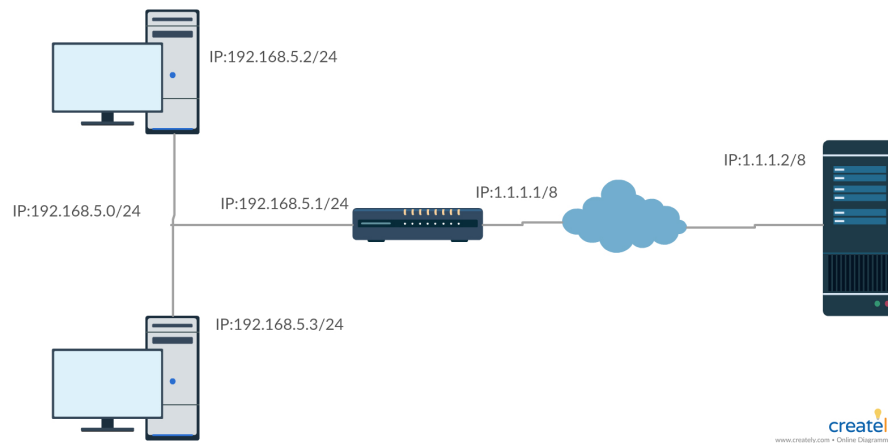


Network Address Translation

Antunez Joaquin, Gonzalez Alejo, Nielsen Maximiliano

Junio 2019

Somos el encargado de redes en la empresa “noQueremosLaburarAsiQueRobamosCodigo”. En esta empresa contamos únicamente con 2 equipos, uno que corresponde al pasante de la empresa y otro a Martín el desarrollador principal. En esta empresa contamos con un servicio especial por parte de “StackNotInOverflow” que nos provee con código de asistencia para el desarrollo de nuestra aplicación. “StackNotInOverflow” nos provee el código mediante una página web que está hosteada en un servidor apache. Los hosts de nuestra red poseen todas ip privadas, mientras que el servidor de “StackNotInOverflow” está bajo una ip pública. Este es el gráfico de nuestra red:



El host 2 de la red privada corresponde a Martín y el 3 al pasante.

INTRODUCCION TEORICA DE NAT

Introducción Lab Nat con IPtables

¿Qué es NAT?

El proceso de Network Address Translation es un mecanismo usado por los routers IP para que dos redes puedan intercambiar paquetes aunque tengan direcciones incompatibles. En una estructura donde varios hosts tienen direcciones IP privadas (red interna), cuando estos quieren enviar paquetes fuera del router NAT se encarga de traducir esa dirección interna (de rango privado) en una externa (de rango público). Cuando se creó el Internet en el año 1969, no se lo pensó con la magnitud que hoy tiene. El protocolo IPv4 consta de 32 bits y, a día de hoy, un número limitado de direcciones IP; por eso es tan necesaria la NAT. Gracias a la misma se logra que, por ejemplo, en una red de una empresa

donde hay cientos de computadoras, se arme una red interna donde cada host tiene una dirección interna (privada) y así se tenga solo una dirección pública o a lo sumo unas pocas más, en vez de tener cientos de direcciones públicas. Esto es fundamental para el ahorro de direcciones IP públicas IPv4 ya que estas, como todos sabemos, no son infinitas y en algún momento se van a agotar. Los usuarios internos utilizan normalmente la Source NAT para acceder a Internet; la dirección de origen se traduce y por lo tanto se mantiene privada. El NAT de destino se realiza en los paquetes entrantes cuando el firewall traduce una dirección de destino a una dirección de destino diferente; por ejemplo, traduce una dirección de destino pública a una dirección de destino privada.

Tipos de NAT.

