1.Introducción

¿Qué es DNS y por qué es necesario?

- El DNS (Domain Name System/Service) tiene como objetivo principal resolver nombres inteligibles (próximos al lenguaje) por identificadores binarios para poder localizar y direccionar los equipos conectados a la red, tanto en sentido directo como inverso.
- Con un Servidor de Nombres, un host solo necesita conocer la dirección física de un Servidor de Nombres y el nombre del recurso para consultarlo y encontrar su dirección o cualquier otro atributo almacenado, ya que la función más conocida del DNS es la asignación de nombres a direcciones IP (mapeo directo) y la subsiguiente localización de, por ejemplo, un servidor de correo electrónico. También puede traducir la dirección IP física al nombre de un recurso, lo que se denomina mapeo inverso.
- Se necesitan servicios de resolución de nombres en diversos escenarios, como la navegación web, el correo electrónico y las redes internas. Sin un Servicio de Nombres no existiría una Internet viable.
- Antes del DNS, cada host necesitaba conocer la dirección IP física de un recurso (como una página web) para acceder a él, una tarea imposible con millones de hosts y miles de millones de páginas web.
- Aunque el servicio DNS surge en 1983, el concepto de Servidores de Nombres se creó a mediados de los años 70 para mantener los atributos de un recurso nombrado en una ubicación conocida, lo que simplificó y dinamizó la gestión de la red.

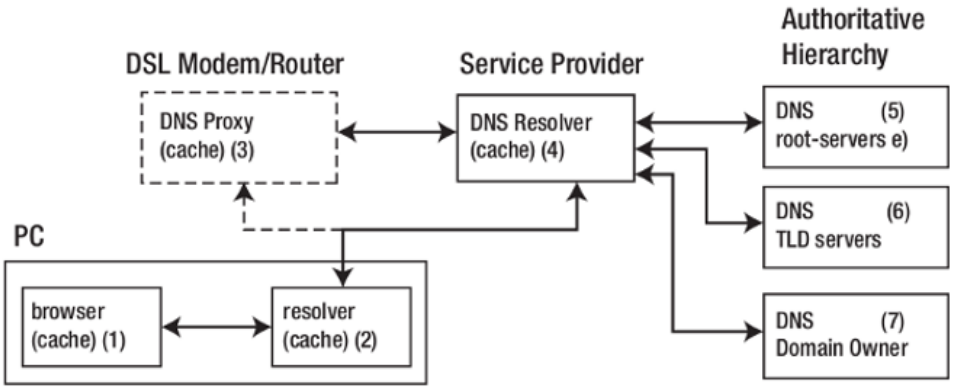
¿Qué motiva la aparición del servicio?

Existían problemas de usabilidad con direcciones IP, motivo por el cual surge el servicio DNS:

- Antes del DNS, se utilizaba el fichero /etc/hosts (en sistemas UNIX) para resolver nombres, centralizando en un servidor la relación de todos los nombres de forma exhaustiva.
- Sin embargo, el uso del archivo /etc/hosts presentaba varios inconvenientes:
  - Era poco escalable.
  - Generaba excesivo tráfico en el servidor.
  - Ocasionaba problemas de inconsistencias en las copias locales.
  - Llevaba a la aparición de nombres duplicados.
  - Por estas razones, no se podría usar para resolver nombres de manera general.
- En Microsoft Windows, existe un archivo similar en la ruta .../system32/drivers/etc/hosts.
- La utilidad real de /etc/hosts hoy en día es para una red simple sin servicio DNS.
- Un nuevo problema con los Servidores de Nombres es que, si el servidor no funciona, el host no puede acceder a ningún recurso en la red, lo que lo convierte en un recurso crítico. Para resolver esto, nacieron los conceptos de Servidores de Nombres Primarios y Secundarios (muchos sistemas permiten terciarios o más).
- A medida que la red crece, se acumula una gran cantidad de Nombres en la base de datos del Servidor de Nombres, lo que genera tres nuevos problemas:
  - La búsqueda de entradas en la base de datos se vuelve cada vez más lenta.
  - La carga en los Servidores de Nombres se vuelve muy alta.
  - La gestión de muchos registros de Nombres se vuelve cada vez más difícil debido a que todos intentan actualizar los registros al mismo tiempo.
- Esto condujo a la necesidad de una jerarquía de nombres, de distribuir las cargas operativas de los servidores de nombres y de delegar la administración de los mismos.

¿Cómo se construye el servicio?

- El DNS se construye como una gran base de datos distribuida y jerárquica.
  - Como base de datos distribuida, la información se reparte por toda la red, lo que permite que desaparezca la carga excesiva en la red y en los hosts.
  - La consistencia de la información se logra al actualizarse automáticamente sin intervención del administrador, eliminando la duplicidad de nombres al existir dominios controlados por un único administrador (puede haber nombres iguales pero en dominios diferentes).
- Se utiliza una arquitectura cliente-servidor para la resolución de nombres:
  - El DNS utiliza TCP para transferencias de zona y UDP para consultas, ambos en el puerto 53.
  - Para las consultas, el cliente también usa el puerto 53 como puerto de origen.
  - Un sistema DNS, según RFC 1034, incluye tres partes:
    - Datos que describen los dominios:
      - Un solo servidor DNS puede soportar muchos dominios.
      - Los datos para cada dominio se definen en forma de Registros de Recursos (Resource Records - RRs) textuales organizados en Archivos de Zona (Zone Files).
      - El formato de los Archivos de Zona está definido en RFC 1035 y es compatible con la mayoría del software DNS.
    - Uno o más programas de Servidor de Nombres, que típicamente realiza tres funciones:
      - Lee un archivo de configuración (como named.conf para BIND) que define las zonas de las que es responsable.
      - Un archivo de configuración puede describir varios comportamientos, como el almacenamiento en caché (caching) o no.
      - Responde a consultas de hosts locales o remotos.
    - Un programa o biblioteca de resolución (resolver):
      - Se encuentra en cada host y permite traducir una solicitud de usuario (por ejemplo, www.institucionlasalles.es) en una o más consultas a servidores DNS utilizando protocolos UDP (o TCP).
      - El resolver en todos los sistemas Windows y la mayoría de los sistemas *nix es en realidad un 'resolver stub' (mínimo), que solo funciona con un DNS que soporte consultas recursivas y que disponen de una caché para acelerar las respuestas y reducir el uso de la red.



2.Contenidos

Introducción:

- Resolución directa: traduce un nombre de host a una dirección IP.
- Mapeo inverso (resolución inversa): dada una dirección IP, obtiene el nombre de host correspondiente (el término “consultas inversas” quedó en desuso según RFC 3425; hoy se habla del mapeo inverso).

Dominios especiales:

- IPv4: in-addr.arpa, que se usa para el mapeo inverso de direcciones IPv4.
- IPv6: ip6.arpa, que se usa para el mapeo inverso de direcciones IPv6.

- Para integrar una IP en estos dominios es necesario invertir el orden de sus componentes (por ejemplo, los octetos en IPv4). Esto se hace para que la jerarquía de nombres (más genérica a la derecha) coincida con la estructura de las direcciones IP.

Cómo funciona (aspectos técnicos y flujo):

- El mapeo inverso se implementa mediante consultas DNS que apuntan al dominio especial correspondiente.
- El tipo de registro empleado en DNS para devolver el nombre es el registro PTR (Pointer).
- Las consultas pueden realizarse de forma recursiva o iterativa según el resolver y la configuración; el proceso general es una resolución DNS normal aplicada al espacio in-addr.arpa / ip6.arpa.
- Evita búsquedas exhaustivas en el conjunto de dominios porque la inversión del orden permite resoluciones jerárquicas eficaces.

Ejemplos de consulta:

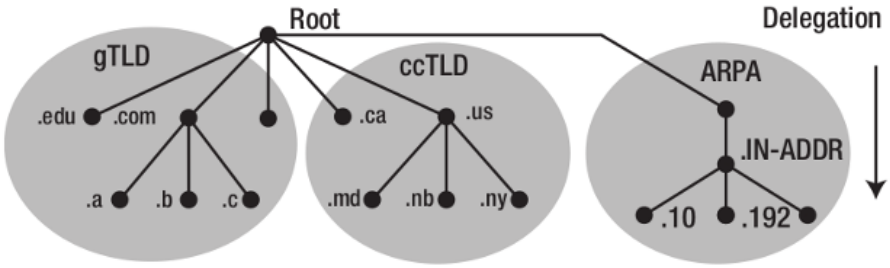
- Directa: "¿Cuál es la IP del host www en el dominio mydomain.com?"
- Inversa / mapeo inverso: "¿Cuál es el nombre del host cuya IP es x.x.x.x?" (o x:x:x:x:x... en IPv6)

Usos prácticos y razones:

- Diagnóstico: localizar el nombre asociado a una IP para solucionar problemas.
- Seguridad / trazabilidad: rastrear actividad maliciosa (hackers, spammers).
- Correo electrónico: la mayoría de los sistemas de correo realizan una verificación sencilla en dos pasos, primero traducción de nombre por IP y después traducción de IP por nombre, como medida básica de autenticación y filtrado.

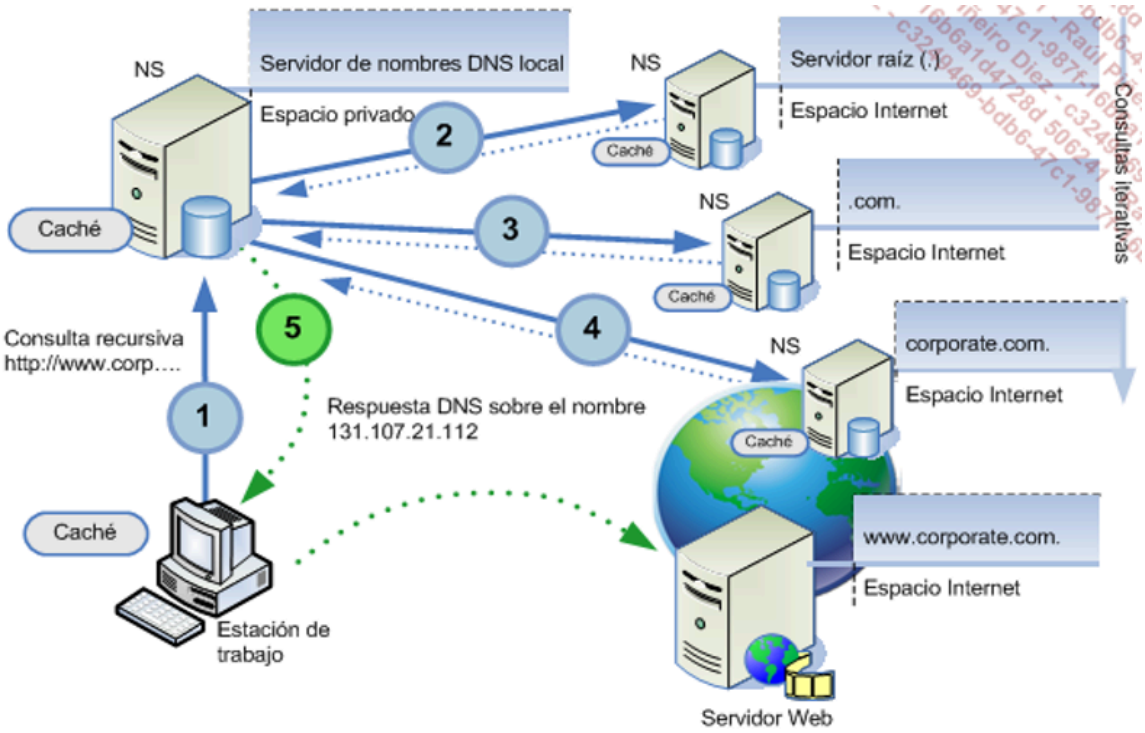
Estado y requisitos prácticos:

- IPv4: el mapeo inverso no fue estrictamente obligatorio históricamente, pero hoy su implementación y mantenimiento son prácticamente esenciales para hosts que envían correo (MTA/MUA) y para sistemas que dependen de validaciones basadas en origen.
- IPv6: inicialmente exigía mapeo inverso, pero ese requisito fue relajado y ya no es obligatorio en la práctica.
- RFC 3425: marca el paso del término clásico "consultas inversas" hacia el uso del concepto de mapeo inverso; por eso es habitual referirse ahora al proceso como 'reverse mapping'.



Tipos de consulta

Observa la imagen siguiente:



Consulta Recursiva

- Definición: El servidor debe devolver la respuesta completa al cliente, consultando otros servidores si es necesario.
- Obligatoriedad: No es obligatorio que todos los servidores soporten recursión.
- Funcionamiento:
  - El resolver (servidor de nombres con caché) actúa en nombre del cliente (stub resolver) y sigue la pista en la jerarquía DNS hasta obtener la respuesta final.
  - Internamente, el resolver recursivo típicamente utiliza consultas iterativas para obtener la información requerida.
  - El bit RD (Recursion Desired) en la cabecera de la consulta indica si se solicita recursión (RD=1) o no (RD=0).
- Posibles respuestas:
  - Respuesta completa con registros A y CNAME; se indica si la respuesta es autoritativa o no.
  - NXDOMAIN (el dominio o equipo no existen).
  - Error temporal (p. ej. inaccesibilidad de servidores o problemas de red).

