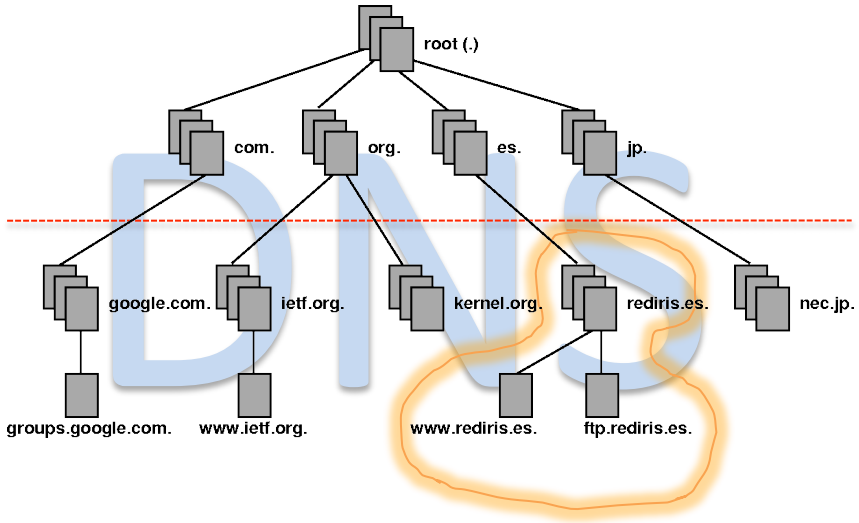
DNS: Internet domain name service

- There are 2 *key* systems (pieces) on the **Internet:**
  - **Domain naming** system (DNS).
  - **Internet routing** system (Routing).
- Usually both systems are **NOT greatly** considered in network environments:
  - We assume that others "*above*" are in charge → ISPs.
- However, they are two **critical** aspects for our subnet (network):
  - The subnet might become *inoperative* because of **link cuts:**
    - *Hosts* won't be able to use DNS.
  - We won't even be able to use a simple network service such as *ssh* or "browse" on web if router failures happen.
- There are 2 main mechanisms for naming resolution:
  - **Local** → `/etc/hosts`.
    - …very **limited.**
  - **Global** → (3party) **DNS servers.**

→ Both of them can be used (together).

DNS: Motivation

- There are currently millions of computers (hosts) interconnected on the Internet:
  - Originally, every host had to know each other's IP/hostname:
    - `/etc/hosts` (synchronized).
  - In the early 80's, the number of hosts interconnected on the *pre*-Internet (ARPANET – TCP/IP) became enormous…:
    - That issue was causing continuous updating all the time.
- High **locality in space and time** for DNS translations:
  - [IP ←→ Hostname].
- Solution:
  - **Distribute** *hierarchically* this information throughout the whole network (**tree**):
    - Huge distributed DB.
    - **Decentralized** translations.
  - Every node of that tree has a **group of servers** which manages the domain translation for that node.



DNS: Roles

- The DNS service has actually many **jobs** (roles or tasks):
  - The main roles are:
    - **To translate between hostname and IP address:**
      - **"FOWARD" DNS lookups:**
        - » Hostname *(FQHN[1])* → Network address *(IP).*
      - **"REVERSE" DNS lookups:**
        - » Network address *(IP)* → Hostname *(FQHN[1]).*
    - **To play an essential role in routing of email, web server access and other internet services:**
      - **From a hostname (FQDN), it returns the IP(s) of the domain mail server.**
      - **It helps to route the mail data.**
  - Other roles:
    - Management of publics keys in asymmetric cryptography and **mailing validation (SPF).**
    - It can deploy **load balance** and **quality of service (QoS)** mechanisms:
      - **Google clusters.**
      - **"On line" newspapers.**
      - **Load balance for LDAP requests on replication environments (N-way master/master).**

    → One *servername* ←→ Several **IPs:**
      - **Each DNS request will be answered with an IP/hostname:**
        - » Geographical location.
        - » Computational load.
      - **`$ host -t a www.elpais.es`:**
        - » Run that command repeatedly…:
          - Load balance algorithm → "Round robin".

