

Si el tráfico entra a tu servidor a través del balanceador de carga HAProxy y tienes como backend máquinas virtuales o servidores físicos con Apache, seguramente habrás notado que Apache no lee el tráfico como proveniente de la IP real del cliente sino como proveniente de la IP del host donde está instalado HAProxy.

Para que Apache sepa de que IP real provienen las peticiones que le llegan a través de HAProxy, se debe:

### PRIMERO

Indicar a HAProxy que marque con la cabecera **X-Forwarded-For** todos los paquetes que va a enviar a los respectivos servidores. Y eso se puede hacer editando el archivo de configuración de HAProxy de dos formas:

O bien agregando

```
option forwardfor
```

... al frontend o al backend, o bien haciéndolo mediante http-request agregando

```
http-request add-header X-Forwarded-For %[src]
```

...al frontend o al backend para luego, en ambos casos, reiniciar el servicio HAProxy para aplicar los cambios y proceder a lo

### SEGUNDO

**Solución A:** Hacer que Apache lea la IP indicada en la cabecera X-Forwarded-For, es decir, la del cliente real, modificando el código de la página web de forma que, en PHP, en vez de buscar la IP mediante `$_SERVER[«REMOTE_ADDR»]` (que nos daría como resultado la IP del host de HAProxy) la busque mediante `$_SERVER[«HTTP_X_FORWARDED_FOR»]`. Pero eso es un coñazo. Y lo mismo ocurre con los logs. Para que Apache lea de los logs las IPs reales de los clientes conectados a HAProxy se suele modificar la configuración del almacenamiento de los logs. Pero hay que andar tocando historias que no te permiten, por ejemplo, pasar una máquina virtual a un sitio donde no esté detrás de HAProxy porque, de hacerlo, tendrías que volver a tocar el código de la web y el archivo de configuración de los logs. Lo peor es que así se hizo durante mucho tiempo, pero por suerte hoy te traigo dos soluciones (B y C) para este incordioso segundo paso:

**Solución B:** Hacer que Apache lea la IP indicada en la cabecera X-Forwarded-For, es decir, la del cliente real, mediante la instalación de un módulo de Apache que, una vez configurado y activado, hará el trabajo de «derivar» la IP a la que preguntas en PHP con `$_SERVER[«HTTP_X_FORWARDED_FOR»]` dentro de `$_SERVER[«REMOTE_ADDR»]`, de forma que si preguntas por la segunda, en vez de devolverte la IP del host de HAProxy, te devolverá la IP real del cliente porque la sacará de la cabecera X-Forwarded-For. Esto se hace en

#### Apache 2.4 o superior (Debian 8 y 9)

... instalando, configurando y activando el módulo **remoteip** con:

(Si has incrustado la cabecera **X-Forwarded-For** y la IP del host de HAProxy esta en la red pública):

```
a2enmod remoteip
a2enmod headers
echo "RemoteIPHeader X-Forwarded-For" > /etc/apache2/conf-available/remoteip.conf
echo "RemoteIPTrustedProxy 85.86.87.88" >> /etc/apache2/conf-available/remoteip.conf
echo 'LogFormat "%a %l %u %t \"%r\" %>s %0 \"%{Referer}i\" \"%{User-Agent}i\" combined' >> /etc/apache2/conf-available/remoteip.conf
a2enconf remoteip
service apache2 restart
```

...donde 85.86.87.88 es la IP del host donde está instalado HAProxy, y con:

(Si has incrustado la cabecera **X-Forwarded-For** y la IP del host de HAProxy esta en la red privada):

```
a2enmod remoteip
a2enmod headers
echo "RemoteIPHeader X-Forwarded-For" > /etc/apache2/conf-available/remoteip.conf
```

```
echo "RemoteIPInternalProxy 192.168.0.10" >> /etc/apache2/conf-available/remoteip.conf
echo 'LogFormat "%a %l %u %t \"%r\" %>s %0 \"%{Referer}i\" \"%{User-Agent}i\"" combined' >> /etc/apache2/conf-
available/remoteip.conf
a2enconf remoteip
service apache2 restart
```

...donde 192.168.0.10 es la IP del host donde está instalado HAProxy. En

### Apache 2.3 o inferior (Debian 7)

Instalando, configurando y activando el módulo **RPAF** (**R**everse **P**roxy **A**dd **F**orward) con:

```
apt-get -y install libapache2-mod-rpaf
a2enmod rpaf
service apache2 restart
```