Consulta Iterativa (No recursiva)

- Definición: El servidor responde si tiene la información, o devuelve una referencia (lista de servidores de nombres) a otro servidor que pueda resolver la petición.
- Obligatoriedad: Todos los servidores DNS deben soportar consultas iterativas. Los servidores raíz y TLD solo aceptan iterativas.
- Funcionamiento y propiedades:
  - Puede devolver una respuesta autoritativa o no; NXDOMAIN; error temporal; o una referencia (lista de IPs de servidores de nombres).
  - Es más rápida cuando el servidor tiene la respuesta en caché; si no, devuelve inmediatamente una referencia.
  - Otorga mayor control al solicitante y es especialmente útil para diagnóstico.
  - Técnicamente es una consulta DNS normal con recursión deshabilitada (RD=0).

Flujos de consulta (ejemplos)

Flujo iterativo: proceso general

1. Cliente consulta su servidor DNS configurado; no hay respuesta autoritativa ni en caché.
2. El servidor consulta uno de los servidores raíz.
3. El servidor raíz devuelve una referencia al servidor(es) del TLD.
4. El servidor consulta al TLD correspondiente.
5. El TLD devuelve una referencia al servidor autoritativo del dominio solicitado.
6. El servidor consulta al servidor autoritativo del dominio.
7. El servidor autoritativo devuelve la respuesta final al cliente.

Nota: en este proceso cada servidor devuelve referencias (o la respuesta si la tiene), por lo que es un ejemplo de consulta iterativa.

Flujo recursivo: Ejemplo detallado (www.example.com)

1. El usuario escribe www.example.com en el navegador.
2. El navegador llama al stub resolver local.

3. El stub resolver envía una consulta al DNS resolver local (servidor recursivo) con RD=1 pidiendo recursión.
4. El DNS resolver busca en su caché; si no encuentra la entrada, consulta un servidor raíz (normalmente con RD=0, es decir, iterativa).
5. El servidor raíz responde con referencia a los servidores TLD .com.
6. El DNS resolver consulta un servidor TLD de .com (iterativa).
7. El TLD responde con referencia a los servidores de nombres de example.com.
8. El DNS resolver consulta uno de los servidores autoritativos de example.com (iterativa).
9. El servidor autoritativo devuelve el registro A para www.example.com (respuesta completa).
10. El DNS resolver envía la respuesta al stub resolver y la guarda en su caché.
11. El stub resolver guarda la información en su caché (muchos stub resolvers modernos tienen caché) y responde a la llamada del navegador.
12. El navegador recibe la respuesta, la almacena en caché y procede a iniciar la sesión HTTP.

Resumen: el stub solicita recursión al DNS resolver; el resolver implementa la recursión usando consultas iterativas a otros servidores. Si un equipo apunta a un servidor DNS que solo soporta consultas iterativas y no existe un resolver recursivo disponible, la resolución completa no funcionará.

Conceptos clave

- Stub resolver: resolver mínimo (presente en PCs y sistemas Unix/Windows) que no sigue referencias; en sistemas modernos suele incluir caché.
- Resolver (servidor recursivo): servidor con caché que puede realizar consultas recursivas en nombre del stub resolver, utilizando internamente consultas iterativas.
- Referencias: listas de servidores de nombres que apuntan al siguiente nivel en la jerarquía DNS (p. ej. del raíz al TLD, del TLD al dominio autoritativo).

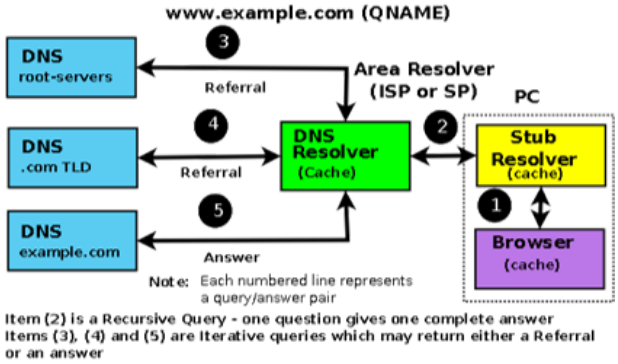


***Imagen tomada de Zytrax.com**

Servidores DNS y almacenamiento en caché

Funcionamiento del DNS Resolver con caché

- Obtiene información de un servidor autoritativo y la guarda localmente en caché.
- En solicitudes posteriores, responde desde caché hasta que caduque el tiempo de vida del registro (TTL), momento en que actualiza los datos desde el servidor autoritativo.
- Comportamiento según tipo de consulta:
  - Con recursivas: responde completamente o devuelve error.
  - Con iterativas: responde desde caché si tiene la respuesta, o devuelve referencia/error.
- Si obtiene datos directamente del servidor autoritativo, la respuesta será autoritativa; si provienen de caché será no autoritativa.

Tipos de servidores y características

- Solo autoritativo: no almacena en caché.
- Servidor de reenvío (forwarder): almacena en caché resultados obtenidos, que obtiene a partir de consultas realizadas a los servidores que tienen configurados como reenviadores
- Configuraciones comunes:
  - Servidor maestro/esclavo para algunas zonas y como resolver con caché para otras (propósito general).
  - Servidor solo resolver (solo caché) para minimizar tráfico externo o compensar enlaces lentos.

En el caso del software DNS que nosotros vamos a usar (BIND):

- Almacenamiento en caché activado por defecto (recursion yes).
- Asociado con type hint en declaración de zona (necesario para consultas recursivas a servidores raíz).
- La caché se vacía al reiniciar BIND.

Seguridad y buenas prácticas

- Mezclar zonas maestras/esclavas con servicios de caché/recursivos puede aumentar riesgos de seguridad.
- En entornos de alto volumen, se recomienda usar solo servidores autoritativos sin caché/recursión.

Resolución de nombres en GNU/Linux

Ya se comparó en un apartado anterior la diferencia entre resolver nombres usando un servicio DNS o un archivo como /etc/hosts.

Mecanismo de resolución modular

- En GNU/Linux, el archivo /etc/nsswitch.conf permite registrar varias fuentes de información (ej.: /etc/hosts, mDNS, servidores DNS de /etc/resolv.conf).
- Define el orden de búsqueda para información administrativa (hosts, passwd, group, shadow, networks, etc.).

Archivo /etc/host.conf

- Contiene el orden de resolución, ej.: order hosts, bind.
- Opción multi permite múltiples IP para un mismo host.
- Está obsoleto, sustituido por /etc/nsswitch.conf.

Archivo /etc/resolv.conf

- Contiene servidores de nombre (manual o por DHCP), sufijos de búsqueda y dominios a añadir a nombres no cualificados.
- Es estático si resolvconf no está instalado.
- Con resolvconf, se convierte en enlace simbólico y su contenido se gestiona automáticamente mediante scripts.

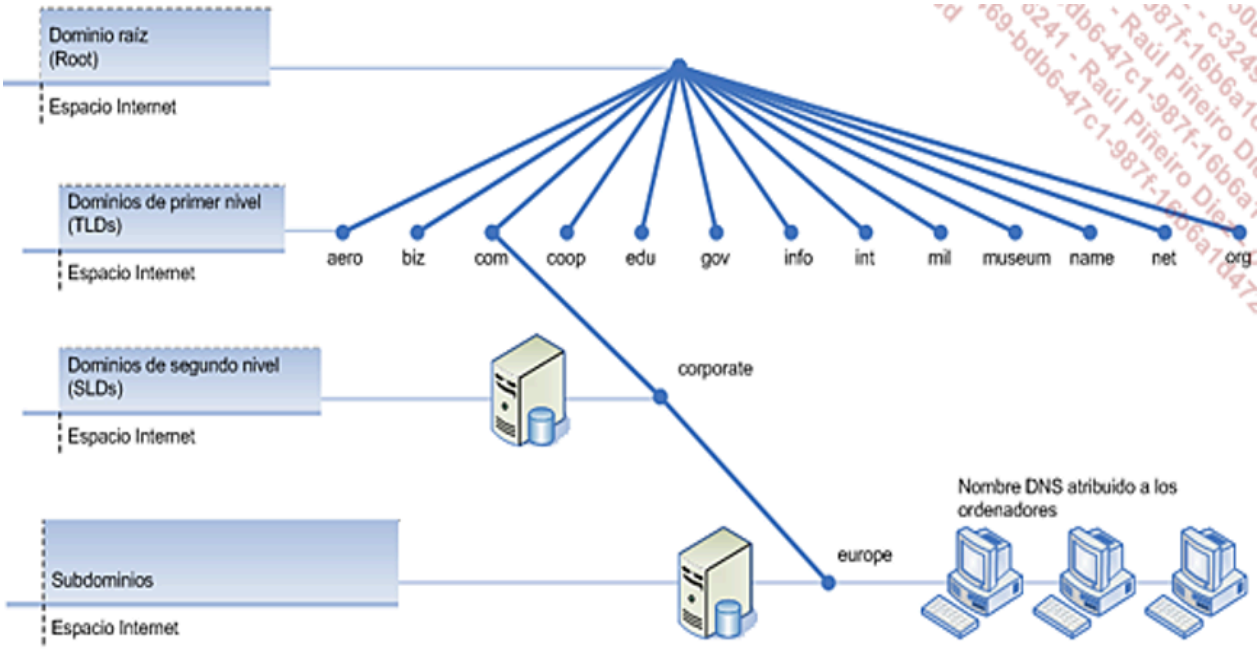
Nosotros utilizamos el software asociado a **Systemd**

3.Contenidos

Jerarquía del Sistema de Nombres de Dominio (DNS)

Dominio raíz y TLD

- El dominio raíz (RD) es el punto más alto del árbol DNS, sin nombre asociado, representado como un punto ('.').
- Debajo están los TLD (Top Level Domains):
  - gTLDs genéricos: .com, .org, .edu, administrados por ICANN.
  - ccTLDs: códigos de país (ISO 3166, dos letras), delegados a cada país.
  - Nuevos gTLD desde 2004: patrocinados (sTLD, con administración restringida, ej.: .museum, .gal, .cat, .aero, .travel, .jobs) y no patrocinados.
  - Política desde 2011: sin restricciones, con pago y cumplimiento de procedimientos.
  - gTLDs históricos: categorización flexible, algunos con registro limitado (.mil, .edu, .gov, .int).
  - Nuevos dominios gTLDs: https://gtldresult.icann.org/application-result/applicationstatus
- ICANN gestiona IPs, identificadores de protocolo, TLD, ccTLD y servidores raíz, y autoriza registradores.



Estructura jerárquica y nombres

- Bajo los TLDs: dominios de segundo nivel (SLD) y subdominios.
- Cada etiqueta (máx. 63 caracteres) es una subdivisión; nombre total máx. 255 caracteres; sin distinción de mayúsculas/minúsculas.
- FQDN (Fully Qualified Domain Name): nombre absoluto que termina en punto (ej.: www.example.com.) y define un nombre único hasta la raíz.
- Un nombre sin punto final no es FQDN.
- IDN (Internationalized Domain Names, RFC 3490): representación ASCII de etiquetas no ASCII. Por ejemplo para poder usar la 'ñ'.
- "Domain Name" = nombre de dominio + TLD, leído de izquierda a derecha (más específico a la izquierda).
- En gTLD, la parte registrada es el SLD; en ccTLD, puede ser tercer nivel (ej.: example.co.uk).

Delegación y autoridad

- ICANN tiene autoridad sobre la raíz.
- gTLD administrados por ICANN y delegados a registradores; ccTLD delegados a países.
- Cada nivel puede delegar autoridad al siguiente (zona).
- Lectura de derecha a izquierda permite rastrear delegaciones.

Nombre de host y subdominios

- Parte más a la izquierda es el nombre de host (ej.: www en www.example.com), que puede referirse a un host o servicio.
- Un host debe ser único dentro del "Domain Name".
- Partes intermedias (ej.: us en www.us.example.com) son subdominios gestionados por el propietario del dominio.
- El propietario puede delegar cualquier subdominio y es responsable de su gestión y de proveer registros DNS autoritativos.

Tipos de Servidores DNS

Servidores Raíz

- 13 servidores, responsabilidad de ICANN, operados por un consorcio (RFC 2870).
- Instancias múltiples (Anycast), hasta 200, para resiliencia.
- Referencias a servidores de dominios de segundo nivel; reciben consultas que no pueden resolverse localmente.
- Asignados FQDN de a.root-servers.net a m.root-servers.net; llamados “Sugerencias Raíz”.
- Estándares de operación definidos en RFC 2870 por solicitud del RSSAC a la IETF.
- Conocidos por todos los DNS públicos; punto de partida de la búsqueda de nombres.
- Anycasting: mismas IPs, datos enviados a instancia más cercana.

Servidor Autoritativo

- Posee información de una zona localmente (bien por ser primario o por haberla recibido por transferencia de zona)
- Responde como “Autoritativo” (bit AA en respuesta) cuando tiene autoridad.
- “Solo Autoritativo”: maestro o esclavo de zona, sin caché; usado en DNS Stealth o de alto rendimiento.
- Por ejemplo, en BIND, lo lograremos desactivando recursión ('recursion no' en named.conf).

