Tema 5: Servicios de red implicados en el despliegue de una aplicación web

Resolutores de nombres

Internet y por extensión la gran mayoría de las redes de ordenadores en la actualidad utilizan el conjunto de protocolos TCP/IP. La capa de red de este conjunto es la que define las direcciones de los diferentes dispositivos conectados, conocidas como direcciones IP. En la versión actual (4) del protocolo IP las direcciones se componen de cuatro grupos o números de 8 bits y suelen representarse como cuatro puntos del 0 al 255 separados por puntos. No es el propósito de este módulo abundar en este tema, baste decir que lo que nosotros conocemos como "direcciones de Internet" o URLs no son más que asociaciones de un nombre más fácil de recordar o que se asocie a algo con una de estas direcciones IP.

El protocolo de este conjunto que se encarga de esta tarea es DNS (*Domain Name System*). El proceso de buscar la dirección IP asociada a un nombre de dominio determinado (por ejemplo www.nba.com) se conoce como resolver una dirección.

Como todo lo que venimos viendo en este curso sigue un modelo Cliente/Servidor en el que los clientes (**resolutores** o *resolvers*) forman parte de los sistemas operativos de las máquinas y se encargan de contactar con los servidores para que les indiquen la dirección IP asociada a una URL cuando sea necesario. Hay que tener en cuenta que un servidor DNS puede ser cliente de otro servidor DNS.

Los servidores se conocen como Servidores de Nombres (*Name Servers*) y funcionan según el modelo jerárquico y distribuido:

- Jerárquico porque se organiza como un árbol invertido. En la raíz (dominio arpa que recibe su nombre del proyecto germen de Internet) estaría la autoridad principal que delega las terminaciones principales a organismos, entidades o países. Estas terminaciones incluyen elementos como .com, .org, .net, .gov, .es, .us, .uk, .fr etc. Los encargados de gestionar cada una de estas terminaciones (dominios de nivel superior o top-level) pueden a su vez subdividir y delegar el dominio. Por ejemplo en Gran Bretaña han creado uno comercial y su terminación es .co.uk y así .co sería una zona delegada dentro de .uk
  Un ejemplo completo de todo esto sería por ejemplo www.ii.uam.es en el que (de derecha a izquierda) se indica que es un dominio de España, de la Universidad Autónoma de Madrid, de la Escuela de Ingeniería Informática y dentro de eso que hace referencia al servidor web. En este último apartado puede haber otros como por ejemplo ftp. Las dos direcciones de ejemplo vistas arriba son *hostnames* ya

que tienen una o más direcciones IP asociadas. Las partes de la derecha indican dominios o subdominios, pero no se asocian con direcciones concretas.

- Distribuido porque no es necesario que un resolutor DNS sea capaz de conocer todas las posibles asociaciones sino que puede conocer algunos y pasarle las que no conoce a otros servidores así como recibir a su vez peticiones de otros pares para resolver las que sí conoce. Esto lo veremos en más detalle en el proceso de resolución de nombres.

Como nota para el futuro, conviene decir que nos encontramos en un proceso de transición tanto en IP de la v4 a la v6 como en DNS en el que se pretende flexibilizar el conjunto de terminaciones y se están estudiando posibles consideraciones para incluir nuevos dominios de nivel superior.

Proceso de resolución de un nombre de dominio

Vamos a ver el ejemplo con www.ii.uam.es

Un resolutor de nombres realiza su función mediante consultas a diferentes servidores. Siempre empieza por el nivel superior (el que está más a la derecha en la URL).

1. El **resolutor del cliente** le envía una petición a un servidor de nombres de la zona raíz (*root zone*) preguntando por ese subdominio de nivel superior (es). La dirección IP de al menos algunos servidores de la zona raíz está disponible mediante unos archivos que actualiza periódicamente una autoridad. Realmente el ordenador de nuestra casa necesita las direcciones IP de servidores DNS de un ISP (u otros organismos) como veremos más abajo. El servidor raíz le contesta con la dirección IP de quién gestiona el dominio de nivel superior .es (o con la información más completa que tenga que generalmente es a quién preguntar)
2. El cliente envía otra consulta a uno de los servidores de nivel superior que gestionan el subdominio .es preguntando por la dirección IP del servidor que gestiona .uam.es
3. El cliente redirige la consulta a dicho servidor preguntado por .ii.uam.es y contesta con su IP
4. Por último, se realiza una consulta de www a ii.uam.es que ya es un servicio determinado.
5. Con la IP de esta máquina la aplicación que solicitó la consulta (por ejemplo un navegador web) puede conectarse a su destino.