[1] **F**ully **Q**ualified **H**ost **N**ame

DNS: Some keys…

- Conceptually, a system administrator may define a ***sub-domain:***
  - → Groups of *hosts.*
  - – Geographical, organizational (company) reasons.
- Once created, it is possible **to assign** naming management to a **"dedicated server":**
  - – Then, the **authoritative unit** is created:
    - That sub-domain and its owner are managed separately.
- This unit is called **"zone":**
  - – The name of a **zone** will be the same as its subnet **domain name:**
    - Sequence of concatenated names.
    - P.e: *www.unican.es*.
  - – Each **dedicated server** has information on every *host* in its domain:
    - Unless these hosts belong to sub-domain (sub-zone) and it had been **delegated:**
      - – **P.e: *ce.unican.es.***
  - – Each zone is usually managed by one **DNS server** (at least):
    - **Authoritative zone** → ***authoritative*** server.
      - – **If more than one server exists, then the DNS service can provide HA and LB to clients:**
        - » Primary and secondary server…
  - – DNS servers may manage more than one zone simultaneously.

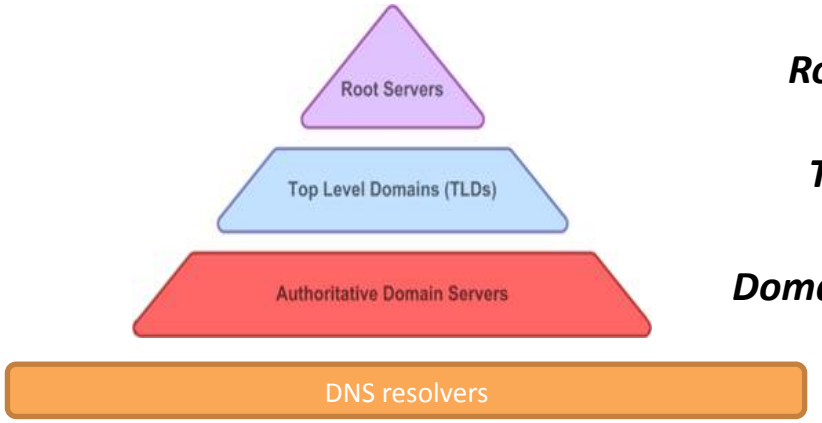
DNS: DNS hierarchy on the Internet

→The **DNS service** on the Internet is composed of a suite of DNS servers. This suite has a **hierarchical tree structure**:

- → **Domain name space**:
  - Each level (leaf) has:
    - A **label.**
    - 0 or more **RR** (Resource Records).

- ***Root*** *zone:*
  - **Root level** servers:
    - `a.root.servers.net`…
- **Top level domain (TLD)** *zones:*
  - **Top Level Domain** servers:
    - **ccTLD** (country/international code).
    - **gTLD** (generic).
    - → http://www.iana.org/domains/root/servers.
- **Domain/sub-domain** *zones:*
  - ***Authoritative/delegated*** **servers.**

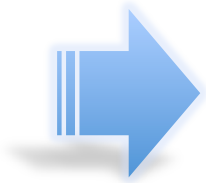
- **DNS resolvers:**
  - **The client side.**



DNS: Server types (Administrative servers)

- **Root** Servers (http://www.iana.org/domains/root/servers):
  - There are **13 Root DNS servers** on the Internet…:
    - To be exact, there are **many more:**
      - **9 of them are multiple distributed servers *(anycast).***
    - Names:
      `letra.root-server.net:`
      - **Letters: A … M.**
  - They hold a list of **TOP LEVEL domain** servers for the top-level domains on the Internet:
    - It contains around 1058 TLDs:
      - **.es, .org, .com…**
    - → E.g.: TLD servers for **.es** generic domain:

```
es.     70262   IN      NS      f.nic.es.
es.     70262   IN      NS      ns15.communitydns.net.
es.     70262   IN      NS      sns-pb.isc.org.
es.     70262   IN      NS      ns1.cesca.es.
es.     70262   IN      NS      ns3.nic.fr.
es.     70262   IN      NS      ns-ext.nic.cl.
es.     70262   IN      NS      a.nic.es.
```



