# ***Première approche d'un outil merveilleux hping***

Il était l'heure de faire une doc un peu plus sérieuse, alors voilà je me suis lancé tête baissée dedans, donc je pense que ça va être seulement la partie 1, d'abord parce que je vais jamais me rappeller de tout, puis ensuite parce que hping est un outil tellement complet qu'on pourrait en écrire des livres.

Alors déjà hping c'est quoi ? Si on prend un Os au hasard (gentoo, au hasard j'ai dis!) un petit eix hping nous dis : A ping-like TCP/IP packet assembler/analyzer, « mouais ça nous dit pas ce que c'est » me direz vous, et vous auriez raison.
Pour le décrire je dirais que c'est un outils qui permet de manipuler, créer, « forger » des paquets réseaux.
Alors à peu près tous les genres de paquets qui existent, que cela soit TCP, UDP, IP, ICMP, RAW packet, et pleins d'autres, c'est un outil qui peut être utilisé pour tester les règles de firewall sur son réseau, la solidité des services et ou systèmes (en testant une surcharge tcp ou udp par exemple), enfin en fait je crois que hping fait tout sauf le café. Le site officiel : http://www.hping.org/ .

Cet article n'a pas pour but d'être utilisé par les petits scripts kiddies ou lamers qui trainent dans le coin, mais est fait pour les administrateurs réseaux, ou passionnés de l'informatique désireux de sécuriser leur réseau. Je ne pourrais être tenu pour responsable en cas de mauvaise utilisation que ça soit clair.
Une bonne connaissance de quelques protocoles est selon moi indispensable pour la lecture de l'article, bonne lecture à tous.

Le premier pas va être l'installation :

Gentoo : emerge hping
Debian/Ubuntu : aptitude install hping3 (ou le 2 comme vous voulez)
FreeBSD : cd /usr/ports/net/hping/; make install clean
NetBSD : cd /usr/pkgsrc/net/hping/; make install clean
OpenBSD : cd /usr/ports/net/hping/; make install clean

Bon une fois qu'il est installé, on va faire simple je vais stopper iptables sur ma machine pour faciliter les tests et on commence (le ~# étant mon prompt root) :

```
 ~ # hping lamusic
HPING lamusic (eth0 192.168.1.13): NO FLAGS are set, 40 headers + 0 data bytes
len=40 ip=192.168.1.13 ttl=99 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=0.2 ms
len=40 ip=192.168.1.13 ttl=99 DF id=0 sport=0 flags=RA seq=1 win=0 rtt=0.2 ms

--- lamusic hping statistic ---
2 packets tramitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.2 ms
 ~ #
```
(ctrl+c sinon ça boucle)

c'est un hping par défaut sans aucun argument mis à part le host, un paquet TCP est émis par défaut sur le port 0 avec aucun drapeau (c'est suffisament écrit en gros je crois), comme aucun serveur n'écoute, que ma pile TCP/IP est plus ou moins intelligente et bien on récolte quelques informations que nous allons détailler; on voit clairement :

len=40 ip=192.168.1.13 ttl=99 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=0.2 ms

ip je pense que ce n'est pas la peine de dire à quoi ça correspond tout les gens censés lire doivent le savoir. (mais c'est celle de la cible bien évidemment).

Ttl c'est le time to live, durée de vie d'un paquet, cela peut être très utile vu qu'il se décrémente chaque fois qu'il passe un saut (hop) c'est la méthode qu'utilise le très célèbre outil traceroute.

Flags ce sont les drapeaux tcp ici reset/ack.

Win c'est la window size, taille de fenêtre TCP.

Id c'est le ip id qui permet notamment d'avoir certaines informations marrantes sur les systèmes windows (et autres), mais ça ne sera pas traité dans cet article.

Pour la suite, vu que ce n'est pas un cours sur les compositions de datagrammes, faudra se documenter si vous ne connaissez pas les autres champs et leur utilités, par exemple ici :
http://frameip.com/

Un petit coup dans le help et dans le man et magie, on peut choisir pleins de trucs!

```
~ # hping -S lamusic -p 21 -V
using eth0, addr: 192.168.1.14, MTU: 1500
HPING lamusic (eth0 192.168.1.13): S set, 40 headers + 0 data bytes
len=40 ip=192.168.1.13 ttl=99 DF id=0 tos=0 iplen=40
sport=21 flags=RA seq=0 win=0 rtt=0.2 ms
seq=0 ack=306231913 sum=7e57 urp=0

len=40 ip=192.168.1.13 ttl=99 DF id=0 tos=0 iplen=40
sport=21 flags=RA seq=1 win=0 rtt=0.2 ms
seq=0 ack=638404457 sum=b0b6 urp=0


--- lamusic hping statistic ---
2 packets tramitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.2 ms
~ #
```