Servidor Maestro (Primary)

- Obtiene datos de fuente local (archivo de zona).
- “Maestro” sustituye a “Primario” desde BIND 8.x.
- En BIND se puede definir en el archivo named.conf con 'type master'.
- Puede haber múltiples maestros por zona; sincronización obligatoria.
- “Primary Master” es el del registro SOA en actualizaciones dinámicas.
- Puede notificar cambios (NOTIFY) a esclavos; puede ser oculto (no listado en NS públicos).

Servidor Esclavo (Secondary)

- Datos obtenidos por transferencia de zona desde un maestro.
- Responde como autorizado si está definido como esclavo y tiene zona válida.
- “Esclavo” sustituye a “Secundario” desde BIND 8.x.
- Puede haber múltiples esclavos por zona; si bien también pueden ser maestros para otros.
- En BIND se consigue configurando el archivo named.conf con 'type slave'.
- Maestro definido en cláusula masters; si no accesible al expirar SOA, deja de responder.
- Parámetro 'file' opcional para guardar zonas y reutilizarlas tras reinicio antes de expirar.

Servidor Resolver (Recursivo o Caché)

- Obtiene datos de un servidor autorizado y los almacena en caché hasta que caduque el TTL.
- Respuesta autoritativa si proviene de maestro; no autoritativa si de caché.
- En BIND, almacenamiento en caché por defecto (recursion yes).
- Tener en cuenta aque la caché se pierde al reiniciar BIND.
- Suele incluir zona raíz con 'type hint' para dominios no definidos como maestro o esclavo.

Servidor DNS Reenviador (Forwarder)

- Reenvía solicitudes a otro DNS y almacena en caché; útil cuando acceso externo es lento/costoso, para reducir tráfico o centralizar gestión.
- Puede sanear tráfico o usarse en configuración Split Server.
- En BIND, configurado con forward y forwarders globalmente o por zona; forward only desactiva consultas recursivas.

Servidor DNS Oculto (Hidden/Stealth)

- No aparece en registros NS públicos; usado para proteger información interna.
- Externos: solo respuestas autoritativas, sin caché, sin recursión.
- Archivo de zona pública con solo datos visibles (SOA, NS, MX, A de www/ftp).
- Transferencias permitidas solo entre servidores públicos; prohibidas con el Stealth.
- Servidor interno puede ofrecer servicios internos y externos con zona privada.
- Declaración 'view' en BIND 9 permite respuestas distintas para solicitudes internas y externas, pero no protege contra compromiso del host.

Nomenclatura de zonas, FQDN, TTL, zonas directas e inversas

Zonas

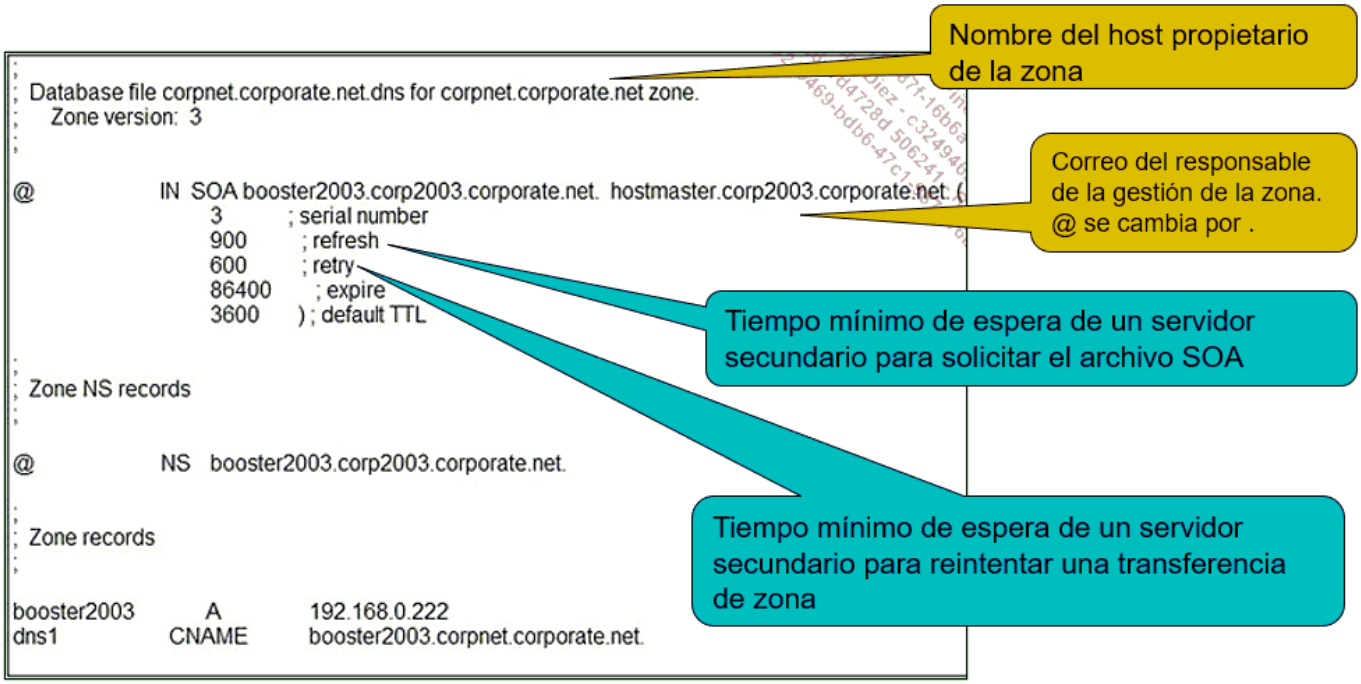
- Una zona es una parte contigua del árbol de nombres que se administra como una unidad.
- Los dominios de primer nivel y la mayoría de los de segundo nivel se dividen en zonas para facilitar la administración.
- Una zona puede almacenar registros con información de múltiples dominios.
- Diferencia con dominio:
  - Un dominio es un subárbol del espacio de nombres.
  - Una zona es una parte contigua del espacio de nombres DNS almacenada en un fichero y que puede incluir varios dominios.
- RFC 1034: describe las zonas de subdominios como subzonas. Una zona es una conveniencia operativa para el software DNS y no parte de la jerarquía de nombres.

Ejemplo de contenido de un archivo de zona:

```
; zone file for example.com
$TTL 2d    ; 172800 secs default TTL for zone
$ORIGIN example.com.
@             IN      SOA   ns1.example.com. hostmaster.example.com. (
                        2003080800 ; se = serial number
                        12h        ; ref = refresh
                        15m        ; ret = update retry
                        3w         ; ex = expiry
                        3h         ; min = minimum
                        )
              IN      NS      ns1.example.com.
              IN      MX  10  mail.example.net.
joe           IN      A       192.168.254.3
www           IN      CNAME   joe

Algunas preguntas al respecto del fichero de zona:
```

- ¿Cuál es el nombre del host propietario de la zona?
- ¿Cuál es el correo del responsable de la gestión de zona? ¿Qué particularidad presenta?
- ¿Cuál es el tiempo mínimo de espera de un servidor secundario para solicitar el registro SOA?
- ¿Cuál es el tiempo mínimo de espera de un servidor secundario para reintentar una transferencia de zona?



TTL (Time-To-Live)

- Indica cuánto tiempo un registro puede almacenarse en caché.
  - Información estable: valores grandes (ej. 86400 segundos = 1 día).
  - Información volátil: valores pequeños (ej. 60 segundos).
- El TTL vigente determina cuándo los 'resolvers' actualizarán sus cachés desde el servidor autorizado.
- Es un valor de 32 bits (1 a 2147483647). Cero significa que no se almacena en caché.
- Se puede indicar en periodos de tiempo

Convención de nombres de archivos de zona

- Ubicación típica:
  - /var/named/ en sistemas Unix/Linux.
  - %systemroot%\system32\drivers\etc en Windows.
- Estructura de subdirectorios:
  - /var/named/master → archivos de zona maestros.
  - /var/named/slave → archivos de zona esclavos.
  - /var/named/views → cuando se utilizan vistas.
- Nombres de archivos:
  - Maestros: master.example.com, master.sub-domain.example.com.
  - Esclavos: slave.example.com, slave.sub-domain.example.com.
  - Servidor raíz: root.servers (ej. named.ca o named.root en BIND).
  - Mapeo inverso: número de subred + .rev (ej. 192.168.23.rev para 23.168.192.IN-ADDR.ARPA).
  - Zona localhost:
    - Directo: master.localhost (ej. localhost.zone).
    - Inverso: localhost.rev (ej. named.local).
- Se recomienda aplicar rigurosamente las convenciones de nombres y no depender únicamente de las de la distribución que se use.

Tipos de registros DNS (A, AAAA, CNAME, MX, NS, PTR, SOA, TXT)

Generalidades

- La información de las zonas se almacena en ficheros de zona (bases de datos) que contienen nombres de equipos, servidores de correo, servicios, configuraciones DNS y subdominios delegados.
- Formato estandarizado en RFC 1035, lo que asegura interoperabilidad entre distintos software DNS.
- Cada entrada contiene 5 campos: Nombre_dominio, TTL, Clase, Tipo, Dato_Registro.
- Nombre_dominio: puede omitirse, tomando el último declarado.
- Clase: actualmente solo se usa IN (Internet).

- Los registros de recursos (RRs) describen las características de una zona o dominio y existen en:
  - Formato texto (archivos de zona): owner-name ttl class type type-specific-data.
  - Formato binario (consultas/respuestas): name ttl class type rdlen rdata.

Registros principales

- SOA (Start of Authority): primer registro de una zona. Define:
  - Servidor primario de la zona.
  - Contacto administrativo (correo sin @).
  - Número de serie (AAAAMMDDSS).
  - Intervalos: actualización, reintento, expiración.
  - TTL por defecto.

  Es esencial actualizar el número de serie cada vez que cambie cualquier registro de zona.
- NS (Name Server): define servidores de nombres autoritativos del dominio o subdominio.
- A: dirección IPv4 (32 bits).
- AAAA: dirección IPv6. Recomendado por la IETF.
- CNAME: alias que apunta al nombre canónico verdadero. Redirección válida solo para un RR en el owner-name.
- MX: identifica servidores de correo. Incluye preferencia y nombre de host. Definido en RFC 974.
- PTR: puntero que asocia una IP con un nombre de host (zonas inversas).
- TXT: texto libre. Usado para comentarios y configuraciones como SPF y DKIM.
  - SPF (RFC 7208): debe definirse en registros TXT (no en SPF, ya obsoleto).
  - DKIM: DomainKeys Identified Mail (DKIM) es un método de autenticación de correo electrónico que ayuda a evitar que los spammers y otros elementos maliciosos se hagan pasar por un dominio legítimo.

Otros registros

- HINFO: información opcional sobre hardware y SO.
- WKS: lista de servicios conocidos. Deprecado en favor de SRV.
- SRV: define servicios disponibles en la zona (ej. LDAP, HTTP, SIP). Usado en Active Directory.
- Registros adicionales (selección):
  - A6 (RFC 6563): obsoleto. Sustituido por AAAA.
  - AFSDB (RFC 1183): ubicación de servidores AFS. Experimental.
  - DNAME (RFC 6672): redirección de todo un subárbol DNS.
  - DNSKEY, DS, NSEC, RRSIG: usados en DNSSEC.
  - LOC (RFC 1876): coordenadas geográficas (experimental).
  - NAPTR (RFC 3403): reglas para aplicaciones DDDS (VoIP, ENUM).
  - RP (RFC 1183): persona responsable. Experimental.
  - SPF (RFC 4408): obsoleto, sustituido por TXT.
  - URI (RFC 7553): devuelve cadena URI completa para un servicio.
  - Otros: EUI48/EUI64, KEY, NXT (obsoleto), RT, SIG (obsoleto), X25.

Directivas de archivos de zona

- $ORIGIN: define el origen de nombres en el archivo.
- $INCLUDE: incluye otro archivo.
- $TTL: TTL predeterminado de los registros (RFC 2308, obligatorio en BIND 9).
- $GENERATE: directiva no estándar de BIND.

Notas de formato

- Los comentarios comienzan con ; y llegan hasta el final de línea.
- Los RRs ocupan una línea cada uno, salvo que usen paréntesis.
- Separadores: espacios o tabuladores.
- El primer registro siempre debe ser SOA.