Letra	Dirección IPv4	Dirección IPv6	Número de sistema autónomo[2]	Nombre antiguo	Operador	Ubicación #sitios (global/local)[3]	Software
A	198.41.0.4	2001:503:ba3e::2:30	AS26415	ns.internic.net	Verisign	distribuido (anycast) 4/0	BIND
B	192.228.79.201 (desde Enero 2004; originalmente era 128.9.0.107)[4]	2001:478:65::53 (no en la zona raíz todavía)	AS4	ns1.isi.edu	USC-ISI	Marina Del Rey, California, U.S. 1/0	BIND
C	192.33.4.12	2001:500:2::c (no en la zona raíz todavía)	AS2149	c.psi.net	Cogent Communications	distribuido (anycast) 8/0	BIND
D	199.7.91.13 (desde el 3 de enero de 2013; originalmente era 128.8.10.90)[5]	2001:500:2d::d	AS27	terp.umd.edu	Universidad de Maryland	College Park, Maryland, U.S. 1/0	BIND
E	192.203.230.10	No disponible	AS297	ns.nasa.gov	NASA	Mountain View, California, U.S. 1/11	BIND
F	192.5.5.241	2001:500:2f::f	AS3557	ns.isc.org	Internet Systems Consortium	distribuido (anycast) 4/51	BIND 9[6]
G	192.112.36.4	No disponible	AS5927	ns.nic.ddn.mil	Defense Information Systems Agency	distribuido (anycast) 6/0	BIND
H	128.63.2.53	2001:500:1::803f:235	AS13	aos.arl.army.mil	U.S. Army Research Lab	Aberdeen Proving Ground, Maryland, U.S. 2/0	NSD
I	192.36.148.17	2001:7fe::53	AS29216	nic.nordu.net	Netnod (antes Autonómica)	distribuido (anycast) 41/0	BIND
J	192.58.128.30 (desde Noviembre 2002; originalmente era 198.41.0.10)	2001:503:c27::2:30	AS26415		Verisign	distribuido (anycast) 62/13	BIND
K	193.0.14.129	2001:7fd::1	AS25152		RIPE NCC	distribuido (anycast) 5/12	NSD[7]
L	199.7.83.42 (desde Noviembre 2007; originalmente era 198.32.64.12)[8]	2001:500:3::42	AS20144		ICANN	distribuido (anycast) 130/0	NSD[9]
M	202.12.27.33	2001:dc3::35	AS7500		Proyecto WIDE	distribuido (anycast) 4/1	BIND

- **Top-Level domain (TLD)** Servers (http://www.iana.org/domains/root/db/es.html):
  - They manage information about **AUTHORITATIVE domain servers** for a particular domain (**zone**):
    - They are installed in the root zone of a **name space.**
  - For each DNS request, they return a *list of authoritative domain servers:*
    - For that zone.
- **Authoritative/delegated domain** Servers:
  - They hold the **naming information** [IP ←→ FQDN] for translation of a **zone.**
  - They give **answers** in response to DNS requests from DNS clients.
  - **Also, a part of the zone (sub-zone) can be delegated → Transfer zone:**
    - **Then, the authoritative domain** server contains the IPs of delegated servers for that particular sub-domain.

Roots servers → ccTLD and gTLD

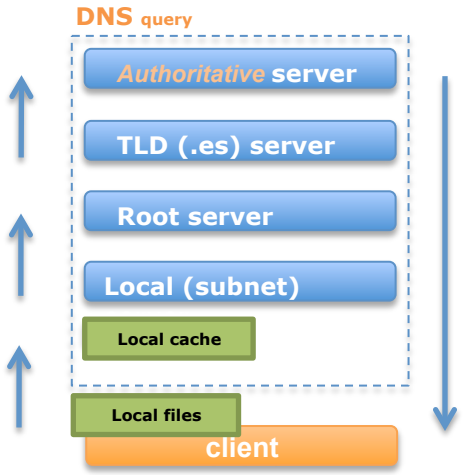
Source: www.root-servers.org.

