DNS

FICHEROS HOST

En los orígenes de internet, cuando había pocas máquinas conectadas, el problema de traducir un nombre a una IP se resolvió de un modo muy sencillo: mediante un fichero que asociaba cada dirección IP al nombre del equipo. Este fichero, que sigue existiendo actualmente, se llama hosts y se encuentra en distintas localizaciones según el sistema operativo:

Sistema Operativo	Localización
GNU/Linux - Unix	`/etc/hosts`
Windows XP / 2003 / Vista / 7 / 8 / 10	`C:\Windows\System32\drivers\etc\hosts`
Mac OS	`/private/etc/hosts`

En un principio el fichero hosts se podía crear a partir de una base de datos oficial de hosts que se mantenía en el Network Information Control Center (NIC), aunque eran necesarios frecuentemente cambios locales para poner al día el fichero respecto a alias no oficiales y/o hosts desconocidos. Su sintaxis es muy sencilla:

```
127.0.0.1       localhost
192.168.1.1     router.asir.com   router
192.168.1.10    www.asir.com      www
```

Fichero NSSWITCH

El mecanismo de resolución de nombres en GNU/Linux es modular y puede utilizar varias fuentes de información declaradas en el archivo /etc/nsswitch.conf (Name Server Switch). Este fichero nos permite buscar cierto tipo de información administrativa (hosts, passwd, group, shadow, networks, etc.), especificando qué fuentes queremos comprobar (qué bases de datos) y en qué orden se harán estas comprobaciones.

```
# cat /etc/nsswitch.conf
passwd:         compat
group:          compat
shadow:         compat
gshadow:        files

hosts:          files mdns4_minimal [NOTFOUND=return] dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

NSSWITCH

files: indica que se busque el nombre del dominio en el fichero /etc/hosts.

mdns4_minimal [NOTFOUND=return]: especifica que se debe usar mDNS (Multicast DNS) para encontrar el nombre del dominio. Este protocolo permite saber en todo momento la relación entre la dirección IP y el nombre de una máquina dentro de la red local. Cuando se utiliza mDNS, en lugar de encargar la resolución de nombres a un servidor DNS, esta labor se distribuye, y cada equipo se encarga de la resolución de su propio nombre a través de un mecanismo multicast (224.0.0.251). Existen diferentes implementaciones de este protocolo, siendo las más conocidas Bonjour (Apple) y Avahi (GNU/Linux, suele instalarse por defecto). El protocolo mDNS usa el pseudodominio de primer nivel .local, por lo que él se encarga de buscar la IP de cualquier nombre de dominio terminado en .local, que se debe corresponder con un equipo de la red local. Si no lo encuentra, el código [NOTFOUND=return] detiene la búsqueda y no se pasa a la siguiente fuente. De esta manera, si tenemos una máquina llamada pc1 (el nombre de un equipo se encuentra en el fichero /etc/hostname) y queremos, por ejemplo, hacerle ping, podremos hacerlo con el comando: ping pc1.local; la IP de pc1.local la buscará el equipo desde donde ejecutamos el comando ping, utilizando el protocolo mDNS (implementado por el software Avahi en GNU/Linux, paquete avahi-daemon en Debian) generando un tráfico multicast en la red local que pregunta por la IP de dicho nombre. Todo este procedimiento solo afecta a los nombres acabados en .local.