Esta práctica debe realizarse en casa desde la máquina virtual de Debian instalada

1. Consulta qué servidor DNS tienes configurado en la máquina. Anótalo e intenta saber qué compañía lo gestiona (quizá al final de la práctica te resulte más sencillo) Consulta también el servidor DNS que tienes configurado en tu máquina anfitrión. Si no es el mismo reflexiona por qué.
2. Los siguientes son servidores DNS públicos y conocidos. Se trata de servidores a los que podemos interrogar y quizá alguno de ellos sea el que tienes configurado en tu máquina. También es probable que tu máquina tenga su propio servicio DNS para redirigir las consultas y cachear los resultados (clientes Ubuntu más modernos). ¿Por qué están puestos por parejas?
   - Telefónica: 80.58.61.250/80.58.61.254
   - Google: 8.8.8.8/8.8.4.4
   - OpenDNS: 208.67.222.222/208.67.220.220
3. Existen múltiples rutinas que permiten hacer consultas DNS, que en la mayoría de los casos existe para Linux y para Windows:

- host (observar la utilización de –r para evitar la recursividad).
- nslookup (sobrevive en sistemas Windows)
- dig (potente y versátil)
- En la página de Zytrax dispones de información al respecto de las dos últimas
- En esta práctica vamos a utilizar host por ser nativo y estar disponible en cualquier sistema Linux
- Si ejecutas el comando "host" sin ningún argumento obtendrás como resultado la sintaxis del comando (también puedes recurrir a 'man')

4. Realiza las siguientes consultas DNS, relaciona la sintaxis de las mismas con la sintaxis del comando host y observa y extrae conclusiones sobre los resultados:
   1. host google.com. ¿Por qué da como resultado más de uno?
   2. host lasalleinstitucion.sallenet.org. ¿Por qué el resultado está compuesto de 2 en realidad?
5. Ejecuta las consultas anteriores con el parámetro "-t A" (después de host) Después repite estas mismas consultas cambiando el DNS al que interrogas y poniendo el específico de Google o de OpenDNS y compara los resultados con los de la pregunta 4 (host -t A google.com 8.8.8.8)
6. Inicia "Wireshark" para poder capturar las siguientes capturas DNS (filtro DNS) y observa las preguntas y respuestas que se producen en la capa de aplicación:
   - host -t AAAA elmundo.es
   - host -t MX elmundo.es
   - host -t A elmundo.es
   - host -t CNAME elmundo.es
   - host -t NS elmundo.es
   - host -t NS elmundo.es ns1-02.azure-dns.com
7. Accede a la web root-servers.org y observa qué servidores DNS raíz hay en Madrid, quién los mantiene y cuál es su dirección IP. Compara la de uno de ellos con la de otro que se encuentre en algún lugar de Estados Unidos, ¿qué cambia?
8. Ejecuta las siguientes consultas, de manera consecutiva, observa las respuestas que dan y reflexiona por qué se produce la misma
   - host -r -a lasalleinstitucion.sallenet.org c.root-servers.net
   - host -r -a lasalleinstitucion.sallenet.org c0.org.afilias-nst.info
   - host -r -a lasalleinstitucion.sallenet.org dns18.ovh.net
   - host -r -t A lasalleinstitucion.sallenet.org dns18.ovh.net
   - host -r -t A lasalleinstitucion.sallenet.org 8.8.8.8
   - host -t A lasalleinstitucion.sallenet.org 8.8.8.8
9. Hay muchas utilidades web que hacen consultas DNS, entre ellas la de Google: https://toolbox.googleapps.com/apps/dig/
   - Prueba de nuevo a consultar alguno de los dominios anteriores
   - ¿Qué información se muestra?
   - ¿Qué tipos de registros son los más comunes?

4.Contenidos

Primeros pasos

1. ¿Es necesario actualizar el sistema?

Puesto que nosotros acabamos de instalar el sistema no es necesario. En caso de que instalemos pasado cierto tiempo es importante actualizar los repositorios y el software ya instalado antes de instalar cualquier otra cosa. Ahora bien, siempre se hará con precaución, sobre todo si hemos manipulado el archivo 'sources.list'.

El código para hacerlo sería:

```
sudo apt update
sudo apt upgrade -y
```

2. ¿Cómo instalar BIND9?,?

El código para instalar el servidor principal de BIND9 es:

```
sudo apt install bind9
```

Para instalar las herramientas de administración, utilidades y documentación asociada:

- bind9utils: herramientas de administración.
- bind9-doc: documentación de BIND.
- dnsutils: utilidades de prueba (ej. dig).

```
sudo apt install bind9utils bind9-doc dnsutils -y
```

3. ¿Cómo verificar el estado del servicio?

Utilizaremos systemctl igual que lo hemos hecho con systemd-networkd, por ejemplo para ver el estado, iniciar, parar, reiniciar y habilitar el servicio para que se inicie de forma automática:

```
sudo systemctl status bind9
sudo systemctl start bind9
sudo systemctl stop bind9
sudo systemctl restart bind9
sudo systemctl enable bind9
```

4. ¿Cuáles son los archivos principales de configuración?

- /etc/bind/named.conf: configuración principal.
- /etc/bind/named.conf.local: definición de zonas y ajustes locales.
- /etc/bind/named.conf.options: opciones globales.

¿Qué contiene el archivo de configuración principal named.conf?

En general su contenido es un conjunto de 'includes' que permiten llamar a otros archivos de configuración para facilitar la compresión:

¿Qué contiene el archivo /etc/bind/named.conf.options?

Opciones globales. ¿Para qué sirven las opciones allow-query, dnssec-validation, auth-nxdomain y listen-on-v6?

¿Qué contiene el archivo /etc/bind/named.conf.local?

Contiene las zonas que serán directamente gestionadas por nuestro servidor. Inicialmente estará vacío porque no tenemos ninguna zona gestionada que hayamos añadido.

¿Qué contiene el archivo /etc/bind/named.conf.default-zones?

Las zonas, que por defecto se crean al instalar BIND9. Fijaos que algunas "sustituyen" a localhost, que hay zona de mapeo inverso para localhost, etc. Estas zonas están catalogadas como maestras (master) y se indica dónde se encuentran alojadas las mismas con el parámetro 'file'. Revisa todos los archivos de esas zonas y encuentra qué tienen en común.

La zona "." son los servidores Raíz y que está marcada como "hint".

Configuración de un servidor autoritativo para una zona directa

Método clásico de actualización

- Consiste en editar manualmente el archivo de zona y reiniciar el servidor para propagar cambios.
- No es práctico cuando el volumen de cambios es elevado.

Servidor DNS Maestro

- Define uno o más archivos de zona para los que es autoritativo (type master).
- La zona debe estar delegada mediante un registro NS hacia este servidor en el caso de tratarse de un dominio público. Recordemos el principio de autoridad.

Comportamiento de un servidor no autoritativo

Depende de la configuración (named.conf en BIND):

- Con caché y consultas recursivas permitidas: responde completamente o devuelve error.
- Con caché y consultas iterativas permitidas: responde con datos de caché, una referencia o error.
- Sin caché (solo autoritativo): devuelve una referencia o error.

Se podría configurar un servidor DNS maestro con caché y recursividad pero limitado solamente a un conjunto de IPs locales, lo que significaría que el DNS está 'cerrado'.

Qué diferencias hay entre tener un DNS abierto y tenerlo cerrado

- DNS Abierto: permite consultas recursivas desde cualquier usuario. Muy desaconsejado puesto que puede ser usado en ataques DDoS y aumenta el riesgo de envenenamiento de caché.
- DNS Cerrado: la recursión solo está permitida para un rango definido de direcciones IP.

Ejemplo de configuración

Servidor de nombres maestro con caché y allow-recursion limitado a IPs locales (DNS cerrado).

Buenas prácticas de seguridad

- Desplegar solo los servicios estrictamente necesarios.
- Un servidor DNS seguro debería cumplir una sola función:
  - Solo autoritativo
  - o solo caché
- En la práctica, muchos entornos combinan ambas funciones en un servidor mixto.

Edición de archivos de zona y configuración en BIND

Archivos de Zona

- Comentarios:
  - Comienzan con punto y coma (;) hasta el final de línea.
- Directivas:
  - Inician con $ (ejemplo: $TTL, $ORIGIN).
- Registros de recursos (RR):
  - Usan una línea por registro.
  - Excepciones con paréntesis para entradas largas.
- Separadores: espacios o tabuladores.

Archivo named.conf

- Controla el comportamiento y funcionalidad de BIND.
- Es el único archivo utilizado (ignorar referencias a boot.conf de BIND 4).
- Ubicación típica según SO:
  - FreeBSD: /etc/namedb o /usr/local/etc (instalación no base).
  - Unix/Linux: /etc.
  - Windows: \Program Files\ISC BIND 9\etc o \Windows\system32\dns\etc en versiones antiguas.
- Incluye lista de declaraciones y opciones disponibles (ubicadas en /usr/share/docs/bind-version/misc/options en FC o /usr/src/contrib/bind9/doc/misc/options en FreeBSD).
- BIND soporta comentarios en:
  - Estilo C: /* ... */.
  - Estilo C++: //.
  - Estilo Perl/Shell: #.
- Validación:
  - El archivo completo se analiza antes de usarse.
  - Errores previos a cláusula logging van a syslogd (/var/log/messages).
- Rigurosidad:
  - BIND exige precisión en paréntesis, corchetes, comas y puntos y comas.

Estructura General de Cláusulas

- Sin views:
  1. acl (si se requieren).
  2. logging.
  3. options.
  4. Otras cláusulas según necesidad.
  5. zone (incluyendo las requeridas).
- Con views:
  1. key (si son necesarias).
  2. acl (si son necesarias).
  3. logging.
  4. options.
  5. view (cada una con sus propias options y zone).

Archivos de Zona Requeridos

- root.servers:
  - Aplica solo si el servidor soporta consultas recursivas.
  - Define lista de servidores raíz (a.root-servers.net a m.root-servers.net).
  - Configurado como zone "." type hint.
  - No necesario en servidores solo autoritativos o redes internas cerradas.

    - BIND 9 incluye una lista interna por defecto.
- localhost:
    - Resuelve localhost a 127.0.0.1.
    - Usado por aplicaciones locales; su ausencia genera retardos y consultas externas innecesarias.
    - Se define como type master.
    - allow-update { none; } es el valor predeterminado (opcional pero recomendable).
- reverse-map:
    - Traduce direcciones IP a nombres (dominio IN-ADDR.ARPA para IPv4).
    - La falta de zonas inversas privadas (RFC 1918) provoca consultas innecesarias a la jerarquía DNS.
    - BIND incluye empty-zones-enable yes por defecto para mitigarlo.
    - Ejemplo: 0.0.127.IN-ADDR.ARPA mapea 127.0.0.1 a localhost.

Cláusulas Soportadas por BIND

acl, controls, dlz, include, key, logging, lwres, managed-keys, masters, options, server, statistics-channels, trusted-keys, view, zone.

Clasificación de Declaraciones

- Consultas
- Transferencia
- Operaciones
- Seguridad
- Estadísticas

Pruebas locales con dig y nslookup

Consultas básicas

- Al consultar el nombre localhost., se obtienen resultados específicos.
- El comando host -r -a a.root-server.net 127.0.0.1 obtiene valores del archivo /usr/share/dns/root.hints.

nslookup

- Deprecado oficialmente en favor de dig, pero aún ampliamente disponible (especialmente en Windows).
- Generalmente devuelve registros A o PTR, aunque puede modificarse con opciones específicas.
- Modos de uso:
    - Línea de comandos: ejemplos:
        - nslookup www.example.com → buscar un host específico.
        - nslookup -type=ANY example.com → obtener registros MX y NS.
        - nslookup -all -type=SOA example.com → obtener registro SOA y parámetros por defecto.
    - Interactivo:
        - Formato genérico: nslookup [-opt] target [dns] o nslookup [-opt] [-] [dns].
        - Prompt único > para introducir comandos.
        - Salida con CTRL-C (Windows/*nix), CTRL-D (*nix) o exit.
        - Permite modificar parámetros de configuración (-all o set all).
- Ofrece múltiples opciones que alteran el procesamiento, algunas exclusivas del modo interactivo.

dig

- Proporciona información más detallada y útil que nslookup para administradores.
- Se incluye con la instalación de BIND en Windows.

Herramientas de validación y administración

- named-checkzone: valida archivos de zona para verificar su corrección.
- named-checkconf: verifica la configuración de named.conf.
- rndc:
    - Herramienta de acceso remoto para recarga selectiva de zonas.
    - Conlleva implicaciones de seguridad graves si no se configura correctamente.
- nsupdate:
    - Permite actualizaciones dinámicas (DDNS) de archivos de zona.
    - Requiere extrema precaución: una mala configuración puede exponer archivos de zona al mundo.

Práctica 1

Requisitos:

- Tener instalado el servidor BOOKWORMXXB con la configuración de red doble: DHCP en modo puente y estática (10.0.128+XX.2) en modo red interna
- Recuerda que tienes a tu disposición la documentación de Zytrax donde puedes consultar sobre conceptos y sintaxis
- **Durante todo el proceso de la práctica deberán hacerse capturas de imagen de cada uno de los pasos dados así como capturas y almacenaje de los paquetes de wireshark (recuerda que puedes filtrar los paquetes DNS y guardar solamente como archivo .pcap lo filtrado) . Algunos de ellos serán solicitados por el profesor una vez concluida la práctica.**

Pasos:

- BIND9 es el servidor DNS del ISC que vamos a utilizar. Instálalo en tu máquina BOOKWORMXXB utilizando apt. Observa la versión que se instala y la actual versión estable accediendo al sitio web del ISC.
- Comprueba si el servicio está arrancado y/o si tiene errores. Puedes reiniciarlo y volver a probar.
- Para evitar problemas, de momento, vamos a usar la propia máquina BOOKWORMXXB como cliente también. Para ello especificaremos que el servidor que resuelve sea él mismo, es decir, localhost o 127.0.0.1. Una consulta sobre www.google.com utilizando la utilidad 'host' será: `host www.google.com 127.0.0.1`
- Arranca Wireshark en BOOKWORMXXB en el interfaz de red interna (también lo puedes hacer en el otro) y captura los mensajes DNS que se producen cuando haces la siguiente consulta `host elpais.com 127.0.0.1`. Observa la respuesta en consola. Observa si se producen mensajes DNS en wireshark. Si hay mensajes busca explicación a cada uno de los mensajes. Si no hay mensajes, explica las razones por las que esto se produce.
- ¿Qué ocurre si preguntamos por "servidor.asirXX.asir."? ¿Por qué? Responde del mismo modo que la pregunta anterior.

