

Análisis del Flujo de Datos y Protocolos en una Comunicación Web Segura

OBJETIVO

Desplegar un entorno de red local que simule el acceso a un sitio web interno con dominio personalizado (<https://intranet.miempresa.local>), y analizar detalladamente la participación de los protocolos del modelo TCP/IP mediante herramientas de inspección como Wireshark y DevTools.

ANÁLISIS POR CAPAS – MODELO TCP/IP

Capa TCP/IP	Protocolo(s)	Función Principal
Aplicación	HTTPS, DNS	Establece la interfaz con el usuario. HTTPS cifra la comunicación; DNS resuelve la IP
Transporte	TCP	Garantiza entrega confiable, ordenada y libre de errores de los segmentos
Internet	IP (IPv4/IPv6)	Direcciona los paquetes hacia su destino usando direcciones IP
Enlace de Datos	Ethernet, MAC	Maneja la entrega física en red local mediante tramas; usa direcciones MAC

INSPECCIÓN DE TRÁFICO – FLUJO REAL

Herramienta:

DevTools (F12 → Red) en Chrome o Wireshark con filtros específicos.

Simulación:

Navegación hacia <https://intranet.miempresa.local>

Paquetes Observados:

1. DNS Request

- Dominio: `intranet.miempresa.local`
- Acción: Resolver a IP → 127.0.0.1 (simulado en hosts)

2. TCP SYN / ACK

- Puerto destino: 443
- Establecimiento de la conexión confiable

3. TLS Handshake

- Inicio de sesión segura
- Intercambio de claves (client hello, server hello)

4. HTTP/HTTPS GET Request

- Solicitud del recurso: `index.html`
- Cifrado dentro de túnel TLS

5. HTTP/HTTPS 200 OK

- Entrega de contenido cifrado
- Inclusión de HTML en respuesta

RESPUESTAS A PREGUNTAS TÉCNICAS

Pregunta	Respuesta
¿Qué protocolo tradujo el nombre de dominio en una IP?	DNS
¿Qué protocolo garantizó que los datos llegaron correctamente?	TCP
¿Qué protocolo cifró la conexión para que fuera segura?	TLS (HTTPS)
¿Qué protocolo se encargó del direccionamiento de red?	IP

IMPLEMENTACIÓN LOCAL DEL SERVICIO CON DOMINIO SIMULADO

Estructura de Proyecto

mi-proyecto/

├── docker-compose.yml

├── Dockerfile

├── nginx.conf

├── app/

| └── index.html

docker-compose.yml

version: '3.8'

services:

nginx:

image: nginx

ports:

- "80:80"

volumes:

- ./nginx.conf:/etc/nginx/conf.d/default.conf
- ./app:/usr/share/nginx/html:ro

networks:

- mi_red

app:

build: .

networks:

- mi_red

networks:

mi_red:

driver: bridge



nginx.conf

server {

listen 80;

server_name intranet.miempresa.local;

root /usr/share/nginx/html;

index index.html;


```
location / {  
    try_files $uri $uri/ =404;  
}  
}
```

Dockerfile

```
FROM nginx:alpine
```

app/index.html

```
<!DOCTYPE html>  
<html lang="es">  
<head>  
    <meta charset="UTF-8">  
    <title>Intranet de Mi Empresa</title>  
</head>  
<body>  
    <h1>Bienvenido a la intranet de miempresa.local</h1>  
</body>  
</html>
```



CONFIGURACIÓN DE DOMINIO LOCAL

Modificación de **hosts** (Windows)

```
127.0.0.1 intranet.miempresa.local
```

Instrucciones:

1. Ejecutar Bloc de notas como **administrador**
2. Ir a `C:\Windows\System32\drivers\etc\hosts`
3. Añadir la línea y guardar

PUESTA EN MARCHA

`docker-compose up --build`

Verificación

- Abrir en navegador: `http://intranet.miempresa.local`

Comprobar resolución:

`ping intranet.miempresa.local`

-