De l'efficacité du fichier hosts

```
**************************************************************************
$Id       : hosts.odt $
$Source   : http://www.bobotig.fr/contenu/documents/hosts/ $
$Revision : 2 $
$Date     : 20101010 $
$Autor    : BoboTiG (www.bobotig.fr) $
$Contrib  : Alden MagiX, Lord, fifi et Amarina du forum d'AdZHosts.fr $
**************************************************************************
```

Sommaire

1. Descriptif

Le fichier hosts est un simple fichier texte qui est d'office présent dans votre système, que vous utilisiez Microsoft Windows, GNU/Linux, Mac ou Unix. Il contient les correspondances entre adresses IP[1] et un ou plusieurs nom(s) de domaine (comprendre par là un lien Internet comme « www.example.org » par exemple).

Étant donné qu'il s'agit d'un fichier universel, il est pris en compte quelque soit le navigateur que vous utilisez (Mozilla Firefox et ses dérivés, Internet Explorer, Google Chrome, Safari, Opera, Midori, et caetera) et la compatibilité de son contenu est totale.

[1] *Numéro qui identifie chaque ordinateur connecté à Internet.*
Exemples : 127.0.0.1 et 123.45.67.89

2. Fonctionnement

Voici une ligne du fichier hosts : 127.0.0.1 www.example.org. Cette ligne veut dire que lorsque n'importe quel programme ou outil (votre navigateur, commande ping, et caetera) qui tentera d'accéder à www.example.org sera redirigé à l'adresse du serveur ayant pour IP 127.0.0.1. Simple comme bonjour.

2.1. Généralités

Dans mon navigateur préféré, j'entre l'adresse www.example.org, que se passe t-il ? Et bien le navigateur regarde dans le fichier hosts si www.example.org est attribué à une adresse IP.

Si oui, alors www.example.org vous amènera au serveur qui a l'adresse IP définie telle que dans ce dernier.

Si non, le navigateur interrogera divers serveurs spécialisés (serveurs DNS) afin de « résoudre » l'adresse IP de www.example.org ; ce qu'on peut traduire par « trouver », tout simplement.

Puis le navigateur affiche la page d'accueil de www.example.org.

[Voir 3. Pour aller plus loin pour des détails plus techniques]

2.2. Avantage sur les logiciels tiers

Il existe quelques outils spécialisés qui permettent de bloquer un nom de domaine ou une adresse IP. C'est le cas de beaucoup de bloqueurs de publicités (anti-pub).

L'utilisation d'un fichier hosts modifié est différent dans l'approche mais identique dans le processus. Je m'explique, ces logiciels (ou modules pour votre navigateur) contiennent des bases de données des adresses IP et noms de domaine "mauvais" (serveurs espions, axés sur la pornographie, le piratage (warez), ou tout autres sujets qui ont une réputation dite mauvaise pour les éditeurs de ces logiciels ou modules).

Lorsque ces outils se rendent comptent que votre navigateur fait une requête à un de ces serveurs, ils bloquent leur affichage. Simple, net et sans bavure (ou presque[1]).

L'avantage premier du fichier hosts sur tout ces outils est qu'il y a moins de requêtes ; c'est-à-dire qu'avec un fichier hosts bien configuré, mon navigateur n'essayera même pas d'envoyer une requête au serveur "mauvais", mais plutôt au serveur local (votre machine). Du coup il ne s'agit plus là de bloquer l'affichage de la publicité puisqu'on ne l'a pas du tout téléchargée.

D'autres avantages sous forme de liste :

- aucun besoin de logiciel (ou module) tiers ;
- petit fichier texte -> pas de surcharge de votre machine ;
- facilité de modification ;
- utilisation transparente, ni logo ni icône nulle part ;
- navigation sur Internet nettement plus rapide, saine et agréable ;
- et j'en oublie certainement...

[1] *Même si vous ne voyez pas la publicité, votre navigateur a téléchargé et gardé en mémoire divers scripts et images.*

2.3. Mise à jour

Bien entendu, un fichier hosts dépassé ne sert pas à grand chose. C'est là qu'AdZHosts tombe à pic !

AdZHosts est maintenu par Alden MagiX et n'est ni plus ni moins qu'un ficher hosts optimisé et régulièrement mis à jour. Vous pouvez le trouver sur le site officiel http://www.adzhosts.com et pour toute question/suggestion il y a un forum où vous pourrez poster.

Pour les utilisateurs de GNU/Linux, Unix et certainement Mac, j'ai écrit un script Perl pour automatiser la mise à jour et "purifier" ce que notre ami Alden MagiX nous a concocté (par mesure de sécurité, non pas que je ne lui fasse pas confiance). Ce script se nomme adzhosts.sh (version anglaise) ou adzhosts-fr.sh (pour la version française) et se trouve ici :

http://bobotig.fr/contenu/contrib/scripts/adzhosts.sh
http://bobotig.fr/contenu/contrib/scripts/adzhosts-fr.sh

Il est simple d'utilisation et il explique tout ce qui se passe, rien n'est caché à l'utilisateur.

Bon à savoir, l'emplacement de ce fameux fichier hosts :

```
GNU/Linux et Unix        : /etc/hosts
Mac OS(X)                : /etc/hosts
Windows 95/98            : c:\windows\[1]
Windows NT/2000          : c:\winnt\system32\drivers\etc\
Windows XP               : c:\windows\system32\drivers\etc\
Windows 2008/Vista/Seven : c:\windows\system32\drivers\etc\hosts
OS/2                     : boot\mptn\etc\hosts
BeOS                     : /boot/beos/etc/hosts
```

[1] Le fichier n'existe peut être pas par défaut, un fichier HOST.SAM peut être présent, vous pourrez l'utiliser comme base de départ sans oublier de le renommer HOST sans aucune extension.