Práctica 2:

Requisitos:

- Haber hecho la práctica 1 o, en su defecto, tener, al menos instalado el servidor DNS BIND9
- **Durante todo el proceso de la práctica deberán hacerse capturas de imagen de cada uno de los pasos dados así como capturas y almacenaje de los paquetes de wireshark. Algunos de ellos serán solicitados por el profesor una vez concluida la práctica.**

Antes de empezar:

- Recuerda que tienes a tu disposición la documentación de Zytrax donde puedes consultar sobre conceptos y sintaxis.
- Sustituye el valor 'XX' por tu número de puesto asignado que venimos utilizando
- Toma nota de todo lo que vayas haciendo, incluyendo errores y soluciones sobre los mismos, así como capturas
- Para poder hacer consultas DNS a tu servidor Debian BOOKWORMXXB desde el propio servidor indícalo de forma expresa en los comandos 'host'.

Pasos:

1. Comprueba el estado del servicio DNS utilizando el comando sobre consola correspondiente
2. Con el servicio arrancado, interroga a tu servidor, utilizando el comando host, por ejemplo `host -a www.google.com 127.0.0.1` (El 127.0.0.1 indica quien resuelve el nombre). ¿Hay alguna diferencia entre hacerlo desde el colegio o hacerlo desde casa?
3. Crea una zona de resolución directa maestra denominada "asirXX.asir", donde XX es tu número de lista.
   - El nombre de la zona debe ser el del dominio "asirXX.asir" y el del archivo que la contiene "asirXX.asir.hosts" (ten cuidado con los permisos del fichero).
   - El SOA deberá tener como valores:
     - Correo electrónico: uno con sentido (sustituyendo la @ por un .)
     - Número de serie acorde con el formato AAAAMMDDNN
     - Refresco de la zona cada hora (en segundos)
     - Reintento en caso de no poder transferir la zona cada 10 minutos (en segundos)
     - ¿Cuándo debe de ofrecer autoridad sobre los registros? Pasadas 2 semanas (en segundos)
     - TTL por defecto de 3600 segundos
   - El Servidor Maestro es el propio servidor DNS BOOKWORMXXB, por lo que habrá que crear el registro NS correspondiente.
4. Ve introduciendo los siguientes registros de uno en uno. Según los introduzcas recarga el servicio para hacer las correspondientes comprobaciones:
   - Registro NS que apunte a tu máquina BOOKWORMXXB, pues será él mismo el servidor DNS y propietario de la zona que estamos creando
   - Registro A para tu máquina con la dirección correspondiente como servidor según venimos utilizando: IP 10.0.128+X.2 para bookwormXXb.asirXX.asir. Se cuidadoso con el nombre. Si lo cualificas acábalo en punto, si no lo cualificas no pongas el punto al final.
   - Registro A para 'bookwormXX' a la dirección correspondiente a tu red.
   - Registro A para una máquina denominada 'servidor' en la dirección IP 10.0.128+XX.200 de tu red
   - Registro AAAA para una máquina denominada 'servidor' en la dirección IP fe80:XX:XX::XX+200 /48
   - Registro CNAME para www que se traduzca por el nombre de tu máquina 'bookwormXXb'
   - Registro MX para 'correo.asirXX.asir' que se traduzca por el nombre de tu máquina con prioridad 10

5. Haz pruebas desde tu servidor para comprobar si las interrogaciones funcionan correctamente (recuerda la Práctica 0). Interroga específicamente sobre los tipos de registro, por ejemplo: `host -t CNAME www.asirXX.asir. 127.0.0.1`
6. Haz pruebas desde tu máquina cliente LINUXXX para comprobar si las interrogaciones funcionan correctamente (recuerda la Práctica 0). Para ello:
    1. Modifica la configuración de red del cliente LINUXXX para que el DNS sea el 10.0.128+XX.2
    2. Interroga con comandos análogos a los del apartado 5 pero sin poner el 127.0.0.1 del final para que sea el valor de DNS configurado el que determine quién resuelve los nombres.

NOTA: Recuerda que los registros log del servidor se pueden localizar utilizando `journalctl`. Puedes usar grep para hacer búsquedas concretas de texto y parámetros de tiempo con `--since=` y `--until=` para acotar la búsqueda además de otros muchos que puedes encontrar en la página de ayuda de `journalctl.` En ocasiones el servidor puede funcionar a pesar de tener errores en alguna o algunas zonas, en este caso no las carga pero sigue funcionando con el resto de zonas.

Práctica 3:

Requisitos:

- Haber hecho la práctica 2.
- **Durante todo el proceso de la práctica deberán hacerse capturas de imagen de cada uno de los pasos dados así como capturas y almacenaje de los paquetes de wireshark. Algunos de ellos serán solicitados por el profesor una vez concluida la práctica.**

Antes de empezar:

- Recuerda que tienes a tu disposición la documentación de Zytrax donde puedes consultar sobre conceptos y sintaxis.
- Sustituye el valor 'XX' por tu número de puesto asignado que venimos utilizando
- Toma nota de todo lo que vayas haciendo, incluyendo errores y soluciones sobre los mismos, así como capturas
- Debes utilizar el cliente LinuxXX para hacer todas las consultas DNS de prueba, por lo que deberá estar correctamente configurado ~~vía el DHCP de BOOKWORMXX~~ **utilizando alguno de los modos de configuración vistos, preferiblemente systemd-networkd**
- **Recuerda que cada vez que actualices el fichero de zona tienes que incrementar el número de serie del SOA**
- **Puede instalar el paquete 'net-tools' para poder usar 'netstat' y con ello todas las características del mismo, por ejemplo observar los puertos abiertos.**
- **Puede resultarte útil utilizar el interfaz 'any' en Wireshark para poder hacer capturas en todas las interfaces simultáneamente.**

Prueba:

- Además de utilizar el comando 'host' vamos a comenzar a utilizar el comando 'dig' (en el caso de no disponer de él deberás instalarlo). La sintaxis más sencilla es `dig servidor.asirXX.asir`, por ejemplo. Puedes consultarlo de manera completa usando el 'man' o la documentación disponible en Zytrax. Observa cómo serían las consultas no recursivas, cómo se especificaría un tipo concreto de registro o cómo hacer que la consulta se haga a un servidor concreto.
- Debes realizar las pruebas desde el cliente LinuxXX, al que deberás haber configurado tu DNS como el DNS que debe utilizar LinuxXX para la resolución de nombres.
- Dispones además de comandos de comprobación de sintaxis para el BIND9:
    - Podemos comprobar un archivo de configuración con, por ejemplo: `named-checkconf /path/to/named.conf`
    - Podemos comprobar una zona con, por ejemplo: `named-checkzone example.net /etc/bind/example.net`

Pasos:

1. Crea un subdominio (subzona, sin delegación) denominado "subasirXX.asirXX.asir" dentro de la propia zona creada en la práctica anterior. Utiliza para ello la directiva `$origin`
2. Crea los siguientes registros adicionales debajo de la directiva $origin (Puedes seguir el ejemplo que se encuentra en el "howto" del apartado 9 de Zytrax al respecto de subdominios):
    1. servidor1 debe resolverse como un alias de servidor.asirXX.asir (es decir servidor1.subasirXX.asirXX.asir es un alias de servidor.asirXX.asir)
    2. ftp debe resolverse con la IP 10.0.128+XX.2
3. Recarga el servicio y haz pruebas para comprobar que se resuelven los nombres tanto del dominio como del subdominio correctamente.
4. Realiza consultas DNS desde el cliente haciendo capturas de ello como en prácticas anteriores.

Práctica 3 (parte 2):

Requisitos:

- Haber hecho la primera parte de la práctica 3
- **Durante todo el proceso de la práctica deberán hacerse capturas de imagen de cada uno de los pasos dados así como capturas y almacenaje de los paquetes de wireshark. Algunos de ellos serán solicitados por el profesor una vez concluida la práctica.**

Antes de empezar:

- Recuerda que tienes a tu disposición la documentación de Zytrax donde puedes consultar sobre conceptos y sintaxis.
- Sustituye el valor 'XX' por tu número de puesto asignado que venimos utilizando
- Toma nota de todo lo que vayas haciendo, incluyendo errores y soluciones sobre los mismos, así como capturas
- Debes utilizar el cliente LinuxXX para hacer todas las consultas DNS de prueba, por lo que deberá estar correctamente configurado **utilizando alguno de los modos de configuración vistos, preferiblemente systemd-networkd**.
- **Recuerda que cada vez que actualices el fichero de zona tienes que incrementar el número de serie del SOA**
- Puede instalar el paquete 'net-tools' para poder usar 'netstat' y con ello todas las características del mismo, por ejemplo observar los puertos abiertos.
- Puede resultarte útil utilizar el interfaz 'any' en Wireshark para poder hacer capturas en todas las interfaces simultáneamente.

Prueba:

- Todas las pruebas deben realizarse desde el cliente LinuxXX utilizando **dig.**
- **Para que nuestro cliente LinuxXX tenga el DNS deseado es importante evitar que tenga habilitada la interfaz externa. La interna, en caso de hacerlo con systemd-networkd, en el archivo de configuración correspondiente a 'enp0s8' habría añadir DNS=10.0.128+XX.2**
- **Aunque para esta práctica no es necesario, si queremos que nuestro servidor DNS no tenga habilitado cliente DNS podemos añadir en la configuración de nuestro interfaz externo la configuración:**

```
[DHCPv4]
UseDNS=false

[DHCPv6]
UseDNS=false
```

- Recuerda que dispones además de comandos de comprobación de sintaxis para el BIND9:
  - Podemos comprobar un archivo de configuración con, por ejemplo: `named-checkconf /path/to/named.conf`
  - Podemos comprobar una zona con, por ejemplo: `named-checkzone example.net /etc/bind/example.net`

Pasos:

1. Crea una zona MAESTRA de resolución inversa con los registros PTR correspondientes a las IPs que hemos tratado en las anteriores prácticas.

2. El nombre de la zona será '128+XX.0.10.in-addr.arpa.rev'. Será también una zona maestra con los mismos registros SOA y NS que la de las prácticas anteriores (excepto el número de serie en el caso del SOA). Recuerda que aquello que no cualifiques será cualificado con "128+XX.0.10.in-addr.arpa".

3. Los registros adicionales que deben crearse son PTR

4. ¿Cómo sería la consulta con dig para resolver la IP 10.0.128+XX.200? Prueba que funciona

5. El nombre de la zona y por tanto el sufijo indica el rango de red. ¿Cómo sería si en lugar de /24 fuera /26?

Recordando...

El objetivo principal del diseño del sistema de nombres de dominio fue una administración descentralizada. Esta descentralización se hace a través de la delegación. Ahora bien, delegación no significa independencia, sino coordinación.

¿Cómo responde el dominio padre?

Si al dominio padre se le plantean consultas sobre nombres incluidos en sus subdominios delegados, puede hacer referencia a dichos subdominios.

¿Quién hace la delegación?

- Una organización que administra un dominio puede dividirlo en subdominios y delegarlos.
- Cada subdominio puede ser delegado a diferentes organizaciones, lo que implica que esa organización será responsable de mantener los datos (registros de recursos) de ese subdominio.
- El dominio padre solamente contiene enlaces a los responsables del subdominio delegado, de forma que pueda hacer referencia a ellos cuando se le planteen consultas sobre nombres en dicho subdominio delegado.
- La delegación es una acción que siempre se puede revocar.

¿Qué parte del nombre se puede controlar?

- Al leer un nombre de dominio de DERECHA a IZQUIERDA, se puede rastrear su delegación. Esta unidad de delegación también puede denominarse zona.
- El propietario de un "Domain Name" controla todo lo que está a la IZQUIERDA de su nombre delegado y es responsable de administrar esta delegación (ejecutando un DNS con información Autoritativa para su "Domain Name" o zona).
- Un nombre de subdominio es un nombre arbitrario asignado por el propietario del "Domain Name".
- Un nombre de subdominio incluirá completamente el "Domain Name" (por ejemplo, es.example.com es un nombre de subdominio válido de example.com).
- Técnicamente, el nombre de un subdominio también es un "Domain Name".