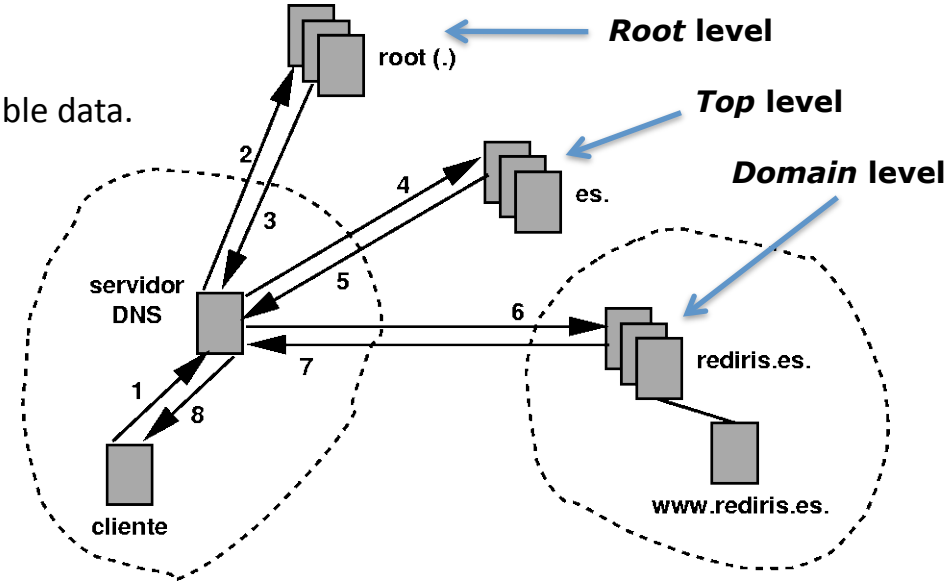
DNS: Server types (operational servers)

- **Primary** servers:
  - Store data of a subnet (domain) **within their local files.**
  - Provide their own **authority services** to respond to translation queries.
- **Secondary** servers (**slave mode**):
  - Contain copies of DNS data:
    - **It is usually obtained from synchronization directly with the master server.**
  - Provide:
    - **Improved performance.**
    - **Backup mechanisms.**
- **Recursive or cache** servers (**cache mode**):
  - They use the same software as primary/secondary DNS servers.
  - They **cache** the query results in RAM memory (no disk):
    - **When an address query is done, this server resends it to a primary/secondary domain server which answers the client.**
    - **Then, this server caches the result into memory (RAM):**
      - » This data has an **expiration time**: TTL.
    - **If clients asks the same query, the "cache server" answers itself.**
- **Forwarding** servers:
  - They only act as *"relay"*. They don't hold any data!!
  - Resend EVERY query to registered servers.

DNS: Operation; How does it work?

- DNS works over **UDP** protocol (port 53).
- The lookups **are delegated** (up…) → "Recursive" query:
  - Each DNS server only contains the data of its **immediately lower level.**
- DNS server *administrative* types:
  - ***Roots* servers:**
    - The rest of DNS servers on the Internet know them.
  - ***TLD (Top Level Domain)* servers:**
    - They hold information about other DNS servers (**zones/sub-zones**).
    - Internationalized domains → **ccTLD:**
      - **.es, .ti, .fr… → ISO-3166.**
    - Generic domains → **gTLD:**
      - **.com, .net, .org…**
  - ***Authoritative* domain servers:**
    - Manage their own "zones" → 100% reliable data.





BIND9: A DNS protocol deployment

- **BIND**: ***B**erkeley **I**nternet **N**ame **D**omain:*
  - Client/server architecture:
    - Server: **BIND.**
    - Client: **Resolver (OS internals):**
      - **Clients send queries about hostname/IP.**
      - **Server answers…**
  - Implementation initially developed at California University (Berkley) in the late 80's:
    - UNIX BSD 4.3.
    - Nowadays, it is maintained by the *Internet Systems Consortium.*
  - It is the most widely used implementation on Internet (Unix/**Linux**).
- Versions:
  - BIND v4 (1985):
    - 4.9 y 4.9.1 → Developed by "Digital" ← HP.
    - 4.9.2 → Developed by "Vixie Enterprises".
    - 4.9.3 onwards → ISC (Internet Software Consortium).
  - BIND v8 (1997).
  - BIND v9 (2000): Fully rewritten.

