Laboratorio DNS Interno con Docker: Análisis Técnico Profundo

Propósito Real

Simular un servidor DNS interno autoritativo para el dominio `empresa.local`, utilizando Docker y BIND9. Esta configuración permite estudiar cómo se resuelven nombres locales sin depender de DNS públicos, emulando un entorno de red empresarial real.

El dominio `intranet.empresa.local` será un subdominio ficticio que apunta a una IP interna, simulando una intranet.

Estructura de Archivos y Función Técnica

`docker-compose.yml`

Rol: Orquesta el entorno Docker que ejecuta BIND9 como servidor DNS.

services:

bind:

image: internetsystemsconsortium/bind9:9.18

Función técnica:

- Usa la imagen oficial del **servidor BIND9** del ISC.
- Expone **el puerto 53 TCP y UDP**, requeridos para DNS.
- Monta tres archivos esenciales como volúmenes:
    - `named.conf.options` (opciones globales)
    - `named.conf.local` (zonas personalizadas)
    - `db.empresa.local` (base de datos de la zona `empresa.local`)
- La línea `cap_add: NET_ADMIN` permite al contenedor gestionar interfaces de red si es necesario (no siempre obligatorio, pero recomendable para DNS).

`named.conf.options`

Rol: Define el **comportamiento general del servidor DNS**.

```
options {
  directory "/var/cache/bind";
  forwarders { 8.8.8.8; };
  allow-query { any; };
  listen-on { any; };
};
```

Función técnica:

- `directory`: indica dónde buscar archivos de zona.
- `forwarders`: si el DNS no tiene una respuesta, reenviará la consulta a `8.8.8.8` (Google DNS). Esto permite que el servidor también funcione como **caché y proxy** DNS.
- `allow-query { any; };`: permite que cualquier cliente consulte al servidor (ideal en laboratorio, en producción se recomienda restringir).
- `listen-on { any; };`: escucha en todas las interfaces. Permite aceptar peticiones tanto desde dentro del contenedor como desde el host.

`named.conf.local`

Rol: Declara la **zona DNS personalizada** y enlaza con su archivo de datos.

```
zone "empresa.local" {
    type master;
    file "/var/cache/bind/db.empresa.local";
};
```

Función técnica:

- Informa a BIND que este servidor es **"maestro" o autoritativo** para el dominio `empresa.local`.
- Indica que las respuestas a ese dominio deben extraerse del archivo `db.empresa.local`.
- Este archivo es indispensable: sin él, el dominio no será reconocido ni respondido.

`db.empresa.local`

Rol: Es el **archivo de zona**. Contiene los registros DNS del dominio `empresa.local`.

```
$TTL 604800
@ IN SOA ns1.empresa.local. admin.empresa.local. (
      2 ; Serial
 604800 ; Refresh
  86400 ; Retry
2419200 ; Expire
 604800 ) ; Negative Cache TTL
@    IN NS ns1.empresa.local.
ns1  IN A  172.20.0.53
intranet IN A 172.20.0.100
```

Función técnica (clave por línea):

- `$TTL`: Tiempo de vida por defecto para los registros en segundos.
- `SOA (Start of Authority)`: Define al **servidor primario** (`ns1.empresa.local.`) y un correo administrativo (`admin.empresa.local.`).
- `NS`: Declara el **servidor de nombres** para la zona.

- `A` (Address):
    - `ns1`: Dirección IP del servidor DNS (simulado aquí como `172.20.0.53`).
    - `intranet`: Dirección IP del servidor web local simulado (`172.20.0.100`), accesible como `intranet.empresa.local`.

Este archivo es la **base de datos DNS real**. Define qué respuestas entregará tu servidor al resolver subdominios del dominio.

Flujo de Resolución Simulado

1. Tu sistema solicita `intranet.empresa.local`.
2. Consulta al servidor DNS (en Docker).
3. El contenedor analiza la zona `empresa.local`.
4. Encuentra `intranet` y responde `172.20.0.100`.

Este flujo reemplaza completamente el uso de servidores públicos como `8.8.8.8` para esa zona específica.

Lecciones Avanzadas Derivadas

- **Autoritativo vs. Reenviador:** BIND puede ser ambas cosas. Aquí es autoritativo para `empresa.local` y reenviador para todo lo demás.
- **Seguridad:** `allow-query` debe limitarse por red interna. DNS mal expuestos son vectores comunes de exfiltración o ataques de amplificación.
- **Tiempos DNS (TTL)**: afectan la propagación y la caché. TTL bajo en pruebas, alto en producción.
- **Zonas y Subzonas:** puedes crear múltiples zonas como `dev.empresa.local`, `vpn.empresa.local` en archivos similares.

Blandskron

El archivo `hosts` en Windows: Función y Rol en la Simulación

¿Qué es?

El archivo `hosts` es una **tabla de resolución local** que permite forzar manualmente la asociación entre un **nombre de dominio** y una **dirección IP**, antes de que el sistema consulte un servidor DNS.

Ubicación:

C:\Windows\System32\drivers\etc\hosts

Su propósito es **interceptar peticiones de nombre antes de que lleguen al servidor DNS**. Si el sistema encuentra el nombre ahí, lo resuelve directamente, **omitiendo completamente cualquier servidor DNS configurado**.

¿Por qué se usa en este laboratorio?

Porque:

- Permite probar la resolución del dominio `intranet.empresa.local` incluso si el DNS en Docker aún no está operativo.
- Sirve como contingencia si no quieres modificar la configuración de red global.
- Es útil para pruebas locales, hacking ético, desarrollo web interno y bypass temporal de DNS.

Cómo modificar `hosts` correctamente en Windows

Requisitos

El archivo `hosts` es **protegido por el sistema**, por lo que debes editarlo **como administrador**.

Procedimiento recomendado

1. Abre Visual Studio Code como administrador:

- Busca `Visual Studio Code` en el menú de inicio.
- Clic derecho → **"Ejecutar como administrador"**.

2. Abre el archivo desde VSCode:

- `Archivo > Abrir archivo...`
- Navega a: `C:\Windows\System32\drivers\etc`
- Cambia el filtro abajo de `*.txt` a `Todos los archivos (*.*)`
- Selecciona `hosts` y ábrelo

3. Añade la siguiente línea al final del archivo:

127.0.0.1 intranet.empresa.local

O si estás apuntando a un servicio en contenedor o red interna:

172.20.0.100 intranet.empresa.local

4. Guarda (`Ctrl + S`) y cierra VSCode.

Verificación de funcionamiento

Desde terminal (Git Bash, CMD o PowerShell):

ping intranet.empresa.local

Debe responder con:

Respuesta desde 127.0.0.1 ...

Y también:

curl http://intranet.empresa.local

Debe devolver contenido HTML si hay un servidor web local escuchando en esa IP/puerto.

Errores comunes

- Editar sin permisos: cambios no se guardan.
- No cambiar el filtro a “Todos los archivos” al abrir: no aparece el archivo.
- No hay salto de línea al final: puede no interpretarse correctamente.
- IP incorrecta o servicio no levantado en esa IP: da error de conexión o timeout.

Valor técnico agregado

Editar el archivo `hosts` no solo simula un DNS local. También:

- Permite **realizar redireccionamientos controlados** (útil en pentesting o MITM).
- Ayuda a **bloquear dominios maliciosos** (redirigiéndolos a `127.0.0.1`).
- Es una técnica base en el desarrollo web local y en entornos air-gapped o cerrados.