On va décrire un peu ce que j'ai fait, alors j'ai rajouté un -S « mais c'est quoi » c'est simple c'est un paquet TCP avec le drapeau Syn, qui est la demande de connection initiale, et forcément comme aucun server n'est là (win=0 qui nous le confirme) on se prend un R/A logique (reset/ack).
On a un peu plus de détails avec les numéros de ack et de séquences, qui permettent la « fiabilité » de la communication sur TCP/IP.

Bon ensuite il serait bon de changer un peu de protocole voir si ça répond.

```
~ # hping -1 lamusic -p 21 -V
using eth0, addr: 192.168.1.14, MTU: 1500
HPING lamusic (eth0 192.168.1.13): icmp mode set, 28 headers + 0 data bytes

--- lamusic hping statistic ---
2 packets tramitted, 0 packets received, 100% packet loss
```

round-trip min/avg/max = 0.0/0.0/0.0 ms

(ah oui mon kernel est réglé pour ne pas répondre, donc meme sans fw,  tant pis vous testerez chez vous le proto ICMP)

```
~ # hping -2 lamusic -p 21 -V
using eth0, addr: 192.168.1.14, MTU: 1500
HPING lamusic (eth0 192.168.1.13): udp mode set, 28 headers + 0 data bytes
ICMP Port Unreachable from ip=192.168.1.13 name=lamusic.lamusic.dyndns.info
ICMP Port Unreachable from ip=192.168.1.13 name=lamusic.lamusic.dyndns.info
ICMP Port Unreachable from ip=192.168.1.13 name=lamusic.lamusic.dyndns.info

--- lamusic hping statistic ---
3 packets tramitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Normal c'est UDP, toutes façons pour avoir de vraies réponses UDP vu qu'il fonctionne en mode non connecté, le plus simple est d'envoyer des requètes sur la couche application. (Ce qui encore une fois n'est pas le but de cet article donc passons ...)

On va seulement s'interresser à TCP/IP dans cet article c'est assez large et déjà pas mal pour une première approche de ce fabuleux outil.

Aller je suis gentil je vais vous montrer comment scanner vite fait avec, bien qu'il soit pas vraiment un scanneur.

A vos shell :

```
~ # hping -S lamusic -8 33000-33355
Scanning lamusic (192.168.1.13), port 33000-33355
356 ports to scan, use -V to see all the replies
+----+-----------+---------+---+-----+-----+-----+
|port| serv name |  flags  |ttl| id  | win | len |
+----+-----------+---------+---+-----+-----+-----+
33333            : .S..A...  99     0  9999    44
All replies received. Done.
Not responding ports:
~ #
```

Waou on a trouvé un port ouvert, sur un service mais qu'est ce que donc, on est pas là pour la découverte des services, chose qui est le plus souvent assez simple finalement, mais pour la détection des systèmes.

Nous voyons le champ ttl avec une valeur de 99 et une winsize avec une valeur 9999, et le port (33333), c'est un comique celui là.

Voyons de plus près :