dns: indica que se busque en los servidores DNS especificados en /etc/resolv.conf.

Anteriormente, de esta funcionalidad se encargaba el fichero /etc/host.conf, pero el fichero /etc/nsswitch.conf lo ha dejado obsoleto.

Al fichero /etc/hosts se le pueden dar otros usos, como por ejemplo, bloquear el acceso de los usuarios a determinados sitios web porque puedan contener algún software malicioso como spyware o adware. Esto se hace añadiendo entradas para esos sitios que redirigen a direcciones que no existen, por ejemplo, a la de una máscara de subred, 255.255.255.0. Está claro, que un usuario un poco avispado, podrá saltarse esta limitación, escribiendo directamente la IP del sitio web.

Desde el punto de vista de la seguridad, el archivo /etc/hosts puede ser perjudicial cuando es modificado por alguien malintencionado para hacer un ataque de pharming, por ejemplo, un usuario malintencionado redireccionaría www.mibanco.com a una dirección IP como 93.184.216.119, que podría ser una réplica falsa de la página de mi banco, y así capturar datos privados. Normalmente los programas antivirus bloquean la modificación de este fichero.

NIS

DNS sirve un rango limitado de información, siendo la más importante la correspondencia entre el nombre de nodo y la dirección IP. Para otros tipos de información, no existe un servicio especializado así.

Por otra parte, si sólo se administra una pequeña LAN sin conectividad a Internet, no parece que merezca la pena configurar DNS. Ésta es la razón por la que Sun desarrolló el **Sistema de Información de Red** (NIS).

- NIS proporciona prestaciones de acceso a bases de datos genéricas que pueden utilizarse para distribuir, por ejemplo, la información contenida en los ficheros **passwd** y **groups** a todos los nodos de su red. Esto hace que la red parezca un sistema individual, con las mismas cuentas en todos los nodos.
- De manera similar, se puede usar NIS para distribuir la información de nombres de nodo contenida en /etc/hosts a todas las máquinas de la red.

Introducción a DNS

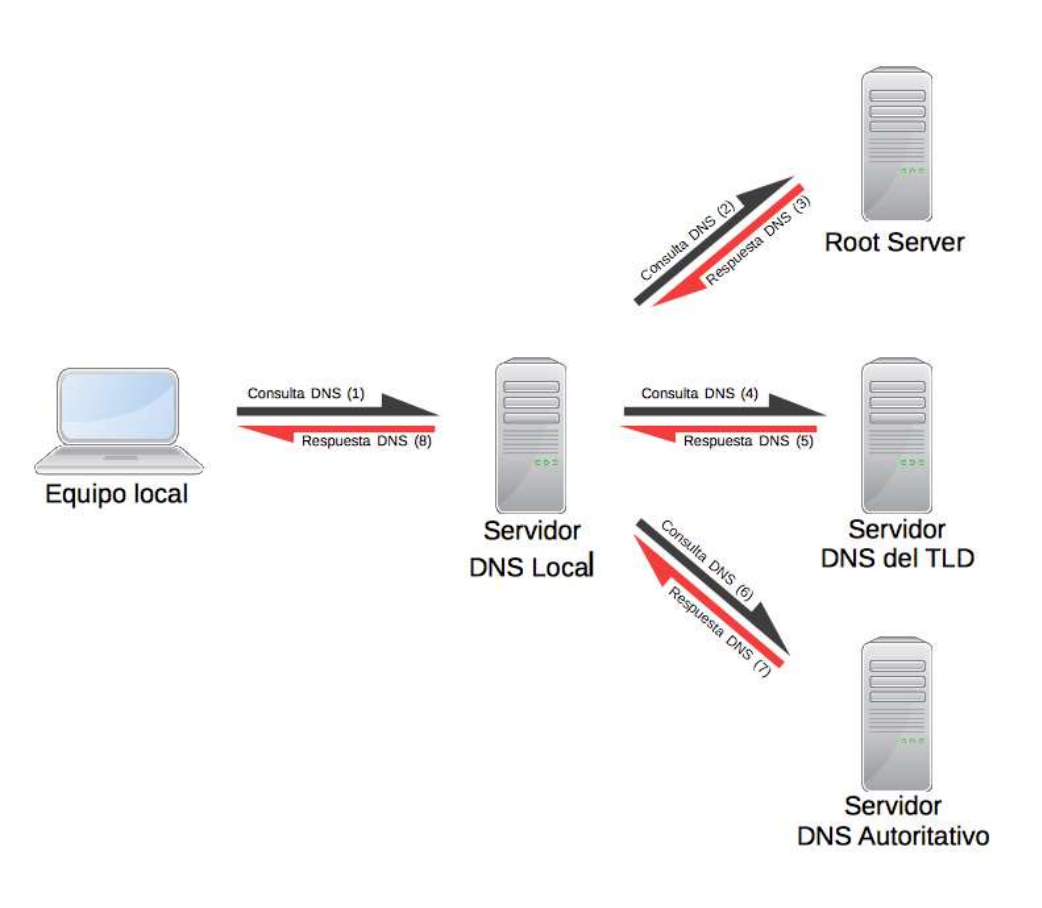
Los sistemas que usan las redes IP tienen que conocer la dirección IP de una máquina para poder conectarse. La mayor parte de los usuarios prefieren usar nombres de máquinas como el nombre de un host.

El uso de nombres de un dominio completamente cualificado tiene ventajas para los administradores del sistema y para los clientes:

- Da flexibilidad a la hora de cambiar las direcciones IP para máquinas individuales sin alterar el nombre de las máquinas.
- La configuración interna de la red es transparente para el usuario.
- En el caso de que una página web utilice una red de distribución de contenidos (CDN) el usuario recibirá la dirección IP del servidor más cercano según su localización geográfica.

El servidor DNS utiliza una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes. Aunque como base de datos el DNS es capaz de asociar diferentes tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a **direcciones IP** y la localización de los **servidores de correo electrónico** de cada dominio.