BIND9: A DNS protocol deployment

- Main features:
  - BSD license.
  - Important safety aspects.
  - DNSSEC, TSIG, Nsupdat:
    - .NS; it is one of the weakest points of the Internet:
      - **Typical attack: *DNS cache poisoning.***
      - **Soporte IPv6.**
  - Protocol improvements:
    - IXFR, DDNS, Notify, EDNS0.
  - Multiprocessor support.
  - DNS service:
    - A stand-alone *daemon* listens on **TCP/UDP port 53.**
- Software and current version of BIND 9:
  - ISC BIND, version 9.10.2…

BIND9: Service installation (ISC DNS)

Source: www.isc.org.

- **Server** & **tools** installation

  BIND9 (Server):

```
$ apt-get update
$ apt-get install bind9
$ apt-get install bind9-doc host dnsutils whois

$ dpkg-reconfigure bind9

$ update-rc.d bind9 defaults
```

Review of the default configuration.

- **Lab 2:** we will deploy a **primary** (local) **and cache DNS server.**

BIND9: Service main configuration

https://ftp.isc.org/isc/bind9/cur/9.11/doc/arm/Bv9ARM.ch06.html

- ***Service*** **primary** configuration files (Main entries):
  - Modular architecture:
    - Well segmented and organized in a group of files.
  - The bind9 configuration is based on a set of **clauses** and **statements:**
    - **Clauses** group together statements:
      - **`(acl, controls, include, logging, options, zone,…).`**
    - **Statements** are defined by { }:
      - **They control the functionality and security of the BIND server.**
    - It can contain comments too → (`//` and `/**/`).

`$ vi /etc/bind/named.conf:`

→ **`Include`** statement:

- **It integrates the configuration from 3 different files:**
  - `/etc/bind/`**`named.conf.options:`**
    - Service Configuration Options and Settings.
  - `/etc/bind/`**`named.conf.local:`**
    - Local zones definition.
  - `/etc/bind/`**`named.conf.default-zones:`**
    - Default *root* zone definition.
- **The format (*formal grammars*) of these files is the same.**

BIND9: Service main configuration

- **`zone`** statement
  - → **`zone zone_name [class] { … };`**
  - **– Defines the specific zones that your DNS server will support:**
    - » Zone type: it defines the *server role* for that zone:
      - Master.
      - Slave.
      - Hint (root).
      - Forward.
      - …
    - » File: PATH to the configuration zone file:
      - Files which define the "**forward**" lookups:
        - *[FQDN → IP address].*
        - One file per authority **domain/zone.**
      - Files which define the **"reverse"** lookups:
        - *[IP address → FQDN].*
        - One file per authority **IP range.**
      - These statements are included in:
        - `/etc/bind/`**`named.conf.local`**`.`
    - » And more options…

BIND9: Service main configuration

- More options:
  - **`acl`** statement
    - → `acl acl-name { … };`
    - » Access control statement.
    - » It defines:
      - The host groups for which access is allowed/denied.
      - Their **access modes.**
  - **`options`** statement
    - → `options {[]…[]};`
    - » Defines a large number of **global options** to be used by BIND:
      - `$ man named.conf.`
  - **`logging`** statement
    - → `logging { ... }`
    - » Configures a wide variety of logging options for the name server:
      - Format options and **severity levels.**
  - …

BIND9: Specific options configuration

- Configuration of **particular options** in **DNS service** (main entries):

  `$ vi /etc/bind/named.conf.options.`

  - It establishes **extra options:**
  - `options` statement:
    - Defines a large number of **global options** to be used by BIND.
    - We can use it to defile the work directory PATH (named) and more and more...
    - `forward:`
      - **Specifies the forwarding behavior.**
      - **`first` → DNS cache.**
      - **`only` → DNS forwarding.**
    - `forwarders:`
      - **Specifies the IP addresses list to be used for forwarding:**
        - » DNS server where the DNS requests must be sent.