Es posible que para que funcione RPAF tengas que modificar su archivo de configuración quitando los ifs de forma que en vez de quedar así:

```
<IfModule mod_rpaf.c>
RPAFenable On
RPAFsethostname On
RPAFproxy_ips 127.0.0.1 ::1
</IfModule>
```

...quede así:

```
RPAFenable On
RPAFsethostname On
RPAFproxy_ips 127.0.0.1 ::1
```

Después de ejecutar la solución B, al preguntar por `$_SERVER[«REMOTE_ADDR»]`, PHP responderá con la IP del cliente, tal cual se la ha indicado la cabecera X-Forwarded-For. Pero el problema entonces es que si también preguntas por `$_SERVER[«HTTP_X_FORWARDED_FOR»]` el resultado será nulo. Es decir, ninguna IP. Porque digamos que los módulos hacen un «cortar y pegar», no un «copiar y pegar» (explicado burdamente), y a lo mejor puedes tener alguna razón para querer que, si pides por `$_SERVER[«HTTP_X_FORWARDED_FOR»]`, se te responda con la IP del cliente. Imaginemos que tienes páginas web con código modificado gracias a que has estado años sufriendo el incordio descrito arriba y que no quieres volver a pasar el coñazo de modificarlo otra vez. Para ello entonces viene la

**Solución C (La de NiPeGun):** Incrustar la cabecera **X-Forwarded-For** pero TAMBIÉN incrustar la cabecera **X-Client-IP**.

Si tienes un poco de conocimiento de HAProxy, sabrás que puedes incrustar X-Forwarded-For con:

```
option forwardfor
```

..., como hemos visto arriba, y **X-Client-IP** con:

```
option forwardfor header X-Client-IP
```

Pero NO puedes incrustar las dos cabeceras indicando ambas opciones al mismo tiempo. Se debe hacer de esta forma:

```
option forwardfor
http-request add-header X-Client-IP %[src]
```

Nótese que en este caso no se utilizó **set-header** sino **add-header**. De haber utilizado **set-header**, se habría borrado el header X-Forwarded-

For pasado por **option forwardfor**, dado que set-header primero borra todas las cabeceras, para luego incrustar la que se le indique.

Entonces si, ahora, deberíamos instalar, configurar y activar los módulos de la solución B, pero indicando X-Client-IP, no X-Forwarded-For. Hazlo de esta forma. En

#### Apache 2.4 o superior (Debian 8 y 9)

Si la IP del host de HAProxy está en la red pública:

```
a2enmod remoteip
a2enmod headers
echo "RemoteIPHeader X-Client-IP" > /etc/apache2/conf-available/remoteip.conf
echo "RemoteIPTrustedProxy 85.86.87.88" >> /etc/apache2/conf-available/remoteip.conf
echo 'LogFormat "%a %l %u %t \"%r\" %>s %0 \"%{Referer}i\" \"%{User-Agent}i\"" combined' >> /etc/apache2/conf-
available/remoteip.conf
a2enconf remoteip
service apache2 restart
```

...donde 85.86.87.88 es la IP del host donde está instalado HAProxy.

Si la IP del host de HAProxy está en la red privada:

```
a2enmod remoteip
a2enmod headers
echo "RemoteIPHeader X-Client-IP" > /etc/apache2/conf-available/remoteip.conf
echo "RemoteIPInternalProxy 192.168.0.10" >> /etc/apache2/conf-available/remoteip.conf
echo 'LogFormat "%a %l %u %t \"%r\" %>s %0 \"%{Referer}i\" \"%{User-Agent}i\"" combined' >> /etc/apache2/conf-
available/remoteip.conf
a2enconf remoteip
service apache2 restart
```

...donde 192.168.0.10 es la IP del host donde está instalado HAProxy. Para

#### Apache 2.3 o inferior (Debian 7)

Simplemente haciéndolo igual que en la solución B.

Ahora si, cuando en PHP preguntes tanto por `$_SERVER[«HTTP_X_FORWARDED_FOR»]` como por `$_SERVER[«REMOTE_ADDR»]` el resultado será el mismo: **la IP real del cliente**.