Práctica 4:

Requisitos:

- Haber hecho la práctica 3.
- **Durante todo el proceso de la práctica deberán hacerse capturas de imagen de cada uno de los pasos dados así como capturas y almacenaje de los paquetes de wireshark. Algunos de ellos serán solicitados por el profesor una vez concluida la práctica.**

Otra forma de conseguir que el cliente LinuxXX funcione con systemd-resolved

Puedes mantener systemd-resolved activado y decirle qué servidor DNS quieres que se utilice y que no use el DNS STUB. Para ello, en el archivo /etc/systemd/resolved.conf tienes que añadir las líneas:

```
DNS=10.0.128+XX.2

DNSStubListener=no
```

Antes de empezar:

- Recuerda que tienes a tu disposición la documentación de Zytrax donde puedes consultar sobre conceptos y sintaxis.
- Sustituye el valor 'XX' por tu número de puesto asignado que venimos utilizando
- Toma nota de todo lo que vayas haciendo, incluyendo errores y soluciones sobre los mismos, así como capturas
- Debes utilizar el cliente LinuxXX para hacer todas las consultas DNS de prueba
- Recuerda que cada vez que actualices un fichero de zona tienes que incrementar el número de serie del SOA
- **Para poder trabajar mejor es necesario instalar wireshark en el cliente LinuxXX**

Prueba:

- Hay que utilizar el comando 'dig' para todas la pruebas
- Dispones además de comandos de comprobación de sintaxis para el BIND9:
  - Podemos comprobar un archivo de configuración con, por ejemplo: **named-checkconf /path/to/named.conf**
  - Podemos comprobar una zona con, por ejemplo: **named-checkzone example.net /etc/bind/example.net**

Pasos:

1. En la zona 'asirXX.asir' crea un nuevo un subdominio denominado "delegadoasirXX.asirXX.asir" del mismo modo que en la práctica anterior. Utiliza para ello la directiva $origin.
2. Asigna a ese subdominio como registro NS el DNS servidor.asirXX.asir (se trata de un DNS ficticio, que no existe) y vuelve a cargar la zona. Para ello tendrás que habilitar un registro NS específico para 'delegadoasirXX.asirX.asir'. Para ello, después de la directiva $origin, usa la sintaxis:
   - @ IN NS servidor.asirXX.asir.
   - No añadimos ningún registro más
3. Vamos a consultar desde nuestro cliente Linux el nombre "prueba.delegadoasirXX.asirXX.asir" usando el comando dig. Si no indicamos nada de forma expresa, la pregunta **recursiva**. Observa qué ocurre. Observa la respuesta que se da. Observa la captura de paquetes que se produce en la interfaz de la red interna del cliente LinuxXX y extrae conclusiones. Recuerda por filtrar por DNS para que sea más sencillo.
4. Pregunta por "prueba.delegadoasirX.asirXX.asir" usando el comando dig, pero haciendo que la pregunta sea **NO recursiva**. Observa qué ocurre y la respuesta que se da. Observa la captura de paquetes que se produce en la interfaz de la red interna del cliente LinuxXX y extrae conclusiones. Recuerda por filtrar por DNS para que sea más sencillo.
5. Extrae conclusiones sobre el concepto de delegación

Práctica 5:

Requisitos:

- Haber hecho la práctica 4.
- **Durante todo el proceso de la práctica deberán hacerse capturas de imagen de cada uno de los pasos dados así como capturas y almacenaje de los paquetes de wireshark. Algunos de ellos serán solicitados por el profesor una vez concluida la práctica.**

Antes de empezar:

- Recuerda que tienes a tu disposición la documentación de Zytrax donde puedes consultar sobre conceptos y sintaxis.
- Sustituye el valor 'XX' por tu número de puesto asignado que venimos utilizando
- Toma nota de todo lo que vayas haciendo, incluyendo errores y soluciones sobre los mismos, así como capturas
- Debes utilizar **tanto el cliente LinuxXX como el cliente WindowsXX** para hacer todas las consultas DNS de prueba
- Recuerda que cada vez que actualices un fichero de zona tienes que incrementar el número de serie del SOA

Prueba:

- Hay que utilizar el comando 'dig' para todas la pruebas
- Dispones además de comandos de comprobación de sintaxis para el BIND9:
  - Podemos comprobar un archivo de configuración con, por ejemplo: **named-checkconf /path/to/named.conf**
  - Podemos comprobar una zona con, por ejemplo: **named-checkzone example.net /etc/bind/example.net**
  - **Dispones de muchos otros comandos que te pueden resultar útiles en el apartado 15 del manual de BIND9**

Pasos:

1. Preparación y acondicionamiento de la máquina BOOKWORMXXA como servidor DNS
   - A partir de la máquina virtual clonada en origen crea una nueva máquina llamado BOOKWORMXXA, con características análogas a BOOKWORMXXB, pero con la dirección IP 10.0.128+XX.1/24 y el correspondiente cambio de nombre.
   - Con respecto a la configuración de red, usa systemd-networkd para la configuración:
     - En el adaptador externo (enp0s3) desactiva el uso del DNS ofrecido por el servidor DHCP (UseDNS=false)
     - En el adaptador interno (enp0s8) especifica como DNS la propia máquina inicialmente (DNS=10.0.128+XX.1)
   - Con respecto a la configuración DNS hazla con systemd-resolved. En /etc/systemd/resolved.conf especifica:
     - Como servidor el 10.0128+XX.2
     - Deshabilita el DNSStubListener poniendo `DNSStubListener=no`
   - Instala el servidor BIND9 en la máquina
   - Comprueba el estado del servicio utilizando los comandos que ya conoces
2. Creación de la zona "delegadoasirXX.asirXX.asir."
   - Crea el archivo de zona en el directorio correspondiente y modifica el archivo de configuración correspondiente para añadir la zona.
   - La zona creada será maestra y de resolución directa.
   - El SOA es el servidor bookwormXXa.asirXX.asir.
   - Debe tener un registro NS que apunte al servidor bookwormXXa.asirXX.asir., pero no necesita un registro A que resuelva dicho nombre por la IP del servidor (10.0.128+XX.1). Este registro A deberá estar en la zona "asirXX.asir".
   - Debe tener un registro CNAME que permita resolver el nombre www.delegadoasirXX.asirXX.asir. por el nombre del servidor 'bookwormXXa'
   - Debe tener un registro A que permita resolver ftps.delegadoasirXX.asirXX.asir. por la IP 10.0.0.1
3. En el servidor 'bookwormXXb' debes modificar la zona "asirXX.asir" para que el subdominio "delegadoasirXX" se delegue en el servidor 'bookwormXXa'. Para ello crea un registro NS que apunte a dicho servidor, según hemos visto en clase (bookwormXXa.asirXX.asir.). Además tendrás que crear un registro A para el servidor en la zona del fichero adecuada para que bookwormXXa.asirXX.asir se resuelva por su dirección IP (10.0.128+XX.1).Prueba que:
   - Existe conectividad entre las máquinas (2 servidores y 1 cliente)
   - Que desde el cliente se resuelven consultas que se hacen sobre registros de la zona "asirXX.asir" sobre el DNS de 'bookwormXX**b**'
   - Que desde el cliente se resuelven consultas que se hacen sobre registros de la zona delegada "delegadoasirXX.asirXX.asir" sobre el DNS de 'bookwormXXa'. Explicita en este caso en la consulta que el servidor consultado es el A.
4. Realiza las siguientes consultas poniendo previamente a capturar el tráfico en los interfaces correspondientes:
   1. Haz una consulta RECURSIVA al servidor b (10.0.128+XX.2) preguntando por "ftps.delegadoasirXX.asirXX.asir" y extrae conclusiones. Observa la captura de wireshark para ello.
   2. Haz una consulta NO RECURSIVA al servidor 2 (10.0.128+XX.2) preguntando por "ftps.delegadoasirXX.asirXX.asir" y extrae conclusiones. Observa la captura de wireshark para ello.
   3. Si la consulta 1 ha funcionado vuelve a realizarla y observa qué cambios ha habido en la misma. Observa la captura de wireshark para ello.

Recordando...

Un servidor DNS reenviador (también conocido como Proxy, Cliente, Remoto) reenvía solicitudes a otro DNS y almacena en caché los resultados. Esta configuración es útil en diversas situaciones:

- Cuando el acceso a la red externa es lento o costoso (el almacenamiento en caché local en el servidor reenviador proporciona resultados rápidos, y el servidor DNS remoto que soporta recursión reduce la congestión de tráfico y el volumen de tráfico).
- Para aliviar la carga de la administración local, proporcionando un único punto donde se pueden gestionar los cambios en los servidores de nombres remotos, en lugar de tener que actualizar todos los hosts.
- Para sanear el tráfico, especialmente en redes privadas grandes, es sensato dirigir el tráfico DNS para el acceso a dominios locales a los servidores DNS locales, mientras se reenvían las solicitudes DNS externas a un DNS de caché (o resolver).
- También se puede utilizar como parte de una configuración de servidor Split para la defensa perimetral.

Configuración en BIND

- BIND permite la configuración de reenvío utilizando los parámetros `forward` y `forwarders`, ya sea a nivel 'global' (en una sección options) o por zona (en una sección zone) del archivo named.conf o en cualquiera de sus archivos incluidos. En cualquiera de ambos casos la consulta se reenvía a otro servidor DNS, esperando la respuesta para entregársela al cliente que hizo la pregunta original y cacheando la misma:
- La opción `forward only` anulará el comportamiento de "recursive query", mientras que la opción `forward first` no lo hace
- En el archivo named.conf.options podemos establecer uno o más reenviadores. Por ejemplo:

```
forwarders {

     8.8.8.8;

};
```

Si queremos que para la zona "example.com" se reenvíen las consultas al servidor DNS 1.1.1.1, se configurará en el archivo named.conf.local un apartado como el siguiente:

```
zone "example.com" {
    type forward;
    forwarders { 1.1.1.1; 1.0.0.1; };
};
```

Práctica 6:

Requisitos:

- Haber hecho la práctica 3.
- **Durante todo el proceso de la práctica deberán hacerse capturas de imagen de cada uno de los pasos dados así como capturas y almacenaje de los paquetes de wireshark. Algunos de ellos serán solicitados por el profesor una vez concluida la práctica.**
- **Esta práctica ha de realizarse 2 veces. Una en clase y otra en casa, con configuraciones de reenvío diferente.**

Antes de empezar:

- Recuerda que tienes a tu disposición la documentación de Zytrax donde puedes consultar sobre conceptos y sintaxis.
- Sustituye el valor 'XX' por tu número de puesto asignado que venimos utilizando
- Toma nota de todo lo que vayas haciendo, incluyendo errores y soluciones sobre los mismos, así como capturas
- Debes utilizar el cliente **WindowsXX** para hacer todas las consultas DNS de prueba.
- Recuerda que cada vez que actualices un fichero de zona tienes que incrementar el número de serie del SOA.
- Recuerda que dispones de 'rndc' para poder interactuar con los servidores DNS

Prueba:

- Utiliza 'nslookup' para realizar las interrogaciones desde la máquina WINDOWSXX
- Dispones además de comandos de comprobación de sintaxis para el BIND9:
  - Podemos comprobar un archivo de configuración con, por ejemplo: named-checkconf /path/to/named.conf
  - Podemos comprobar una zona con, por ejemplo: named-checkzone example.net /etc/bind/example.net
  - Dispones de muchos otros comandos que te pueden resultar útiles en el apartado 15 del manual de BIND9, incluyendo la opción de interaccionar con los servidores DNS utilizando 'rndc'
  - **En ocasiones puede ser bueno liberar la caché del servidor con el fin de que no use las resoluciones que se hayan producido momentos antes. Para ello podemos usar 'sudo rndc flush'**

Pasos:

1. A partir de la configuración de la práctica 3, se pueden configurar reenviadores en el servidor B bien generales, bien para un dominio o dominios específicos.
2. La configuración para clase será: un reenviador general a la dirección 172.16.5.65 y un reenviador condicional para el dominio google.com al servidor 172.16.5.67. La configuración para casa será un reenviador general a la dirección 1.1.1.1 y un reenviador condicional para el dominio google.com al servidor 8.8.8.8.

3. Si no tenemos ningún reenviador configurado el servidor DNS instalado en nuestro servidor B buscará hacer consultas recursivas para obtener respuestas, cachearlas y poder ofrecerlas al cliente WindowsXX. Si hacemos la configuración correspondiente que aparece en el punto 2 y optamos por la opción forward only nuestro servidor DNS dejará de usar la recursividad y reenviará las consultas a los servidores DNS configurados como reenviadores.

Pruebas

1. Consultas DNS a un registro www.lasalle.es y a un registro mail.google.com
2. Mismas consultas anteriores pero forzando la no recursividad con `set norecurse`

Recordando...

- Es el proceso por el cual los servidores DNS actualizan la información de zona cuando no lo hacen desde un recurso local, sino desde otro DNS.
- El objetivo es simplificar el mantenimiento de múltiples servidores DNS, permitiendo que un único servidor (maestro) actualice al resto (esclavos) y mantener la coherencia entre las zonas replicadas en distintos servidores DNS
- Este proceso involucra la transferencia de ficheros de zona de un servidor maestro a otros esclavos.
- Está estandarizado por el protocolo DNS y se realiza sobre el puerto 53 utilizando TCP como transporte.
- Aunque las transferencias de zona son generalmente esenciales para el funcionamiento de los sistemas DNS, también son una fuente de amenaza. Un servidor de nombres esclavo puede ser envenenado si acepta transferencias de una fuente maliciosa. Por ello, se debe tener cuidado durante la configuración para asegurar que el 'esclavo' solo acepte transferencias de fuentes conocidas.

Configuración de transferencias de zona AXFR y IXFR:

Existen dos tipos de transferencias de zona: Completa (AXFR) e Incremental (IXFR).