Este proceso se conoce como DNS estándar y saturaría Internet y los servidores de nombres por lo que en la práctica se utiliza combinado con un proceso de almacenado en **caché** de direcciones de Internet por todos los elementos implicados (desde el resolutor de nombres de nuestro ordenador) y lo que se denomina **DNS recursivo**.

El almacenamiento en caché se produce cada vez que se realiza una consulta. Los dispositivos por los que pasa la información almacenan toda la que van recibiendo (no solo la relacionada directamente con la URL o IP consultada).

El DNS recursivo básicamente permite que uno de estos elementos resuelva las IPs de dominios de otra gente. Si por ejemplo estás en el ordenador de tu casa y quieres buscar algo en google. Tecleas www.google.es y tu navegador primero intenta ver si él mismo sabe la IP de Google España. Si no la sabe se la pregunta al sistema operativo que si tampoco lo sabe la preguntará a los servidores de tu ISP (Internet Services Provider, por ejemplo Telefónica o Jazztel) que si no lo saben ya recorrerán el camino contado anteriormente. La gran diferencia es que en este modelo si el servidor a quién preguntas no sabe la respuesta se encarga él de buscarla (y almacenarla en caché por si le hacen la misma consulta) haciendo de cliente que pregunta a otro servidor. Este proceso se repite hasta que se obtiene una respuesta concreta; generalmente la IP buscada pero también puede ser que la URL consultada no existe.

Este proceso no funcionaría bien sin un tiempo de vida (TTL) de las direcciones almacenadas en caché tras el que se marcarán como no válidas y se volverán a consultar con la primera petición que llegue.

Para todo este proceso se crean zonas directas que contienen registros SOA y NS y pueden contener cualquier otro menos PTR. Vemos los registros un poco más abajo.

Reverse Lookup

También existe el proceso inverso (*reverse* DNS) por el que se pueden obtener los nombres de dominio asociados a una dirección IP determinada. Estas consultas siguen el mismo proceso, pero usan un dominio específico que se llama *in-addr.arpa* (*ip6.arpa* para IPv6). Las direcciones IP se representan invertidas como un nombre de dominio. Por ejemplo la dirección 123.45.67.89 se representaría como 89.67.45.123.in-addr.arpa

Primero se consultaría a quién está asociada 123 y así sucesivamente.

La zona que se debe crear en este caso se llama zona inversa y suele contener registros SOA, NS, PTR y CNAME.

El comando nslookup sirve para hacer consultas DNS. Funciona en Windows, MAC o Linux. Usando la propia ayuda del comando descubre cómo se realiza: una consulta directa, una inversa, una consulta a un servidor DNS concreto (no al por defecto)

Parámetros de configuración y registros del servidor de nombres afectados en el despliegue

Como nos ha hecho falta en varias ocasiones a lo largo del curso, la instalación y configuración de un servidor web se incluyó en los apéndices. Sin embargo faltaba la parte teórica que nos permitirá entender qué hacemos y por qué así como mejorar la configuración si fuera necesario.

En el apartado anterior ya hemos visto para qué se utilizan las zonas directa e inversa y por qué son necesarias. Además, podemos comprender para qué se utilizan los *forwarders* como medio para resolver las consultas que no conocemos, siendo estos *forwarders* generalmente servidores de nombres (DNS) de un ISP u otras organizaciones. Por ejemplo Google tiene servidores de nombres con las direcciones 8.8.8.8 y 8.8.4.4

Sin embargo hay una parte que no ha quedado nada clara y son los registros de los servidores DNS. Un *Resource Record* (RR) es el elemento más básico de DNS. Los registros definen las características de una zona o dominio y se usan en las consultas de nombres. Cada registro tiene un tiempo de vida (TTL *Time To Live*) que indica cuándo dejará de ser válido. En definitiva los registros se utilizan para asociar una URL a una IP.

Los registros tienen un formato estándar: *Owner TTL Class Type RDATA* donde

- **Owner**: es el nombre de la máquina o del dominio DNS al que pertenece este recurso.
- **TTL** (*Time To Live*): El tiempo en segundos que debe mantenerse este registro en caché. Si no aparece se usa el TTL mínimo del registro SOA.
- **Class**: Indica que familia de protocolos se usa. En Internet es casi exclusivamente IN.
- **Type**: El tipo del RR.
- **RDATA**: Los datos del recurso que indica el registro. Por ejemplo una dirección IP.