DNS proceso de consulta



DNS proceso de consulta

1 – Nuestro equipo no conoce la IP a la que debe conectarse, así que preguntará a un servidor DNS que tenga configurado (normalmente todos los equipos conocen al menos dos servidores DNS, por si uno falla).

2 – El servidor DNS si no conoce la respuesta, preguntará al siguiente en la jerarquía, es decir, a uno de sus propios servidores DNS.

3 – Esto ocurrirá un número indeterminado de veces hasta llegar a un **Root Server.** Este no resuelve nombres pero dirigirá la consulta a un **servidor DNS del dominio padre**, o TLD.

4 – El servidor del dominio padre de nuevo contestará la consulta, indicando cuáles son los servidores DNS autoritativos para el dominio buscado.

5 – La petición llegará a estos servidores DNS, y será contestada. Todos los equipos por los que ha pasado la consulta guardarán esta información durante un tiempo (para no tener que repetir la misma consulta).

6 – Por último, nuestro equipo, una vez obtenida la respuesta realizará una conexión a dicha IP.

¡Y todo esto en un par de segundos!

Root server DNS

Como en cualquier jerarquía, cuando hablamos de servidor DNS tiene que existir un nivel superior, un punto en el que una consulta no contestada no pueda subir más y tenga que ser resuelta de un modo u otro. En este nivel superior la consulta será resuelta por un Root Server.

Un **Root Server** es un servidor DNS un poco especial. Él no sabe a qué IP resuelve ningún dominio, pero conoce los servidores DNS de cada TLD bajo su jurisdicción; digamos que el Root Server no sabe nada, pero tiene una lista de todos los servidores que sí que saben, y puede indicar cuál es el que hace falta en cada momento. Actualmente existen 13 Root Servers en todo el mundo, operados y mantenidos por 12 organizaciones independientes.

TLD

TLD significa "**Top Level Domain**". Se trata del dominio "padre", y es responsabilidad de alguna entidad nacional o internacional que se encarga de gestionar los servidores de nombres que tienen información sobre esta extensión. Por ejemplo, los dominios **.es** son responsabilidad de nic.es, los **.com** son responsabilidad de Verisign, etc ...

Del mismo modo que un Root Server, los servidores DNS de los TLD no conocen la IP a la que resuelve ningún dominio, pero saben cuáles son los DNS autoritativos de cada dominio bajo su jurisdicción.

Caché DNS

Si ahora preguntamos a un DNS qué IP resuelve un dominio, y nos contesta y cinco minutos volvemos a preguntar por la misma IP … ¿qué posibilidades hay de que la respuesta haya cambiado? Muy pocas, con lo que es absurdo repetir todo el proceso de consultas explicado antes cada vez que se quiere resolver un dominio.

Normalmente, cuando un equipo obtiene respuesta de un servidor DNS la guarda en cache y durante un tiempo no vuelve a hacer la misma consulta; se fía del resultado anterior.

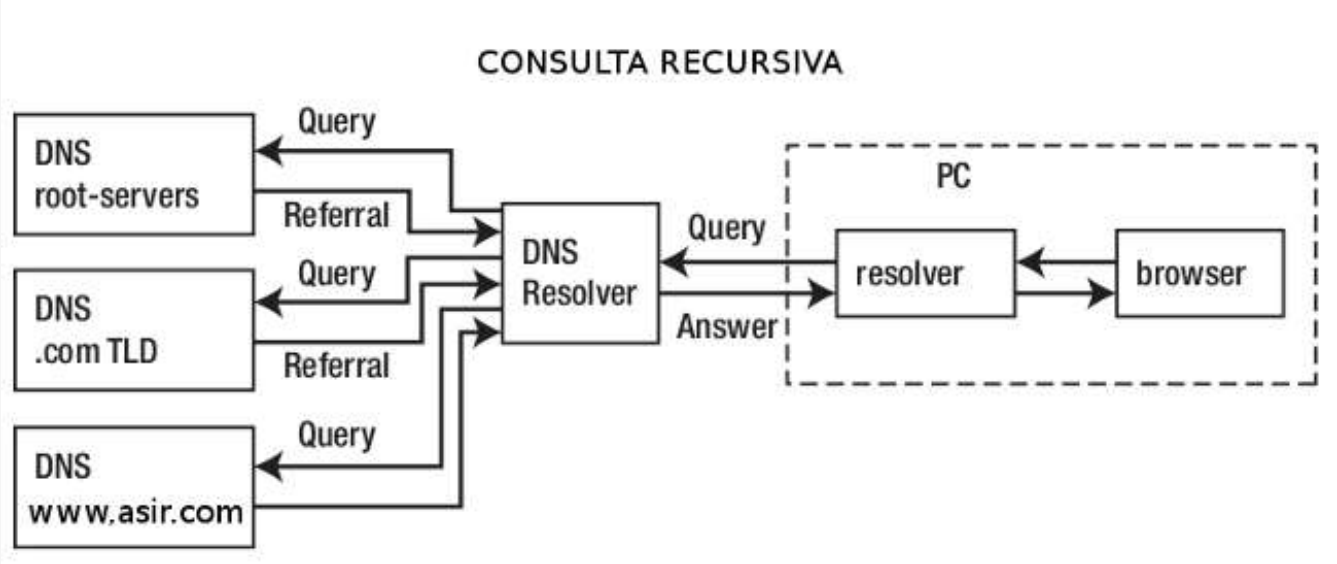
