Daugnaix Loic
De Groodt Alexandre
Dubois Alexandre
Hallantie Emma
Moczulski Alan

Rapport d'Implémentation

Choix d'implémentation

Point de départ: un sous domaine qui gère la procédure de connexion et le site de shopping.
Nous avons choisi d'implémenter notre projet sous le système d'exploitation Linux, comme la majorité des serveurs dans le monde entier. L'implémentation a été faite sur une machine Ubuntu.

Une attaque DDOS (distributed denial-of-service) consiste à surcharger un système informatique connecté au réseau en envoyant un très grand nombre de paquets en très peu de temps, en continu. Pour se protéger, le système est souvent obligé de redémarrer, et c'est cela que nous voulons éviter car si le temps de down-time est trop élevé les clients potentiels du service ne seront plus intéressés par le service. Ce temps doit rester relativement bas pour éviter ce genre de situation.
Pour permettre justement de minimiser la perte de clients pendant une attaque DDOS, nous avons eu l'idée de créer deux sites: l'un étant un sous-domaine de l'autre. Le premier contient la page de connexion et le deuxième contient notre service principal: la plateforme d'achat en ligne.

Le serveur en utilisation aura simplement un **.htaccess** changé, qui forcera une redirection vers le site de login si on ne vient pas déjà de celui-ci. Pour un client lambda, le changement est simplement de devoir se connecter obligatoirement au début. Pour le serveur, il s'agit juste d'un sous-domaine.

L'implémentation sur machine peut se faire de deux manières différentes. Le site principal sera forcément lancé sur une machine, et le site de login sera soit lancé sur une deuxième machine, soit sur une machine virtuelle contenue dans la première machine. La première option nécessite d'acheter un second ordinateur, mais semble plus sûre, les deux machines n'étant pas physiquement liées.

Les aspects techniques

- Nous avons un fichier de configuration pour le site principal (main) et un fichier de configuration pour le site de login. Pour chacun d'eux, le site est indiqué, on précise que ce sont des masters et on précise également le fichier .zone. Enfin, pour l'aspect sécurité, le site principal redirige vers le sous

domaine lorsqu'on tente d'aller vers celui-ci directement, tandis que le site de login n'a pas de restriction sur qui peut lui envoyer des requêtes.
- Il n'y a aucune commande qui doit être exécutée pendant que le script tourne, donc aucune commande particulièrement "utile" à part celle de démarrage de bind.
- Aucun processus n'a besoin de tourner en mémoire vu qu'il s'agit d'un site en ligne.
- Nous n'avons pas eu besoin d'utiliser des ports
- Il faut lancer la commande "*systemctl enable bind9"* pour que bind se démarre automatiquement au lancement de la machine et "*systemctl start bind9"* pour démarrer bind.

Les étapes d'implémentation

Nous partons du principe qu'il y a déjà un serveur qui tourne avec le site principal. Afin de lancer le serveur avec BIND, voici ce qu'il faut faire sur la nouvelle machine pour le sous-domaine (ou simplement une autre machine virtuelle sur le même ordinateur).

Pour cette nouvelle machine, qu'elle soit virtuelle où réelle, il faut tout d'abord installer le package **bind9**, à l'aide de la commande suivante sous Ubuntu: **sudo apt install bind9**.
Il faut bien-sûr un domaine fonctionnel, ce que nous avons déjà dans ce cas.

Ensuite, nous devons aller dans le dossier **etc/bind**, c'est ici que se trouve la majorité des fichiers de configuration que nous devrons modifier.

Tout d'abord, nous avons besoin de créer le sous-domaine, voici comment faire: on doit placer les fichiers **login/etc/bind/named.conf** et **login/etc/bind/login.shopandget.com.zone** dans le dossier **etc/bind** de la nouvelle machine.

Nous devons ensuite modifier le fichier **named.conf.options** et autoriser tout type de requêtes en ajoutant cette ligne: **allow-query { localhost; }**.

Ensuite, nous devons modifier le fichier **named.conf.local** et inclure le fichier **named.conf** dans le fichier en ajoutant la ligne: **#include "named.conf"** au début du fichier.

Le fichier **etc/resolv.conf** doit être modifié par celui donné.

A ce point, toutes les étapes ont été réalisées pour cette nouvelle machine, et on peut donc commencer à changer le site principal (avant de pouvoir démarrer le sous domaine):

Sur la machine principale, le fichier **etc/bind/shopandget.com.zone** devra être modifié également afin d'ajouter le sous-domaine, en ajoutant ceci à la fin: `*login IN A 10.0.1.7 ; login ns*`
Une fois que la partie login (qui est sûrement sur une page web à part) a été extraite sur un site web indépendant, il suffit de mettre la nouvelle version du site web principal en ligne, et une fois que c'est et que BIND a été redémarré comme précisé ci dessous, il faut mettre le fichier **.htaccess** à la racine du site principal. Il faut bien sûr autoriser le .htaccess si ce n'est pas déjà fait, en modifiant le httpd.conf pour ajouter, après le </VirtualHost>:
<Directory /var/www/shopandget.com/public_html>
Options Indexes FollowSymLinks
AllowOverride All
Require all granted
</Directory>

Pour finir, il faut relancer le service BIND sur les deux machines afin que les modifications apportées aux fichiers de configuration soient prises en compte, avec cette commande:
systemctl restart bind9.
Cette commande peut aussi être utilisée pour redémarrer les serveurs sur la machine en cas de problème.

Si l'on veut que le service BIND soit lancé a chaque démarrage du système il faut exécuter la commande:
systemctl enable bind9.

En clair, il faut héberger le sous-domaine et mettre le **.htaccess** sur le site principal ainsi que rajouter une ligne dans le fichier .zone de celui-ci.