Hay muchos tipos diferentes de registros pero los más importantes son:

- **SOA**: Cualquier zona que se cree en un servidor DNS debe contener un registro SOA (*Start of Authority*) al inicio de la definición. Este tipo de registro contiene los parámetros:
  - Owner , TTL , Class y Type como vimos arriba.
  - Authoritative server (Servidor Autorizado): El servidor DNS primario de la zona.
  - Responsible person (persona responsable): correo electrónico de la persona responsable de la zona. Usa un punto en lugar de arroba.
  - Serial number (número de serie): muestra cuántas veces ha sido actualizada la zona.
  - Refresh (refresco): indica como de frecuentemente tienen los servidores DNS secundarios de la zona que consultar si ésta ha cambiado.
  - Retry (reintento): cuánto esperará un servidor secundario la respuesta antes de volver a consultar posibles cambios en la zona al servidor primario.
  - Expire (expiración): cuánto tiempo desde la última actualización sigue siendo válida la información de un servidor secundario.
  - Minimum TTL: El tiempo de vida que se aplica a todos los registros de la zona si no tienen uno específico. Ya hemos visto antes que cada registro puede llevar uno propio pero que es optativo.

En los apéndices se encuentran ejemplos de configuración y aquí podemos ver otro

```
prueba.ejemplo.com. IN SOA (
 ns1.prueba.ejemplo.com. ; servidor autorizado para la zona
 sergio.cuesta.prueba.ejemplo.com. ; e-mail del administrador de zona

 5099 ; serial number
 3600 ; refresh (1 hora)
```

```
    600 ; retry (10 mins)
    86400 ; expire (1 día)
    60 ) ; minimum TTL (1 min)
```

- **NS**: Los registros NS indican los servidores de nombres autorizados para una zona. Pueden indicar servidores tanto primarios como secundarios y una zona tiene que tener al menos un registro NS con el servidor primario. En caso de haber zonas delegadas indican los servidores de dichas zonas también.
  Así la siguiente línea tendría que aparecer en los servidores de nombres de ejemplo.com y prueba.ejemplo.com

```
prueba.ejemplo.com. IN NS ns1.prueba.ejemplo.com.
```

- **A**: Los registros de dirección (*Address*) asocian un nombre de dominio completo (*Fully Qualified Domain Name o FQDN*) con una dirección IP.

```
ns1 IN A 123.45.67.89
```

- **PTR**: Los registros puntero (*PoinTeR*) hacen lo contrario a los registros A. Asocian una dirección IP con el FQDN.

```
89.67.45.123.in-addr.arpa. IN PTR ns1.prueba.ejemplo.com.
```

o

```
89    IN PTR ns1.prueba.ejemplo.com.
```

- **CNAME**: Los registros de nombres canónicos crean un alias. Suelen utilizarse para ocultar la estructura de nuestro sitio. Por ejemplo si nosotros tenemos un servidor FTP que se llama ftp1 y luego lo cambiaremos a otro no queremos que los usuarios se enteren. También sirve para que varios FQDN se asocien a la misma máquina: varios host virtuales o varios servicios como ftp y www.

```
ftp.prueba.ejemplo.com. IN CNAME ftp1.prueba.ejemplo.com.
```

- **MX**: Este registro es específico para servidores de intercambio de correo (*Mail Exchange*). Un servidor de intercambio de correo se encarga de procesar y redirigir los correos de un dominio. Procesar el correo puede significar entregarlo a la dirección adecuada dentro de nuestro dominio o pasarlo a un tipo diferente de trasporte de correo. Redirigir el correo significa enviárselo al servidor de correo de destino, directamente o a través de otros que se encuentren en el camino mediante *Simple Mail Transfer Protocol* (SMTP). Puedes tener varios servidores de correo en el mismo dominio por lo que es posible tener muchos registros MX.

```
*.prueba.ejemplo.com. IN MX 0 mailserver1.prueba.ejemplo.com.
*.prueba.ejemplo.com. IN MX 10 mailserver2.prueba.ejemplo.com.
*.prueba.ejemplo.com. IN MX 10 mailserver3.prueba.ejemplo.com.
```

El número que aparece después de MX es la prioridad. En este caso tenemos un servidor principal con la máxima prioridad y dos delegados con igual prioridad entre ellos. Si un correo llega y el primero no está activo se le pasaría al segundo o al tercero indistintamente. Si tenemos un gran tráfico de correo se pueden establecer todos con máxima prioridad para evitar saturaciones.

El comando dig nos muestra información de este tipo de recursos en una forma muy similar a la que usamos para configurar el servidor DNS. Investiga el comando y comenta sus opciones más útiles. Después contesta a las preguntas de más abajo.

¿Qué pasa si escribes dig sin más? ¿Cómo puedes indicarle a dig el servidor DNS que debe usar?