Basic Nat:

Es el tipo más simple de NAT, provee una traducción de IP de tipo uno a uno. También es llamado como One-to-one nat. En este tipo de NAT solo la dirección IP, IP header checksum y cualquier otro checksum de más alto nivel que incluya la dirección IP son modificados. Este tipo de NAT puede usarse para conectar 2 redes IP que tengan direcciones incompatibles

One-to-many NAT:

La mayoría de NATs mapea múltiples host privados a una dirección IP pública. En una configuración típica, una red local usa una de las direcciones IP privadas designadas. El router esta también tiene una dirección de la red privada pero también está conectado a internet con una dirección pública. A medida que viaja información desde la red local a internet, la dirección de cada paquete es traducida de una dirección privada a la dirección pública. El router mantiene información básica sobre cada conexión activa. cuando una respuesta vuelve al router este usa la información de la conexión, que mantuvo desde que se comenzó la conexión, para enviar la respuesta a la dirección privada correcta

Métodos de traducción.

Full-cone NAT:

- Cuando una dirección interna ($iAddr:iPort$) es mapeada a una dirección externa ($eAddr:ePort$), cualquier paquete de $iAddr:iPort$ es enviado a $eAddr:ePort$
- Cualquier host puede enviar paquetes a $iAddr:iPort$ enviando paquetes a $eAddr:ePort$

(Address)-restricted-cone NAT:

- Cuando una dirección interna ($iAddr:iPort$) es mapeada a una dirección externa ($eAddr:ePort$), cualquier paquete de $iAddr:iPort$ es enviado a $eAddr:ePort$
- Un host externo ($hAddr:any$) puede enviar paquetes a $iAddr:iPort$ enviando paquetes a $eAddr:ePort$ solo si $iAddr:iPort$ este envió previamente un paquete a $hAddr:any$

Port-restricted cone NAT:

Es como Address-restricted-cone NAT pero además restringe el puerto.

- Cuando una dirección interna ($iAddr:iPort$) es mapeada a una dirección externa ($eAddr:ePort$), cualquier paquete de $iAddr:iPort$ es enviado a $eAddr:ePort$
- Un host externo ($hAddr:hPort$) puede enviar paquetes a $iAddr:iPort$ enviando paquetes a $eAddr:ePort$ solo si $iAddr:iPort$ este envió previamente un paquete a $hAddr:hPort$

Symmetric NAT:

- Cada solicitud desde la misma dirección IP interna y puerto a una dirección IP y puerto de destino específicos se asigna a una única dirección IP externa de origen y puerto; Si el mismo host interno envía un paquete incluso con la misma dirección y puerto de origen pero a un destino diferente, se utiliza una asignación diferente.
- Solo un host externo que recibe un paquete de un host interno puede devolver un paquete.

NAT está soportado por iptables mediante la tabla llamada nat y los objetivos SNAT, DNAT y MASQUERADE.

SNAT (Source NAT) implementa la traducción de la dirección de origen de paquetes que salientes (y la transformación correspondiente del tráfico de respuesta).

DNAT (Destination NAT) implementa la traducción de la dirección de destino de los paquetes entrantes (y la transformación correspondiente del tráfico de respuesta).

MASQUERADE is similar a SNAT, pero se usa en situaciones donde la dirección objetivo de la traducción es desconocida (ej: cuando se usa DHCP).

Para poder conectarnos al servidor tenes que habilitar la conexión usando DNAT.

```
iptables -t nat -A PREROUTING -p tcp -d 192.168.5.1 -dport 80 -j DNAT  
-to-destination 1.1.1.2:80
```

Este comando se traduce los intento de conexión TCP al puerto 80 en 192.168.5.1 al puerto 80 al puerto 80del host 1.1.1.2. También necesitaremos usar SNAT:

```
iptables -t nat -A POSTROUTING -s 192.168.5.0/24 -j SNAT -to-source  
1.1.1.1
```

Esta cadena permite que el tráfico que tenga origen en la red 192.168.5.0/24 se lo traduzca a tener origen en 1.1.1.1 (la dirección pública del router).

Para poder ver las cadenas que añadimos usaremos el comando:

```
iptables -t nat -L
```

Ahora solo nos queda iniciar el servidor apache:

```
/etc/init.d/apache2 start
```

Y ahora pondremos a monitorear al acceso al servidor:

```
tail -f /var/log/apache2/access.log
```

Ahora intentaremos conectarnos al servidor mediante la PC1:

```
links 1.1.1.2
```

Tendríamos que ver "It works!".