AXFR (Full Zone Transfer):

- Las especificaciones DNS originales (RFC 1034 y RFC 1035) preveían que los servidores de nombres esclavos "sondearan" al maestro de dominio (o zona).
- El tiempo entre estos "sondeos" está determinado por el valor de actualización (refresh) del registro de recursos SOA del dominio.
- El proceso de sondeo se lleva a cabo mediante el envío por parte del esclavo de una consulta al maestro solicitando su registro de recursos SOA (RR) actual.
- Si el número de serie de este RR es mayor que el actual que mantiene el esclavo, se solicita una transferencia de zona (AXFR).
- Por este motivo, es fundamental ser muy disciplinado a la hora de actualizar el número de serie SOA cada vez que algo cambia en CUALQUIERA de los registros de zona.
- Las transferencias de zona siempre se realizan utilizando TCP en el puerto 53, mientras que las operaciones de consulta DNS normales utilizan UDP en el puerto 53.

IXFR (Incremental Zone Transfer):

- Permiten que el esclavo y el maestro transfieran solo aquellos registros que han cambiado.
- El proceso es el mismo que para AXFR, pero se marca con IXRF (request-ixfr).
- Podría darse el caso de que un servidor no soportara las transferencias incrementales, en cuyo caso se realizaría una AXFR.
- RFC 1995 introdujo las Transferencias de Zona Incrementales (IXFR) para ahorrar tiempo y ancho de banda al transferir archivos de zona muy grandes, especialmente si solo se ha cambiado un solo RR.
- El modo predeterminado para BIND cuando actúa como Esclavo es solicitar IXFR, a menos que se configure lo contrario utilizando el parámetro request-ixfr en la cláusula `server` u `options` del archivo named.conf.
- El modo predeterminado para BIND cuando actúa como Maestro es usar IXFR solo cuando la zona es dinámica, controlado por el parámetro provide-ixfr.

Servidores primarios y secundarios (master/slave):

- El término "maestro" reemplazó a "primario" y "esclavo" a "secundario" con BIND 8.x.
- Un esclavo siempre inicia la operación de transferencia de zona (AXFR o IXFR) usando TCP en el puerto 53.
- El maestro transferirá el archivo de zona solicitado al esclavo.

Política de notificación (NOTIFY):

- El RFC 1912 recomendaba un intervalo de ACTUALIZACIÓN de hasta 12 horas en el intervalo de ACTUALIZACIÓN de un registro de recursos SOA. Esto significaba que, en el peor de los casos, los cambios en el servidor de nombres maestro podían no ser visibles en los servidores de nombres esclavos durante hasta 12 horas, lo cual puede ser inaceptable en un entorno dinámico.
- El RFC 1996 introdujo un esquema mediante el cual el Maestro enviará un mensaje de NOTIFICACIÓN (NOTIFY) a los Servidores de Nombres Esclavos.
- Este mensaje indica que "pudo" haberse producido un cambio en los registros de dominio.
- Los Esclavos, al recibir la NOTIFICACIÓN, solicitarán el último Registro de Recursos SOA y, si el número de serie del RR SOA es mayor que su valor actual, iniciarán una transferencia de zona mediante una Transferencia de Zona Completa (AXFR) o una

Transferencia de Zona Incremental (IXFR).

- El comportamiento de NOTIFY en BIND está controlado por los parámetros `notify`, `also-notify` y `notify-source`.
  - `notify`: Es aplicable tanto a zonas maestras (tipo maestro;) como a zonas esclavas (tipo esclavo;). Si se configura en 'sí' (predeterminado), cuando se carga o cambia una zona (por ejemplo, después de una transferencia de zona), los mensajes NOTIFY se envían a los servidores de nombres definidos en los registros NS para la zona (excepto a sí misma y al servidor de nombres 'Maestro Primario' definido en el registro SOA) y a cualquier IP incluida en cualquier declaración `also-notify`. Si se establece en 'no', los mensajes NOTIFY no se envían. Si se establece en 'explícito', NOTIFY solo se envía a aquellas IPs enumeradas en una declaración NOTIFY explícita.
  - `also-notify`: Se utiliza solo en las zonas maestras. Permite notificar a direcciones IP adicionales además de las definidas en los registros NS.
  - `notify-source`: Define la dirección IPv4 (y opcionalmente el puerto) que se utilizará para las operaciones NOTIFY salientes.
  - Un mensaje NOTIFY no indica necesariamente que el archivo de zona haya cambiado; por ejemplo, si el maestro o la zona se recarga, se activa un mensaje NOTIFY aunque no haya habido cambios.

Proceso de Transferencia de Zona (Detallado):

1. Si el Maestro ha sido configurado para admitir mensajes "NOTIFY", enviará un mensaje "NOTIFY" a cada esclavo configurado cuando el archivo de zona del Maestro cambie de estado. Un mensaje "NOTIFY" no necesariamente indica que el archivo de zona ha cambiado; por ejemplo, si se recarga el maestro o la zona se activa también se enviará un mensaje "NOTIFY".
2. Independientemente de si el Maestro ha sido configurado para enviar mensajes "NOTIFY" o no, el Esclavo siempre utilizará el proceso pasivo o de "sondeo". Cuando se carga un servidor Esclavo, leerá cualquier archivo de zona guardado actual (ver declaración de archivo) o iniciará inmediatamente una transferencia de zona si no hay ningún archivo de zona guardado. Luego inicia un temporizador utilizando el valor de actualización en SOA de la zona.
3. Si el temporizador de actualización del Esclavo expira o recibe un mensaje "NOTIFY", el Esclavo emitirá inmediatamente una consulta para conocer el SOA del Maestro de zona.
4. Cuando llega la respuesta del SOA del Maestro, el Esclavo compara su número de serie SOA actual con el de la respuesta.
5. Si el número de serie del Maestro es mayor, el Esclavo inicia una transferencia de zona.
   1. Si el esclavo no puede leer el SOA del maestro (o no puede iniciar la transferencia de zona), intentará nuevamente después del tiempo de reintento definido en el SOA de la zona. El procedimiento de reintento se repetirá hasta que tenga éxito.
   2. El esclavo siempre inicia la operación de transferencia de zona (AXFR o IXFR) usando TCP en el puerto 53. Esto se puede configurar usando la instrucción transfer-source.
6. El Maestro transferirá el archivo de zona solicitado al Esclavo.
7. Al finalizar, el Esclavo restablecerá sus temporizadores de actualización y caducidad.

Seguridad en transferencias (ACL, TSIG):

ACL y TSIG se utilizan tanto para proporcionar seguridad en las transferencias de zona como para el caso de actualizaciones dinámicas que veremos más adelante:

- Para las Transferencias de zona, BIND proporciona Listas de Control de Acceso (ACLs) que permiten una protección simple basada en direcciones IP. Aunque las ACLs basadas en IP son relativamente fáciles de subvertir mediante la suplantación de IP, son significativamente mejores que no hacer nada y requieren muy poco trabajo. Es posible operar con múltiples maestros (sin esclavos) y eliminar por completo la amenaza de transferencia, aunque las actualizaciones de archivos de zona deberán sincronizarse manualmente.
- TSIG/TKEY: Las especificaciones DNS (RFC 2845 - TSIG y RFC 2930 - TKEY) proporcionan mejoras al protocolo de autenticación para asegurar estas transacciones de Servidor a Servidor.
- Las implementaciones de TSIG y TKEY son desordenadas pero no demasiado complicadas, debido al alcance limitado del problema. Con las transacciones de Servidor a Servidor, hay un número finito y normalmente pequeño de hosts involucrados. Los protocolos dependen de un secreto compartido entre el maestro y los esclavos o actualizadores. Se asume que el secreto compartido se puede transmitir de forma segura al servidor par por medios no cubiertos en el propio protocolo (típicamente cadenas aleatorias largas de caracteres base64), como el teléfono, correo, fax o correo electrónico PGP.
- El secreto compartido está expuesto a ataques de fuerza bruta, por lo que el cambio frecuente de secretos compartidos (mensual o más) puede convertirse en una realidad.
- TKEY permite la automatización del intercambio de claves utilizando un algoritmo Diffie-Hellman, pero comienza con un secreto compartido. TKEY parece tener un uso muy limitado.

Ejemplo de configuración de una zona esclava y su correspondiente maestra:

```
// Configuración de zona esclava para example.com

zone "example.com" {
    // Especifica el tipo de zona como "slave" (esclava)
    type slave;

    // Lista de direcciones IP de los servidores maestros.
    // En este caso, solo el 10.0.0.1
    masters {
```

```
        10.0.0.1;
    };

    // La ruta y nombre del archivo donde BIND almacenará
    // una copia de la zona transferida. que no tiene por qué coincidir con lo real en vuestro servidor
    file "slave/db.example.com";
};
```

Práctica 7:

Requisitos:

- Haber hecho la práctica 5.
- **Durante todo el proceso de la práctica deberán hacerse capturas de imagen de cada uno de los pasos dados así como capturas y almacenaje de los paquetes de wireshark. Algunos de ellos serán solicitados por el profesor una vez concluida la práctica.**

Antes de empezar:

- Recuerda que tienes a tu disposición la documentación de Zytrax donde puedes consultar sobre conceptos y sintaxis.
- Sustituye el valor 'XX' por tu número de puesto asignado que venimos utilizando
- Toma nota de todo lo que vayas haciendo, incluyendo errores y soluciones sobre los mismos, así como capturas
- Debes utilizar el cliente LinuxXX para hacer todas las consultas DNS de prueba.
- Recuerda que cada vez que actualices un fichero de zona tienes que incrementar el número de serie del SOA.
- Recuerda que dispones de 'rndc' para poder interactuar con los servidores DNS

Prueba:

- Hay que utilizar el comando 'dig' para todas la pruebas
- Dispones además de comandos de comprobación de sintaxis para el BIND9:
  - Podemos comprobar un archivo de configuración con, por ejemplo: **named-checkconf /path/to/named.conf**
  - Podemos comprobar una zona con, por ejemplo: **named-checkzone example.net /etc/bind/example.net**
  - **Dispones de muchos otros comandos que te pueden resultar útiles en el apartado 15 del manual de BIND9, incluyendo la opción de interaccionar con los servidores DNS utilizando 'rndc'**

Pasos:

1. Crea una nueva zona maestra de resolución directa en 'BOOKWORMXXA' denominada "otroasirXX.asir" con las siguientes características y haz las correspondientes pruebas que verifiquen su correcto funcionamiento.
   1. SOA: bookwormXXa.otroasirXX.asir y parámetros de tiempo como en prácticas anteriores, salvo los indicados en las siguientes líneas
   2. NS: bookwormXXa.otroasirXX.asir.
   3. A: 10.0.128+XX.1 es la IP de bookwormXXa.otroasirXX.asir.
   4. CNAME: www es alias de bookwormXXa.otroasirXX.asir.
   5. Servidor2 tiene como IP 10.0.128+XX.254
   6. El tiempo de refresco de la zona será de 180 segundos y el de reintento será de 30 segundos
2. Pon a capturar el tráfico en uno de los servidores
3. Crea una nueva zona "esclava" de resolución directa en BOOKWORMXXB denominada "otroasirXX.asir" (la zona del servidor maestro y del esclavo tienen que llamarse igual. El servidor maestro será 'bookwormXXa'. Se indicará con la correspondiente IP. Al recargar la configuración debería producirse una solicitud de SOA y la correspondiente transferencia de zona. Observa el tráfico que se produce y pregunta al servidor B por 'servidor2.otroasirXX.asir' para constatar que funciona.
4. Observa como cada cierto tiempo se produce la consulta del SOA por parte del esclavo pero no hay transferencia de zona
5. Modifica la zona maestra cambiando la IP 10.0.128+XX.254 por la 253, cambia el número de serie y recarga la zona. Observa si al recargar la zona se produce una notificación inmediata, una solicitud de SOA y la transferencia IXFR que permita actualizar la IP que ha cambiado.

En este último capítulo trataremos otros aspectos importantes del servicio DNS

Planteamiento de la situación

En ocasiones necesitamos tener una DNS público pero la dirección IP del mismo cambia porque nuestro proveedor de servicios de Intenernet no nos concede una IP fija. ¿Cómo resolvemos este problema? Utilizando un servicio de **DNS Dinámico (DDNS)**.

¿Cómo funciona el servicio y qué pasos hay que seguir?

1. Elegir un Proveedor de DNS Dinámico (DDNS)

Necesitas un servicio externo que se encargue de actualizar los registros DNS cuando tu IP cambie. Hay muchos proveedores, algunos gratuitos y otros de pago, como:

- **No-IP**
- **DynDNS (Oracle/Dyn)**
- **DuckDNS**
- **Cloudflare** (a través de su API)
- **Servicios ofrecidos por tu registrador de dominios** (como OVHcloud, IONOS, etc.)

Deberás registrarte y crear un **nombre de *host*** (por ejemplo, `miservidor.nombredd.com`).

2. Configurar el Cliente DDNS

El DDNS funciona mediante un *cliente* que se ejecuta en tu red y notifica al proveedor DDNS cada vez que detecta un cambio en tu dirección IP pública. Este cliente puede ser:

- El Rúter (Opción más común y recomendada): Muchos rúteres modernos tienen una sección de configuración de DDNS.
- Software Cliente en el Servidor

3. Configurar el Servidor y el Rúter para el acceso

Para que el tráfico llegue al servidor específico dentro de tu red local, necesitas configurar el reenvío de puertos (*Port Forwarding*)

¿Cuál es la base de funcionamiento?