```
 ~ # hping -S lamusic -p 33333 -V -c 1
using eth0, addr: 192.168.1.14, MTU: 1500
HPING lamusic (eth0 192.168.1.13): S set, 40 headers + 0 data bytes
len=44 ip=192.168.1.13 ttl=99 DF id=0 tos=0 iplen=44
```

sport=33333 flags=SA seq=0 win=9999 rtt=0.2 ms
seq=4028957570 ack=1627448687 sum=f23 urp=0

--- lamusic hping statistic ---
1 packets tramitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.2 ms
~ #

Ah alors là on n'a pas la même chose que pour le premier stimulus ( hping -S lamusic -p 21 -V), nous avons un numéro de winsize ce qui indique qu'un service est en écoute sur ce port, et nous avons en réponse un S/A (syn/ack) ce qui confirme bien la présence d'une socket en écoute.

Alors une winsize de ce type, et un ttl de ce type font partie d'aucun Os connu, le premier réflexe serait de dire que c'est une *Bsd avec un sysctl optimisé, mais en regardant simplement le id=0 ça ne tient pas, toutes les *Bsd ont un id non nul et randomizable simplement de surcroit. Par contre ce n'est pas le cas d'un kernel linux qui joue avec son id seulement une fois la connexion établie; c'est une autre méthode mais qui n'est pas si mal finalement.

Pour vous montrer la diffèrence voici un kernel linux 2.6 par défaut :

~ & hping -S xenhardened -p 22 -V -c 1
using eth0, addr: 192.168.1.13, MTU: 3373
HPING xenhardened (eth0 192.168.1.14): S set, 40 headers + 0 data bytes
len=46 ip=192.168.1.14 ttl=64 DF id=0 tos=0 iplen=44
sport=22 flags=SA seq=0 win=5840 rtt=1.5 ms
seq=2603312411 ack=1446574584 sum=9ea1 urp=0

--- xenhardened hping statistic ---
1 packets tramitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 1.5/1.5/1.5 ms
~ &

Là il n'y a aucun doute possible un seul paquet suffit, pas besoin de sniffer le retour ou netcat le service, on sait que c'est un linux, c'est marqué dessus, le ttl 64 la winsize à 5840 est une marque de fabrique ( y a de forte chance que ça soit openssh en plus là) enfin une petite astuce, reste le ttl par défaut qui diffère selon les linux(quand les dev le modifie), mais ce n'est pas une valeur sure, pour connaître la distribution Linux le plus souvent il faut aller jusqu'à la découverte des services, et donc rentrer en mode connecté.

~ # hping -S xx.xxx.xx.xxx -c 1 -V -p 3389
using eth0, addr: 192.168.1.14, MTU: 1500
HPING xx.xxx.xx.xxx (eth0 xx.xxx.xx.xxx): S set, 40 headers + 0 data bytes
len=44 ip=xx.xxx.xx.xxx ttl=119 DF id=16290 tos=0 iplen=44
sport=3389 flags=SA seq=0 win=65535 rtt=413.6 ms
seq=2588572843 ack=751287042 sum=b052 urp=0

--- xx.xxx.xx.xxx hping statistic ---
1 packets tramitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 413.6/413.6/413.6 ms

~ #
(merci à mon ami Epx pour son ip et son port forward que je masque bien entendu)

Alors là le premier réflexe serait de dire un windows  xp sp2!
Pourquoi ? Parce que!

La marque de fabrique windows :
window size sur 65535 ttl au dessus de 100 souvent 128 voir 126 124, 116 enfin ça peut varier selon les versions mais sinon toujours au dessus de 100 et un id non nul.
Mais freebsd a une winsize = 65535 par defaut aussi, alors pour être sûr qu'Epx ne ce soit pas mis à freebsd va falloir que je regarde d'autre chose, bon le ttl freebsd par defaut n'est pas là, 64 par défaut, donc déjà on a une petite piste.

Bon ce qui va être la clef de notre dilemme va etre le id.

On envoit alors 5 paquets pour regarder l'incrément id :

```
~ # hping -S xx.xxx.xx.xxx -c 5 -p 3389 -r
HPING xx.xxx.xx.xxx (eth0 xx.xxx.xx.xxx): S set, 40 headers + 0 data bytes
len=44 ip=xx.xxx.xx.xxx ttl=119 DF id=16318 sport=3389 flags=SA seq=0 win=65535 rtt=237.5 ms
len=44 ip=xx.xxx.xx.xxx ttl=119 DF id=+1 sport=3389 flags=SA seq=1 win=65535 rtt=269.4 ms
len=44 ip=xx.xxx.xx.xxx ttl=119 DF id=+1 sport=3389 flags=SA seq=2 win=65535 rtt=315.5 ms
len=44 ip=xx.xxx.xx.xxx ttl=119 DF id=+1 sport=3389 flags=SA seq=3 win=65535 rtt=278.5 ms
len=44 ip=xx.xxx.xx.xxx ttl=119 DF id=+1 sport=3389 flags=SA seq=4 win=65535 rtt=324.3 ms

--- xx.xxx.xx.xxx hping statistic ---
5 packets tramitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 237.5/285.1/324.3 ms
 ~ #
```

On voit le calme plat, du coup ça donne 2 indications, là on est convaincu que c'est un windows, et que le service idle c'est à dire qu'il n'y a aucune activité dessus.

La détection de système à distance demande beaucoup de temps pour étudier les multitudes de configurations possibles des diffèrentes piles TCP/IP, donc voyons encore un peu plus d'exemple.

(merci à mon ami geekounet et son server pour le pret de son port ssl le temps de quelques stimuli )
https://**akoya**.homeunix.org/ tant qu'à faire allez y faire un tour :) )