Esto es bueno porque reduce el tiempo que se tarda en acceder a las páginas web y reduce la carga a la que se ven sometidos los servidores.

Pero también tiene su punto flaco: si cambias la IP a la que apunta tu dominio el resto de servidores del mundo tardarán un tiempo en darse cuenta, mientras aún se fíen del último resultado. Este tiempo es lo que se suele conocer como **propagación DNS.**

Tipos de resolución de nombres de dominio

Un servidor DNS puede resolver un nombre de dominio de manera recursiva o iterativa.

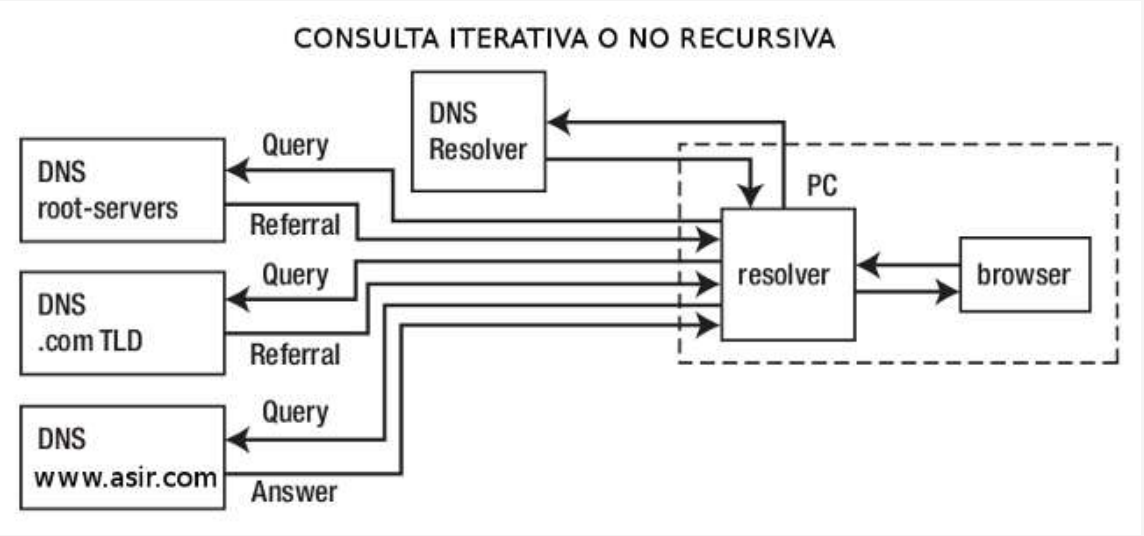
En una **consulta recursiva**, un cliente solicita a un servidor DNS que obtenga por sí mismo la respuesta completa (es decir, dado el dominio mi.dominio.com, el cliente espera recibir la dirección IP correspondiente).



Tipos de resolución de nombres de dominio

Por otro lado, dada una **consulta iterativa**, el servidor DNS no otorga una respuesta completa: para el caso de mi.dominio.com, el primer servidor al que se le realiza la consulta (un servidor raíz), retorna las direcciones IP de los servidores de nivel superior (TDL) responsables del dominio .com. De este modo, el cliente ahora debe realizar una nueva consulta a uno de estos servidores, el cual toma nota del sufijo .dominio.com y responde con la IP del servidor DNS correspondiente, por ejemplo dns.dominio.com. Finalmente, el cliente envía una nueva consulta a dns.dominio.com para obtener la dirección IP de mi.dominio.com.

En la práctica, la consulta de un host a un DNS local es recursiva, mientras que las consultas que realiza el DNS local son iterativas.



Zonas DNS

Las zonas DNS son las “hojas” en las que está la información de cada dominio. Una zona no es más que un fichero de texto en el servidor (exactamente igual que uno que se podría crear con un bloc de notas) pero con un formato específico, que le permite al servidor DNS interpretar la información que hay en ella.

Así, si un servidor DNS tiene información para 500 dominios, tendrá 500 ficheros de texto cada uno con la información de uno de esos dominios; es decir, tendrá 500 Zonas DNS.

DNS Tipos de registros

- **A** = Dirección (address). Este registro se usa para traducir nombres de servidores de alojamiento a direcciones IPv4.
- **AAAA** = Dirección (address). Este registro se usa en IPv6 para traducir nombres de hosts a direcciones IPv6.
- **CNAME** = Nombre canónico (canonical Name). Se usa para crear nombres de servidores de alojamiento adicionales, o alias, para los servidores de alojamiento de un dominio. Es usado cuando se están corriendo múltiples servicios (como FTP y servidor web) en un servidor con una sola dirección IP. Cada servicio tiene su propia entrada de DNS (como ftp.ejemplo.com. y www.ejemplo.com.). Esto también es usado cuando corres múltiples servidores HTTP, con diferentes nombres, sobre el mismo host. Se escribe primero el alias y luego el nombre real. Ej. Ejemplo1 IN CNAME ejemplo2
- **NS** = Servidor de nombres (name server). Define la asociación que existe entre un nombre de dominio y los servidores de nombres que almacenan la información de dicho dominio. Cada dominio se puede asociar a una cantidad cualquiera de servidores de nombres.