DNS cache/ forwarding

BIND9: DNS zone definition (default)

- Configuration of *root* zones → **Root-servers**:

```
$ vi /etc/bind/named.conf.default-zones.
```

- Keeps the same syntax and grammar as `named.conf`:
  - Defines the **"Root" zones:**
    - Also specifies the one dedicated to *localhost.*
  - **Definition only:**
    - **Server type:**
      - » Type **hint** for root servers.
      - » Type **master** for *localdomain* zone and *broadcast*.
    - **Root zone files:**
  - It does NOT hold neither local zone definitions nor **zone data:**
    - That is in a group of a *particular zone*:
      - **`/etc/bind/db.root:`**
        - » The same file for every DNS server.
      - **`/etc/bind/db.0.`**
      - **`/etc/bind/db.255.`**
      - **`/etc/bind/db.local.`**
      - **`/etc/bind/db.127.`**

Don't touch these files!
Keep as default

BIND9: DNS zone definition (local)

- Configuration of *local* zones → **subnet (intranet):**

```
$ vi /etc/bind/named.conf.local.
```

- Keeps the same syntax and grammar as `named.conf`:
  - Defines and describes the **"local" zones.**
  - **Definition only:**
    - **Server type:**
      - » Type hint for root servers.
      - » Type master for *localdomain* zone.
    - **Root zone files.**
  - It does NOT hold either local zone definitions or **zone data:**
    - That is in a group of a *particular zone*:
      - **`/etc/bind/<name_zone>.zone.`**
      - **`/etc/bind/<IP-domain_zone>.zone.`**

BIND9: DNS zone configuration (local)

- **Zone files (Forward** DNS lookup):

  `$ vi /etc/bind/<name_zone>.zone:`

  - These files hold the **local domain (zone) data** to do the forward lookups:
    - [FQDN→ IP address].
  - **Resource Record** (types):
    - **SOA** (Start of Authority): identifies the start of a zone of authority:
      - **Serial number (to record information updates).**
      - **Times for retry and update of information.**
      - **Expiration time.**
      - **Minimum TTL (time-to-live).**
    - **CNAME** – Identifies the canonical name of an *alias:*
      - **hostname → alias_hostname:**
        - `server-01 IN CNAME www.`
    - **MX**, LOC, SRV, TXT.
    - **A** – A host address (In the IN class, this is a 32-bit IP address):
      - **[FQDN → IP address]:**
        - `server-01 IN A 172.16.118.11.` → **Forward** DNS lookup.

BIND9: DNS zone configuration (local)

- **Zone files (Reverse** DNS lookup):

  `$ vi /etc/bind/<IP-domain_zone>.zone`:

  - These files hold the **local domain (zone) data** to do the reverse lookups:
    - [IP address → FQDN].
  - **Resource Record** (types):
    - Common RRs as previous slide.
    - **PTR** – A pointer to another part of the domain name space:
      - **[IP address→ FQDN]:**

```
» 232 IN PTR server-01.localdomain.
```

BIND9: DNS zone configuration (default)

- **Zone** files(**default**):
  `$ vi /etc/bind/db.root`:
  - This file holds the information on **root name servers:**
    - **Root DNS ("." zone)** for everyone:
      - **TOP levels.**
      - **Needed to initialize cache of Internet domain name servers.**
    - Root servers can change → File should be updated.
- `$ vi /etc/bind/db.0`:
  - It holds:
    - BIND data file for **broadcast** zone.
- `$ vi /etc/bind/db.255`:
  - It holds:
    - BIND reverse data file for **broadcast** zone.
- `$ vi /etc/bind/db.local`:
  - It holds:
    - BIND data file for local **loopback** interface:
      - **Forward DNS lookup.**
- `$ vi /etc/bind/db.127`:
  - It holds:
    - BIND reverse data file for local **loopback** interface:
      - **Reverse DNS lookup.**

These files are made available by InterNIC

root

broadcast

localhost

Configuration files included in Bind **as default.** (Don't touch!!!).

BIND9: Daemon main configuration

- **DNS *daemon*** (server) **options** configuration:

  `$ vi /etc/default/bind9`:

  - It holds variables for bind9 server parametrization:
    - **`named`** daemon.
  - Those (variables) are used by the startup **service script:**
    - `/etc/init.d/bind9`.
  - Variables:
    - `OPTIONS='-u bind'`:
      - **Run resolvconf?**
    - `RESOLVCONF=yes`:
      - **Startup options for the server.**

Instances: Cache primary server

Sample

- `/etc/bind/named.conf`:

```
include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
```

As usual, it only includes other DNS configuration files.