```
~ # hping -S akoya.homeunix.org -c 1 -p 443 -V
using eth0, addr: 192.168.1.14, MTU: 1500
HPING akoya.homeunix.org (eth0 62.147.212.31): S set, 40 headers + 0 data bytes
len=44 ip=62.147.212.31 ttl=54 DF id=53993 tos=0 iplen=44
sport=443 flags=SA seq=0 win=65535 rtt=99.4 ms
seq=1084045391 ack=2059339506 sum=d6bc urp=0


--- akoya.homeunix.org hping statistic ---
1 packets tramitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 99.4/99.4/99.4 ms
~ #
```

Alors qu'est ce qu'on a de beau la dedans, (comme je le connais je le vois mal avec un windows, mais sait on jamais on va vérifier pour etre sûr) oui ça ressemble étrangement au paquet de toute à l'heure et oui!

Mis à part le service par rapport au premier paquet sur le serveur d' Epx ça se ressemble beaucoup, le ttl diffère lui aussi, ce qui me fait penser au premier abord à un système non windows, mais plus un cousin unix comme linux ou *Bsd, et la winsize elle me fait penser à une winsize par défaut sur freebsd, mais on est pas sur du tout, vu qu'on aurait pu changer le ttl dans le kernel windows pour nous jouer des tours, malheureusement un seul paquet ne suffira pas là pour connaître le système. Ce n'est pas grave, on va envoyer d'autre, (j'en profite pendant qu'il regarde pas son tcpdump).

Aller une petite volée de 5 stimuli comme pour Epx mais on va changer le temps entre les paquets pour ne pas l'alerter, tant qu'à faire on ne va pas le déranger pour rien.

```
~ # hping -S akoya.homeunix.org -c 5 -p 443 -i 2 -r
HPING akoya.homeunix.org (eth0 62.147.212.31): S set, 40 headers + 0 data bytes
len=44 ip=62.147.212.31 ttl=54 DF id=30684 sport=443 flags=SA seq=0 win=65535 rtt=99.1 ms
len=44 ip=62.147.212.31 ttl=54 DF id=+17419 sport=443 flags=SA seq=1 win=65535 rtt=98.2 ms
DUP! len=44 ip=62.147.212.31 ttl=54 DF id=62365 sport=443 flags=SA seq=0 win=65535
rtt=3097.8 ms
len=44 ip=62.147.212.31 ttl=54 DF id=+47083 sport=443 flags=SA seq=2 win=65535 rtt=98.3 ms
DUP! len=44 ip=62.147.212.31 ttl=54 DF id=13054 sport=443 flags=SA seq=1 win=65535
rtt=3096.9 ms

--- akoya.homeunix.org hping statistic ---
3 packets tramitted, 5 packets received, -66% packet loss
2 out of sequence packets received
round-trip min/avg/max = 98.2/1298.1/3097.8 ms
~ #
```

Alors là on voit la magie du kernel freebsd avec le TCP id randomize d'activer(je dirais même en kernel, celui de pf ne réagit pas de la même façon du moins sur freebsd), le id est carrément imprévisible et la pile nous envoie chier(enfin pf je pense plutot) de temps en temps, bien que ça soit parfaitement sécurisé la détection du système reste possible bien-sûr.

Voyons ce que cela donne avec ma netbsd qui idle :

```
~ # hping -S netbsd -c 5 -p 22 -i 2 -r
HPING netbsd (eth0 192.168.1.16): S set, 40 headers + 0 data bytes
len=44 ip=192.168.1.16 ttl=64 DF id=58107 sport=22 flags=SA seq=0 win=32768 rtt=0.6 ms
len=44 ip=192.168.1.16 ttl=64 DF id=+1 sport=22 flags=SA seq=1 win=32768 rtt=0.2 ms
len=44 ip=192.168.1.16 ttl=64 DF id=+1 sport=22 flags=SA seq=2 win=32768 rtt=0.2 ms
len=44 ip=192.168.1.16 ttl=64 DF id=+1 sport=22 flags=SA seq=3 win=32768 rtt=0.2 ms
len=44 ip=192.168.1.16 ttl=64 DF id=+1 sport=22 flags=SA seq=4 win=32768 rtt=0.2 ms