- **MX** = Intercambio de correo (mail exchange). Asocia un nombre de dominio a una lista de servidores de intercambio de correo para ese dominio. Tiene un balanceo de carga y prioridad para el uso de uno o más servicios de correo.
- **PTR** = Indicador (pointer). También conocido como 'registro inverso', funciona a la inversa del registro A, traduciendo IPs en nombres de dominio. Se usa en el archivo de configuración de la zona DNS inversa.
- **SOA** = Autoridad de la zona (start of authority). Proporciona información sobre el servidor DNS primario de la zona.
- **SRV** = Service record (SRV record).
- **ANY** = Toda la información de todos los tipos que exista. (No es un tipo de registro, sino un tipo de consulta)

DNS Temas de seguridad

Originalmente, las preocupaciones de seguridad no fueron consideraciones importantes para el diseño en el software DNS o de cualquier otro software para despliegue en la Internet temprana, ya que la red no estaba abierta a la participación del público general. Sin embargo, la expansión de Internet en el sector comercial en los 90s cambió los requisitos de las medidas de seguridad para proteger la integridad de los datos y la autenticación de los usuarios.

Muchos temas de vulnerabilidades fueron descubiertos y explotados por usuarios maliciosos. Uno de esos temas es el **envenenamiento de caché DNS**, en la cual los datos son distribuidos a los resolvedores de caché bajo el pretexto de ser un servidor de autoridad de origen, contaminando así el almacenamiento de datos con información potencialmente falsa y largos tiempos de expiración (time-to-live). Subsecuentemente, las solicitudes legítimas de las aplicaciones pueden ser redirigidas a equipos de red operados con contenidos maliciosos.

Las respuestas DNS tradicionalmente no estaban firmadas criptográficamente, permitiendo muchas posibilidades de ataque; las extensiones de seguridad del DNS (DNSSEC) modifican el DNS para agregar la posibilidad de tener respuestas firmadas criptográficamente. DNSCurve ha sido propuesto como una alternativa a DNSSEC. Otras extensiones, como TSIG, agregan soporte para autenticación criptográfica entre pares de confianza y se usan comúnmente para autorizar transferencias de zona u operaciones dinámicas de actualización.

Algunos nombres de dominio pueden ser usados para conseguir efectos de engaño. Por ejemplo, paypal.com y paypa1.com son nombres diferentes, pero puede que los usuarios no puedan distinguir la diferencia dependiendo del tipo de letra que estén usando. En muchos tipos de letras la letra l y el numeral 1 se ven muy similares o hasta idénticos. Este problema es grave en sistemas que permiten nombres de dominio internacionalizados, ya que muchos caracteres en ISO 10646 pueden aparecer idénticos en las pantallas típicas de computador. Esta vulnerabilidad se explota ocasionalmente en phishing.

Técnicas como el FDNS inverso con confirmación adelantada pueden también usarse para validar los resultados de DNS.

Configuración DNS

En una red local

Configuración DNS Linux