3. Pour aller plus loin

Cette partie est plus technique, vous pouvez l'ignorer sans soucis.
Grâce à l'excellent strace[1], voyons comment procède la commande ping[2].

Commande effectuée : ping www.example.org
Réponse :

```
PING www.example.org (192.0.32.10) 56(84) bytes of data.
64 bytes from 192.0.32.10: icmp_req=1 ttl=59 time=22.6 ms
64 bytes from 192.0.32.10: icmp_req=2 ttl=59 time=32.3 ms
64 bytes from 192.0.32.10: icmp_req=3 ttl=59 time=20.7 ms
64 bytes from 192.0.32.10: icmp_req=4 ttl=59 time=21.4 ms
64 bytes from 192.0.32.10: icmp_req=5 ttl=59 time=101 ms
^C64 bytes from 192.0.32.10: icmp_req=6 ttl=59 time=31.9 ms

--- www.example.org ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5005ms
rtt min/avg/max/mdev = 20.784/38.485/101.656/28.646 ms
```

Nous savons que l'adresse IP de www.example.org est 192.0.32.10.
Maintenant voyons ceci plus en détails à l'aide de strace.

Commande effectuée : strace -f -e open ping www.example.org
Réponse :

```
(...)
open("/etc/resolv.conf", O_RDONLY)      = 4
(...)
open("/etc/host.conf", O_RDONLY)        = 4
open("/etc/hosts", O_RDONLY|O_CLOEXEC)  = 4
PING www.example.org (192.0.32.10) 56(84) bytes of data.
open("/etc/hosts", O_RDONLY|O_CLOEXEC)  = 4
64 bytes from www.example.com (192.0.32.10): icmp_req=1 ttl=242 time=183 ms
open("/etc/hosts", O_RDONLY|O_CLOEXEC)  = 4
64 bytes from www.example.com (192.0.32.10): icmp_req=2 ttl=242 time=186 ms
open("/etc/hosts", O_RDONLY|O_CLOEXEC)  = 4
64 bytes from www.example.com (192.0.32.10): icmp_req=3 ttl=242 time=192 ms
open("/etc/hosts", O_RDONLY|O_CLOEXEC)  = 4
64 bytes from www.example.com (192.0.32.10): icmp_req=4 ttl=242 time=186 ms
^C
```

On voit bien que le système va voir ce qui se trouve dans le fichier hosts.

Dans /etc/hosts j'ajoute la ligne : 127.0.0.1 www.example.org
Lorsque je relance la commande sans strace, voici le résultat :

```
PING www.example.org (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_req=1 ttl=64 time=0.015 ms
64 bytes from localhost (127.0.0.1): icmp_req=2 ttl=64 time=0.013 ms
64 bytes from localhost (127.0.0.1): icmp_req=3 ttl=64 time=0.014 ms
64 bytes from localhost (127.0.0.1): icmp_req=4 ttl=64 time=0.015 ms
64 bytes from localhost (127.0.0.1): icmp_req=5 ttl=64 time=0.014 ms
^C
--- www.example.org ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3997ms
rtt min/avg/max/mdev = 0.013/0.014/0.015/0.002 ms
```

Voyez, l'adresse IP de www.example.org est devenue 127.0.0.1.
C'est un exemple parmis tant d'autres, le fonctionnement est identique lorsque vous utilisez votre navigateur.

De part cette manipulation du fichier hosts, on soulève une faille de sécurité : si une personne mal intentionnée a accès à votre fichier hosts et qu'il a les droits suffisants pour le modifier, il pourrait aisément dérouter le site de votre banque ou votre compte courriel afin que www.ma-banque.fr vous amène ailleurs que sur le réel serveur de votre banque ; sur un serveur pirate par exemple...

[1] strace est un outil de déboguage sous Linux pour surveiller les appels système utilisés par un programme.
[2] ping est un outil permettant d'envoyer une requête ICMP 'Echo' d'une machine à une autre

machine.

4. Conclusion

Le fichier hosts est une redoutable arme contre les sites à mauvaise réputation, il est simple et rapide de le modifier et les effets entrent directement en compte. C'est un moyen efficace dont tout le monde devrait avoir connaissance, que vous soyez administrateur système ou simple utilisateur.

Si vous voyez quelque chose à modifier ou des questions/suggestions, vous pouvez nous contacter par courriel à l'adresse suivante : bobotig (@) gmail (.) com.
Ou sur les forums d'AdZHosts : http://adzhosts.free.fr/forum/.
De même, si certains termes sont trop techniques, n'hésitez pas à me le rappeler.

5. Références

Par ordre alphabétique :

http://fr.wikipedia.org/wiki/Domain_Name_Server
http://fr.wikipedia.org/wiki/Hosts
http://fr.wikipedia.org/wiki/Ping_(logiciel)
http://fr.wikipedia.org/wiki/Strace
http://lelogiciellibre.net/tutoriaux/fichier-hosts.php
http://manpagesfr.free.fr/man/man5/hosts.5.html
http://rlwpx.free.fr/WPFF/hosts.htm
http://someonewhocares.org/hosts/
http://www.adzhosts.com
http://www.figlet.org

6. Autres formats

HTML : http://bobotig.fr/contenu/documents/hosts/hosts.html
Texte plain : http://bobotig.fr/contenu/documents/hosts/hosts.txt

7. Historique

[10 octobre 2010]
- correction d'erreurs de grammaire

[29 août 2010]
- modification du site pris pour exemple (www.example.org)
- rectification de l'URL du site d'AdZHosts
- ajout des liens vers les autres formats disponibles
- correction de quelques phôttes daurthogràfe

[28 août 2010]
- version initiale

$ Fin $