Instances: Cache primary server

Sample

- /etc/bind/named.conf.default-zones:

```
zone "." {
        type hint;
        file "/etc/bind/db.root";
};

zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};
zone "127.in-addr.arpa" {
        type master;
        file "/etc/bind/db.127";
};
zone "0.in-addr.arpa" {
        type master;
        file "/etc/bind/db.0";
};
zone "255.in-addr.arpa" {
        type master;
        file "/etc/bind/db.255";
};
```

Root-servers file.

Instances: Cache primary server

Sample

- /etc/bind/named.conf.local:

```
/* direct & reverse lookups of localdomain: */

zone "localdomain" {
        type master;
        file "/etc/bind/localdomain";
        allow-query { any; };
};

zone "0.168.192.in-addr.arpa" {
       type master;
       file "/etc/bind/db.192.168.0";
};
```

Zone/domain name.

Zone defined as primary (authoritative).

ASCII file where we define the zone.

Everyone is allowed.

Instances: Cache primary server

Sample

- /etc/bind/named.conf.options:

```
options {
   directory "/var/cache/bind";

        forward first;
        forwarders {
                8.8.8.8;
                8.8.4.4;
        };

   auth-nxdomain no;    # conform to RFC1035
   listen-on-v6 { any; };
};
```

Directory in which we can define "secondary" zones.

DNS server IPs from the upper level to DNS request resolution

Instances: Cache primary server

Sample

- /etc/bind/localdomain.zone:

```
; BIND file for localdomain zone
;
$TTL    604800
@       IN      SOA     localdomain. server-01.localdomain. (
                        1       ; Serial
                          86400         ; Refresh
                           7200         ; Retry
                        1209600         ; Expire
                          10800 )       ; Negative Cache TTL
;
localdomain.    IN      NS      server-01.
localdomain.    IN      MX      20          mail-01.
;
server-01   IN      A       192.168.0.11
server-02   IN      A       192.168.0.12
server-03   IN      A       192.168.0.13
server-04   IN      A       192.168.0.14
client      IN      A       192.168.0.20
mail-01     IN      CNAME   server-04
```

Forward DNS lookups.

Instances: Cache primary server

Sample

- /etc/bind/db.192.168.0:

```
; BIND reverse data file for local loopback interface
;
$TTL  604800
@  IN   SOA localdomain. server-01.localdomain. (
           1     ; Serial
      604800     ; Refresh
       86400     ; Retry
     2419200     ; Expire
      604800 )   ; Negative Cache TTL
;
@   IN  NS  localdomain.

11              IN      PTR     server-01.localdomain.
12              IN      PTR     server-02.localdomain.
13              IN      PTR     server-03.localdomain.
14              IN      PTR     server-04.localdomain.
20              IN      PTR     cliente.localdomain.
```

Reverse DNS lookups.

Resolver: The DNS Client

- **Resolver:**
  - It is a suite of routines in the **C library** that provide access to the Internet Domain Name System (DNS).

```
$ vi /etc/resolv.conf
```

  - **Resolver** configuration file.
  - It holds:
    - `search:`
      - **The domain names that must be used *automatically:***
        - » The FQDN is determined from the hostname and the domain search path.
    - `nameserver:`
      - **IP address of the DNS servers that will be consulted:**
        - » One line, one server.
        - » Descending order search.

DNS: More Client configuration

`$ vi /etc/hosts`

- For local use only!:
  - It is the **first** "DB" that is consulted (*)...
  - If a DNS request is resolved by it, the DNS request ends.
- For our local network, always keep updated with FQDN/IPs of our **local servers** at least**.**

`$ vi /etc/nsswitch.conf`

- (*) This file configures **the order** in which the resolution/translation systems will be used.
- **NSS.**

DND: Checking

$ named-checkconf, named-checkzone
- Check the BIND configuration.

$ whois <domain> #client/server
- Obtains information about the domain.
- The **whois** service is required.
- alias whois='whois -h <whoisServer>'

$ dig <name/IP> [server]
- Obtains DNS records:

```
# dig @server-01 . ns
```

- Several parameters can be accessed:
  - Server, type of record, translation, etc…

$ **nslookup** <name/IP> [server]-