Admins - Anomalie #1840

Propagation des adresses IP à travers le proxypass nginx

21/12/2016 16:09 - Quentin Gibeaux

Statut:	Fermé	**Début:**	21/12/2016
Priorité:	Normale	**Echéance:**	
Assigné à:	François Poulain	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0.00 heure
Version cible:		**Temps passé:**	2.00 heures
Difficulté:	2 Facile		

Description

Cet après-midi on a remarqué avec CPM que certaines requêtes n'avaient pas les adresses IP des clients dans les logs apache de drupal6, mais celle du bastion nginx.
Cela ne viendrait pas du fait que seules les routes https sont configurées avec les champs suivants ?

```
        proxy_set_header Host       $http_host;
        proxy_set_header X-Forwarded-Proto $scheme;
```

Il y a une raison particulière à ne pas ajouter ces champs également à la configuration sur le port 80, ou est-ce un simple oubli ?

Historique

#1 - 28/01/2017 14:17 - François Poulain

Non, ce qui est utile est ça :

```
/etc/nginx/conf.d/local.conf : proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
```

Le X-Forwarded-Proto sert à faire connaître au serveur backend que le protocole est https. Mais plein d'applications ignorent cette variable.

J'ai créé (le 23 nov dernier) sur chaque apache une conf de log qui utilise le X-Forwarded-For. Cette conf est utilisable en mettant :

```
CustomLog ${APACHE_LOG_DIR}/www.april.org.access.log combined-proxy
```

au lieu de l'habituel

```
CustomLog ${APACHE_LOG_DIR}/www.april.org.access.log combined
```

Ceci étant, effectivement, pour une raison qui m'échappe pour le moment, le logs ne fonctionnent effectivement pas comme attendu avec des requêtes https.

#2 - 28/01/2017 14:34 - François Poulain

J'ai temporairement commi

```
root@drupal6:/etc# cat /var/lib/drupal6/files/eiFeiK5o.php

<?php
$headers = apache_request_headers();

foreach ($headers as $header => $value) {
        echo "$header: $value <br />\n";
}
?>
```

Ça donne ceci :

```
$ curl http://april.org/sites/default/files/eiFeiK5o.php

Host: april.org <br />
HTTP_CLIENT_IP: 82.67.196.191 <br />
Connection: close <br />
```

```
User-Agent: curl/7.38.0 <br />
Accept: */* <br />

$ curl https://april.org/sites/default/files/eiFeiK5o.php

Host: april.org <br />
X-Forwarded-Proto: https <br />
Connection: close <br />
User-Agent: curl/7.38.0 <br />
Accept: */* <br />
```

#3 - 28/01/2017 14:47 - François Poulain

Je pense que le nœud du problème se situe ici : la conf local.conf n'est pas chargé par nginx car dans nginx.conf les inclusions de conf.d/*.conf ne sont faites que pour le proto http.

Si c'est le cas, j'ai du mal à savoir comment le serveur tient debout. :)

#4 - 28/01/2017 17:59 - Vincent-Xavier JUMEL

On voit ici où là l'utilisation de proxy_set_header X-Real-IP $remote_addr;

À titre perso, j'ajouterai bien

```
    proxy_set_header  X-Real-IP $remote_addr;
    proxy_set_header  X-Forwarded-Proto https;
    proxy_set_header  X-Forwarded-For $remote_addr;
    proxy_set_header  X-Forwarded-Host $remote_addr;
```

qui me semble correspondre à l'ancienne configuration.

#5 - 28/01/2017 20:13 - Vincent-Xavier JUMEL

Bon, après quelques tests avec tcpdump pour lire le trafic réseau entre le frontal et l'applicatif, il semble qu'il faille rajouter proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for; dans le bloc location <uri> {} kivabien.

Je ne suis pas sûr qu'un include /etc/nginx/proxy_set_header.conf; soit possible et en tout cas un fichier dans /etc/nginx/conf.d/ ne fonctionne pas. Du moins, je ne sais pas comment nginx lit les directives de configuration et quelles précédences elles ont.

#6 - 28/01/2017 20:16 - Vincent-Xavier JUMEL

François Poulain a écrit :

> Je pense que le nœud du problème se situe ici : la conf local.conf n'est pas chargé par nginx car dans nginx.conf les inclusions de conf.d/*.conf ne sont faites que pour le proto http.
>
> Si c'est le cas, j'ai du mal à savoir comment le serveur tient debout. :)

De façon plus générale, la lecture des directives depuis les includes se fait de façon très hasardeuses.

#7 - 29/01/2017 15:27 - François Poulain

- Statut changé de Nouveau à Résolu

Le soucis est multiple :

D'une part « add_header » ne transmet pas au serveur final le header. Il ne fait que l'ajouter à la réponse en direction du client. Cf http://nginx.org/en/docs/http/ngx_http_headers_module.html#add_header :

> Adds the specified field to a **response** header

Donc clairement dans notre cas les nombreux add_header X_FORWARDED_PROTO https; trouvés dans la conf sont inutiles.

D'autre part il y a eu méprise sur le fonctionnement de proxy_set_header. On en retrouve un peu partout dans la conf, à différents niveaux de blocs. Or, la doc d'nginx précise :

> Allows redefining or appending fields to the request header passed to the proxied server. The value can contain text, variables, and their combinations. **These directives are inherited from the previous level if and only if there are no proxy_set_header directives defined on the current level**.

C'est écrit subtilement mais de fait c'est à prendre au sens strict : il n'y a pas d'héritage comme on pourrait l'attendre intuitivement.

Nous ne sommes pas les seuls à nous être fait avoir, cf par exemple
http://serverfault.com/questions/777363/inherit-proxy-set-header-when-using-it-in-location-block

J'ai donc centralisé le proxy_set_header X-Forwarded-Proto $scheme; dans conf.d/local.conf et retiré tous les proxy_set_header inutiles (et sources de bugs) des blocs des sites.

La conf fonctionne pour le drupal (les logs http et https mentionent bien la bonne ip), le icinga est au vert, et au passage ça a résolu le bug #1350.

#8 - 29/05/2019 12:17 - Quentin Gibeaux

- Statut changé de Résolu à Fermé

#9 - 26/12/2020 02:47 - Christian P. Momon

- Assigné à mis à François Poulain