--- netbsd hping statistic ---
5 packets tramitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.3/0.6 ms
~ #
```

Nous avons notre id qui est le même qu'un windows qui idle heurement le champ ttl et winsize sont là

pour nous indiquer le système. Je vous rassure freebsd fait pareil par défaut sans le random actif. (chose rare sur les *Bsd en général ce sont des utilisateurs avertis qui activent cette option mais on ne peut pas faire de règle).

Ma netbsd avec deux champs modifiés :

```
NetBsd# /sbin/sysctl -w net.inet.tcp.sendspace=65535
net.inet.tcp.sendspace: 32768 -> 65535
NetBsd# /sbin/sysctl -w net.inet.tcp.recvspace=65535
net.inet.tcp.recvspace: 32768 -> 65535
NetBsd# /sbin/sysctl -w net.inet.ip.ttl=128
net.inet.ip.ttl: 64 -> 128
NetBsd#
NetBsd# /etc/rc.d/sshd forcerestart
Stopping sshd.
Waiting for PIDS: 361.
Starting sshd.
NetBsd#
```

Et là on va voir tout simplement la pile TCP/IP d'un windows xp :

```
~ # hping -S netbsd -c 5 -p 22 -i 2 -r
HPING netbsd (eth0 192.168.1.16): S set, 40 headers + 0 data bytes
len=44 ip=192.168.1.16 ttl=128 DF id=58233 sport=22 flags=SA seq=0 win=65535 rtt=0.3 ms
len=44 ip=192.168.1.16 ttl=128 DF id=+1 sport=22 flags=SA seq=1 win=65535 rtt=0.2 ms
len=44 ip=192.168.1.16 ttl=128 DF id=+1 sport=22 flags=SA seq=2 win=65535 rtt=0.2 ms
len=44 ip=192.168.1.16 ttl=128 DF id=+1 sport=22 flags=SA seq=3 win=65535 rtt=0.2 ms
len=44 ip=192.168.1.16 ttl=128 DF id=+1 sport=22 flags=SA seq=4 win=65535 rtt=0.2 ms

--- netbsd hping statistic ---
5 packets tramitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.3 ms
~ #
```

Puis la je vais faire croire à une freebsd :

```
~ # hping -S netbsd -c 5 -p 22 -i 2 -r
HPING netbsd (eth0 192.168.1.16): S set, 40 headers + 0 data bytes
len=44 ip=192.168.1.16 ttl=64 DF id=32805 sport=22 flags=SA seq=0 win=65535 rtt=0.3 ms
len=44 ip=192.168.1.16 ttl=64 DF id=+34860 sport=22 flags=SA seq=1 win=65535 rtt=0.2 ms
len=44 ip=192.168.1.16 ttl=64 DF id=+28408 sport=22 flags=SA seq=2 win=65535 rtt=0.2 ms
len=44 ip=192.168.1.16 ttl=64 DF id=+25880 sport=22 flags=SA seq=3 win=65535 rtt=0.2 ms
len=44 ip=192.168.1.16 ttl=64 DF id=+60864 sport=22 flags=SA seq=4 win=65535 rtt=0.2 ms

--- netbsd hping statistic ---
5 packets tramitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.3 ms
~ #
```

Voilà nous voyons les détails d'une pile TCP/IP identique à celle de la freebsd stimulée plus haut sur le serveur de geekounet, pour démontrer qu'il est parfaitement possible de faire passer un système pour ce qu'il n'est pas en réalité.

La détection de système est une chose qui demande beaucoup d'apprentissage, de tests et de nombreuses heures de lectures, cette doc n'est absolument pas complète et je ferai surement une seconde partie beaucoup plus poussée.

Le mot de la fin, on peut rarement être sûr et certain que l'os est le bon avec des tests de base, bien que bons nombres d'indications peuvent nous faire penser qu'il s'agit de tel ou tel système, il suffit souvent de se documenter sur son système préféré pour voir comment modifier tout cela, enfin en général sysctl est ton ami.
Pour vraiment être sûr du système en général il faudra pousser la détection plus loin, sniffing avec tcpdump par exemple (pour récolter encore plus d'info), et peut-être même aller jusqu' à la couche application.

Hping est beaucoup plus pointu que cela, ce document était simplement l'utilisation de base pour une détection de base. Le mode avancé sera vue dans la partie 2 de mon article que je ferai d'ici peu.

Le vendredi 14 décembre 2007, rédigé par lamusic.