Opération de maintenance

Il ne devrait pas y avoir de maintenance nécessaire en soi, le seul problème potentiel est justement que le site de login soit DDOS, mais cela n'empêche pas aux clients de continuer à faire leurs courses, il s'agit juste de redémarrer le serveur comme n'importe lequel qui serait frappé par une DDOS (mais au moins le site principal ne devrait pas y être vulnérable).

Pour annuler les changements, il suffit juste d'enlever le **.htaccess** du site principal. On peut ensuite remettre le sous-domaine en tant que page web.

Les fichiers de configuration

Tous les fichiers décrits ci dessous sont dans le zip. Nous ne décrivons pas les **.html** ni **.php** puisque ce n'était pas l'objectif du projet, et il s'agit d'un cas assez standard

de page de login redirigeant vers une autre page, à ceci près que la page de login appartient à un sous domaine.

Le .htaccess du site principal est le point clef de ce projet, en effet il permet de rediriger tout visiteur qui se dirige vers le site principal vers le site de login, sans nécessairement venir du site de login.
Bien entendu le site de login ne redirige que si le visiteur a un compte, et on suppose qu'il y a un protection anti-bot au niveau de la création des comptes.
Le code 302 signale que la redirection est permanente, et le premier site est celui dont on vient alors que le suivant est le target de la redirection.

```
<Limit GET>
order deny,allow
deny from all
allow from login.shopandget.com
</Limit>

<If "%{HTTP_HOST} != 'www.login.shopandget.com'">
Redirect 302 http://shopandget.com http://login.shopandget.com
</If>
```

Les fichiers .conf:

Les fichiers named.conf (ici mis l'un à la suite de l'autre) donnent la localisation des fichiers **.zone**. Il est possible de limiter les requêtes du site principal à celles venant du secondaire, mais celà voudrait dire que les clients n'auraient plus eu accès à l'ancien site, ce qui n'est pas souhaitable.

```
zone "shopandget.com" IN {
        type master;
        file "etc/bind/shopandget.com.zone";
        allow-query { any; };
};
```

etc/bind/named.conf principal

```
zone "login.shopandget.com" IN {
        type master;
        file "etc/bind/login.shopandget.com.zone";
        allow-query { any; };
};
```

etc/bind/named.conf du login

Les fichiers resolv.conf (mis en suivant ici aussi) servent simplement à signaler les adresses du domaine principal et du sous domaine pour qu'elles soient accessibles.

```
search domain shopandget.com
nameserver 10.0.0.0
```

etc/resolv.conf principal

```
search domain login.shopandget.com
nameserver 10.0.0.7
```
etc/resolv.conf du login

Les fichiers zones:

Ci-dessous se trouve le fichier principal, celui qui s'occupe du site web où les achats sont faits. C'est un fichier **.zone** assez standard, nous avons rajouté le login qui doit bien-sûr rentrer dans les adresses allouées au domaine, puisque c'est un sous-domaine. Dans ce cas-ci, le domaine étant **10.0.1.0/16**, nous avons accès à toutes les IP entre **10.0.1.0** et **10.0.1.15**, celle finissant par **.7** est donc comprise.
Deux fichiers de zone sont nécessaires car pour une meilleure sécurité, nous avons placé le site de login dans un sous-domaine du domaine du site principal.

Le **$ORIGIN** dans chaque fichier .zone précise que les informations se rapportent à notre site shopandget.com. Les temps de rafraîchissement, de mise en cache, les délais, d'expiration etc. sont plus courts pour le site principal car c'est le site actif sur lequel les clients font leurs courses, il est primordial qu'il soit régulièrement rafraîchi. Pour le site de login c'est moins important, voilà pourquoi les temps sont un peu plus élevés. Cependant, le numéro serial est très petit chez le site de login car il vient d'être implémenté et est incrémenté à chaque mise à jour du site.

Le fichier **.zone** du site principal (*etc/bind/shopandget.com.zone*):

```
1  ; for 10.0.1.0/16
2  $ORIGIN shopandget.com.
3  $TTL 86400          ; 1 day
4  shopandget.com      IN     SOA   main.shopandget.com.   alexandre.dubois.shopanget.com (
5                                    2001062501             ; serial
6                                    3600                   ; refresh after 1 hour
7                                    600                    ; retry after 10 minuts
8                                    604800                 ; expire after 1 week
9                                    86400                  ; minimum TTL of 1 day
10                                   )
11
12                          NS     main.shopandget.com.     ; primary master (authoritative)
13
14 $ORIGIN shopandget.com.
15 main                IN     A     10.0.1.0                ; main ns
16 login               IN     A     10.0.1.7                ; login ns
```

On spécifie que le serveur principal est le serveur de référence pour les données de la zone (faisant autorité). C'est le site principal **shopandget.com**. Juste après on retrouve le nom de l'administrateur système du serveur. Enfin, il connaît l'adresse IP du main site et du site de login.

Le fichier **.zone** du site de login (*etc/bind/login.shopandget.com.zone*):

```
$ORIGIN shopandget.com.
$TTL 86400          ; 1 day
login.shopandget.com       IN     SOA   main.login.shopandget.com.   alexandre.dubois.shopanget.com (
                                  2                          ; serial
                                  21600                      ; refresh after 6 hour
                                  3600                       ; retry after 1 hour
                                  604800                     ; expire after 1 week
                                  86400                      ; minimum TTL of 1 day
                                  )

                    IN     NS    main.login.shopandget.com.    ; master (authoritative)


login               IN     A     7                          ; 10.0.1.7
```

De même, le serveur de login fait autorité dans son domaine (le sous-domaine du main). C'est le site de login login.shopanget.com. Comme l'autre fichier, on retrouve ensuite le nom de l'administrateur système du serveur. La seule adresse connue est la sienne, celle du site de login.