```
# yum install bind bind-utils

# systemctl start named

# systemctl enable named

# named -v
```

Configuración DNS Linux

cp /etc/named.conf /etc/named.conf.orig
vi /etc/named.conf

```
options {
listen-on port 53 { 127.0.0.1; 192.168.56.100;};
    listen-on-v6 port 53 { ::1; };
    directory     "/var/named";
    dump-file     "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    allow-query     { localhost; 192.168.56.0/24;}; ### IP Range ###
    allow-transfer  { none; };   ### Slave DNS IP ###
```

```
acl “miredlocal” {
   192.168.56.0/24;
};
```

Y añadir las zonas directa e inversa de resolución

```
zone "as.local" IN {
type master;
file "db.as.local";
allow-update { none; };  // no DDNS by default
};

zone "56.168.192.in-addr.arpa" IN {
type master;
file "db.192.168.56";
allow-update { none; };
};
```

Configuración DNS Linux

Crear el fichero /var/named/db.as.local

```
$TTL 1D
@   IN SOA linux.as.local. root@linux.as.local. (
        0  ; Serial
        1D ;Refresh
        1H ;Retry
        1W ;Expire
        3H ;Minimun TTL
)

;Servidor
@   IN NS linux.as.local.

;Hosts
linux      IN A 192.168.56.100
w7         IN A 192.168.56.111
ws2019  IN A 192.168.56.101

;Alias
www    IN CNAME linux
```

Configuración DNS Linux

Crear el fichero /var/named/db.192.168.56

```
$TTL 1D
@   IN SOA linux.as.local. root@linux.as.local. (
        0  ; Serial
        1D ;Refresh
        1H ;Retry
        1W ;Expire
        3H ;Minimun TTL
)

;Servidor
@   IN NS linux.as.local.

;Hosts
100  IN PTR linux.as.local.
101  IN PTR ws2019.as.local.
111  IN PTR w7.as.local.
```

Configuración DNS Linux

Después comprobar con

```
# named-checkzone as.local /var/named/db.as.local
# named-checkzone 56.168.192.in-addr.arpa /var/named/db.192.168.56
```

Protecciones, puesta en marcha y firewall

```
# chown named:named /var/named/db.as.local
# chown named:named /var/named/db.192.168.56
# chmod 777 /var/named/db.as.local
# chmod 777 /var/named/db.192.168.56

# systemctl start named
# systemctl enable named

# firewall-cmd --zone=internal --add-service=dns
# firewall-cmd --zone=internal --add-service=dns --permanent
```