La base de funcionamiento de las **actualizaciones dinámicas de los servidores DNS** (a menudo llamadas **DDNS** o **DNS dinámico**) se centra en el protocolo definido por el **RFC 2136: *Dynamic Updates in the Domain Name System (DNS UPDATE)***.

¿Qué permite?

Este mecanismo permite que los registros de recursos (RR) de una zona DNS se modifiquen automáticamente sin intervención manual, lo que es esencial en entornos donde las direcciones IP cambian con frecuencia.

Protocolo de Funcionamiento (RFC 2136)

Protocolo de Funcionamiento (RFC 2136):

- La actualización dinámica se produce mediante un **mensaje de actualización** especial que un cliente envía al servidor DNS. Este mensaje no es una consulta DNS estándar, sino una solicitud de modificación de la base de datos de la zona.
- Componentes Clave:

Componente	Descripción
Cliente de Actualización	La entidad que necesita modificar su registro DNS (puede ser un *host*, un *router* o un servidor DHCP).
Servidor DNS	El servidor autorizado para la zona que contiene los registros a modificar (el servidor primario o maestro).
Mensaje de Actualización	El mensaje enviado por el cliente que incluye tres secciones principales: **Prerrequisitos**, **Instrucciones de Actualización** y **Sección de Datos**.

- Proceso Paso a Paso:
  1. **Detección de Cambio:** El cliente (por ejemplo, un equipo que ha obtenido una nueva IP a través de DHCP, o un *router* cuya IP pública ha cambiado) detecta que su dirección IP ya no coincide con la dirección IP actual registrada en el DNS.
  2. **Preparación del Mensaje de Actualización:** El cliente genera un mensaje de actualización que contiene:
     1. **Prerrequisitos:** Condiciones opcionales que deben ser ciertas para que la actualización se realice (p. ej., "El registro A para 'https://www.google.com/search?q=host.ejemplo.com' debe existir" o "El registro A para 'https://www.google.com/search?q=host.ejemplo.com' no debe existir").
     2. **Instrucciones de Actualización:** Las modificaciones reales que se solicitan, como:
        - **Eliminar** el registro antiguo (dirección IP anterior).
        - **Añadir** el nuevo registro (dirección IP actual).
  3. **Envío al Servidor:** El cliente envía este mensaje al servidor DNS autorizado para la zona.
  4. **Procesamiento y Validación (en el Servidor):** El servidor DNS:
     1. **Valida la Seguridad:** Verifica la identidad del cliente (mediante mecanismos de seguridad como **TSIG** o, en entornos de Windows, mediante **Actualizaciones Dinámicas Seguras** integradas con Active Directory).

2. **Comprueba los Prerrequisitos:** Si se especificaron prerrequisitos, se asegura de que se cumplen. Si fallan, la actualización se rechaza.
3. **Ejecuta la Actualización:** Si la validación es exitosa, el servidor aplica los cambios a la base de datos de la zona, modificando el registro (y a menudo el registro correspondiente) para apuntar a la nueva IP.

5. **Respuesta:** El servidor responde al cliente con una confirmación de éxito o un mensaje de error.
6. **Propagación:** Si el servidor es el primario, debe actualizar el número de serie e informar a los servidores secundarios para que sincronicen la zona modificada.

Escenarios Comunes de Implementación

El mecanismo de actualización del RFC 2136 se utiliza de dos formas principales en la práctica:

1. Cliente de Host Directo

El propio dispositivo (servidor, PC, ruter) es el que ejecuta un *cliente DDNS* que realiza la actualización:

- El *host* monitoriza su propia dirección IP.
- Si la IP cambia, el *host* envía la solicitud de actualización directamente al servidor DNS del proveedor o al servidor DNS local.

1. Actualizaciones Asistidas por DHCP (Escenario Común en Redes Corporativas)

En redes que utilizan un servidor DHCP para asignar direcciones IP, el proceso puede dividirse:

1. **Asignación de IP:** Un cliente solicita una IP al servidor DHCP.
2. **Notificación al DHCP:** El cliente proporciona su nombre de *host* al servidor DHCP.
3. **Actualización a Cargo del DHCP:** El servidor DHCP, al conocer el nombre del *host* y la IP asignada, actúa como el cliente de actualización y envía la solicitud DDNS al servidor DNS. Esto libera al *host* de la tarea de actualización y centraliza la gestión. (El DHCP puede actualizar el registro de la IP a nombre, y el de nombre a IP, o delegar parte de esta tarea al *host* original, dependiendo de la configuración).

Objetivo:

El objetivo principal de DoT es mejorar la **privacidad** y la **seguridad** de las consultas DNS, que tradicionalmente se transmiten sin cifrar (en texto plano).

Base de Funcionamiento de DNS sobre TLS (DoT)

El DNS tradicional utiliza el protocolo UDP (o TCP) sobre el puerto 53 para enviar las consultas y respuestas. DoT, por otro lado, **añade una capa de cifrado** al proceso:

1. **Conexión Inicial (Handshake TLS):**
   - El cliente establece una conexión con el servidor DNS a través de un puerto dedicado para DoT, que suele ser el **puerto TCP 853**.
   - Una vez establecida la conexión TCP, se inicia el protocolo de enlace (**TLS Handshake**), donde el cliente y el servidor negocian qué algoritmos de cifrado usarán y el servidor se autentica ante el cliente usando un certificado digital (similar a cómo funciona HTTPS).
2. **Cifrado de Consultas:**
   - Una vez que el túnel TLS seguro está establecido, todas las consultas y respuestas DNS subsiguientes se envían **cifradas** dentro de esta conexión segura.
   - Este cifrado impide que terceros puedan leer qué sitios web o servicios estás consultando, ni manipular las respuestas DNS.
3. **Autenticación:**
   - El uso del certificado TLS permite que el cliente **autentique** la identidad del servidor DNS, asegurándose de que se está comunicando con el servidor legítimo y no con un atacante que intenta suplantarlo.

Ventajas de DNS sobre TLS

Aspecto	Ventaja	Explicación
Privacidad	**Evita el espionaje**	Impide que el ISP, los operadores de red Wi-Fi pública o entidades gubernamentales puedan ver las consultas DNS de un usuario, protegiendo su historial de navegación y sus actividades en línea.

Aspecto	Ventaja	Explicación
Seguridad	**Previene la manipulación (*Spoofing*)**	El cifrado y la autenticación del servidor impiden ataques como el envenenamiento de caché DNS o la falsificación de respuestas DNS, donde un atacante dirige al usuario a un sitio malicioso.
Integridad	**Garantiza la autenticidad**	El protocolo TLS asegura que la respuesta DNS recibida proviene del servidor legítimo y no ha sido alterada en el camino.
Censura	**Dificulta el bloqueo**	Al ocultar la consulta DNS, se hace más difícil para ciertas autoridades de red realizar bloqueos o filtrados selectivos basados únicamente en el nombre de dominio.

Inconvenientes de DNS sobre TLS

Aspecto	Inconveniente	Explicación
Latencia	**Introduce sobrecarga (*Overhead*)**	La conexión inicial (el **TLS Handshake**) es un proceso que consume recursos y tiempo. Esto puede añadir una pequeña latencia a la primera consulta, aunque las consultas posteriores dentro de la misma sesión suelen ser rápidas.
Inspección de Red	**Dificulta el filtrado/monitoreo**	Los administradores de red (por ejemplo, en empresas o escuelas) pierden la capacidad de inspeccionar el tráfico DNS para aplicar filtros de seguridad, control parental o políticas de uso aceptable, ya que el contenido está cifrado.
Bloqueo	**Facilidad de bloqueo del puerto**	DoT utiliza un puerto específico (**TCP 853**). Si el administrador de una red desea prohibir el uso de DoT, le basta con bloquear el tráfico en ese puerto en su *firewall*. (A diferencia de DNS sobre HTTPS o DoH, que usa el puerto 443, más difícil de bloquear).
Dependencia del Servidor	**Concentración de datos**	La privacidad se transfiere del ISP al operador del servidor DNS (como Cloudflare, Google, etc.). Si confías tu privacidad a este servidor, esencialmente estás confiando en que no registrarán o usarán tus consultas de manera indebida.

Objetivo

El objetivo principal de DoH es **aumentar la privacidad y seguridad** de las consultas DNS.

- **Privacidad:** Evitar que terceros (como proveedores de servicios de internet - ISP, operadores de red o atacantes) puedan interceptar, leer o monitorear las solicitudes de resolución de nombres de dominio de un usuario, lo que podría revelar su historial de navegación y sus intereses.
- **Seguridad:** Prevenir ataques como el *DNS spoofing* (suplantación de DNS) o el secuestro de DNS (*DNS hijacking*), que buscan modificar las respuestas DNS para dirigir a los usuarios a sitios maliciosos.

Base de Funcionamiento

DoH encapsula las consultas y respuestas DNS dentro del protocolo **HTTPS (HTTP seguro)**.

1. **Cifrado (Encriptación):** En lugar de enviar las consultas DNS en texto claro a través del puerto 53 UDP/TCP (el método tradicional y no cifrado), DoH utiliza el protocolo **TLS/SSL** (el mismo que usa HTTPS) para cifrar la comunicación.
2. **Puerto 443:** Las consultas se envían al servidor DNS compatible con DoH a través del **puerto 443 TCP**, el mismo puerto que se utiliza para el tráfico web normal cifrado (HTTPS).
3. **Encapsulación:** La consulta DNS se convierte en una solicitud HTTP, se cifra y se envía al servidor DoH.
4. **Respuesta:** El servidor resuelve la consulta, cifra la respuesta y la envía de vuelta al cliente.
5. **Ocultación de Tráfico:** Al utilizar el puerto 443, el tráfico DNS se mezcla y camufla con el tráfico web HTTPS general, lo que dificulta a los observadores de red distinguirlo y filtrarlo o bloquearlo específicamente.

Ventajas

- **Mayor Privacidad:** Nadie en la ruta de red (incluido el ISP o el administrador de la red local) puede ver los dominios que un usuario está consultando, protegiendo su historial de navegación.
- **Mayor Seguridad:** El cifrado previene la manipulación de las respuestas DNS y protege contra ataques de *spoofing* o *hijacking*.
- **Evasión de Censura/Filtros:** Puede ayudar a los usuarios a eludir la censura o el bloqueo de sitios web basado en DNS que implementan algunos gobiernos o proveedores de red.

- **Mejora de la Accesibilidad:** Al utilizar el puerto 443, que rara vez se bloquea, DoH puede funcionar en redes Wi-Fi públicas o corporativas que restringen el tráfico DNS tradicional (puerto 53).

Inconvenientes

- **Centralización del Tráfico DNS:** La adopción de DoH puede concentrar una gran parte del tráfico DNS en un número reducido de grandes proveedores de servicios DoH (como Google, Cloudflare, etc.), lo que plantea preocupaciones sobre la privacidad (el proveedor DoH ve todas las consultas) y crea un posible punto único de fallo o de control.
- **Dificultad de Monitoreo en Redes Corporativas:** Al cifrar el tráfico DNS y ocultarlo en el puerto 443, DoH puede **eludir los sistemas de seguridad y filtrado de red** que dependen de la inspección del tráfico DNS tradicional para bloquear malware, filtrar contenido o aplicar políticas.
- **Mayor Latencia Potencial:** Aunque a menudo es rápido, el establecimiento de la conexión HTTPS/TLS inicial puede añadir una ligera sobrecarga o latencia en comparación con el DNS tradicional por UDP, que es más simple y sin estado.
- **Conflicto con la Configuración Local:** DoH a nivel de aplicación (navegador) ignora la configuración DNS del sistema operativo o del router, lo que puede anular las medidas de seguridad o control parental implementadas por el administrador de la red.

¿Qué es rndc?

- rndc es una utilidad de línea de comandos para administrar el servidor de nombres BIND.
- Permite controlar el demonio **named** a través de comandos autenticados.
- Utiliza un archivo de configuración (rndc.conf) y una clave secreta compartida para la autenticación
- Puede ejecutar tareas como recargar zonas, comprobar el estado del servidor y detenerlo.

Propósito de rndc

- Administración remota: Permite gestionar el servidor de nombres **named** desde el host local o desde otro host a través de una conexión TCP.
- Autenticación: Utiliza un método de autenticación basado en llaves secretas compartidas para prevenir el acceso no autorizado.
- Control del servidor: Permite ejecutar comandos para interactuar con el servidor de nombres, como recargar configuraciones de zonas, consultar el estado o reiniciar el servicio.

¿Cómo funciona?

- Archivo de configuración (rndc.conf): rndc lee este archivo para obtener la información necesaria sobre cómo contactar al servidor de nombres, incluyendo la dirección, el puerto, el algoritmo de autenticación y la clave secreta compartida.
- Autenticación: Los comandos se envían a través de una conexión TCP y se autentican utilizando firmas digitales basadas en la clave secreta compartida que debe coincidir en el cliente (rndc) y el servidor (named).
- Comandos: Si se ejecuta rndc sin argumentos, muestra un resumen de los comandos y opciones disponibles.

Uso común

- Reiniciar el servidor: `rndc reload` o `rndc restart` para reiniciar el servidor named.
- Consultar estado: `rndc status` para obtener el estado del servidor de nombres.
- Cargar zonas: `rndc reload zone_name` para recargar una zona específica.