

Installation et configuration d'un honeypot : Honeyd.

par Goldkey ([Accueil](#))

Date de publication : 02 juin 2008

Dernière mise à jour : 14 juillet 2008

Ce tutoriel a pour but d'expliquer l'installation d'un honeypot sur une distribution Debian.

I - Prérequis.....	3
II - Qu'est ce qu'un honeypot ?.....	3
III - Installation.....	3
IV - Configuration.....	4
IV-A - Fichier de configuration utilisé.....	4
IV-B - Mise en place.....	5
V - L'exécution.....	5
VI - Les scripts d'émulation de services.....	6
VII - Les premiers pas.....	6
VII-A - Vérification avec une commande ping.....	6
VII-B - Vérification avec l'utilisation d'un script.....	7
VIII - Pour aller plus loin.....	8
IX - Conclusion.....	9

I - Prérequis

Le contenu de cet article a été rédigé en se basant sur une distribution Debian Etch. Les interfaces réseau utilisées seront lo (127.0.0.1) et ath0 (192.168.0.248). Suivant votre configuration n'oubliez pas de les adapter au besoin.

II - Qu'est ce qu'un honeypot ?

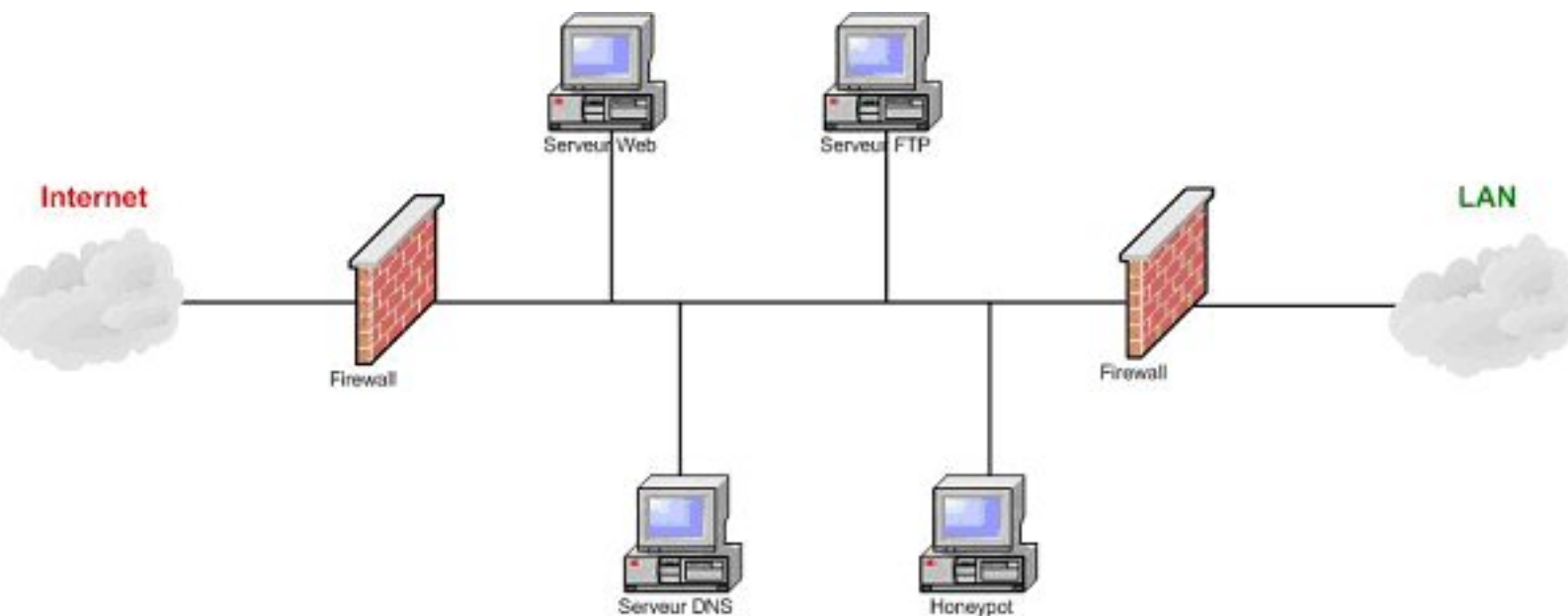
Un honeypot permet d'émuler des services sur une machine afin de simuler le véritable fonctionnement d'une machine de production. Ce système assure ainsi la surveillance du réseau par la collecte et le traitement des informations.

Les honeypots sont principalement divisés en deux catégories : les honeypots à faible interaction et les honeypots à forte interaction.

Les honeypots à faible interaction ne fournissent pas de véritables services, ils se contentent de les simuler par l'intermédiaire de script, comme Honeyd le propose. Ces honeypots ne posent que très peu de problèmes de sécurité.

A contrario, les honeypots à forte interaction fournissent des services bien réels mais non dédiés à la production. Ceux-ci sont très sensibles et pourraient mettre à mal la sécurité de votre entreprise. Il convient donc d'apporter une attention particulière à leur sécurisation.

Ci-dessous, un exemple d'utilisation d'un honeypot :



III - Installation

Dans cet article nous avons choisis d'utiliser un honeypot à faible interaction : **Honeyd**.

La version d'Honeyd actuelle (Janvier 2008) fournie en paquet pour Debian Etch est la 1.5b. Pour installer Honeyd, nous allons utiliser Aptitude, en tapant la commande ci-dessous :

```
apt-get install honeyd
```

Les principaux fichiers installés sont les suivants :

```
/etc/init.d/honeyd
/etc/logrotate.d/honeyd
/etc/default/honeyd
/usr/lib/honeyd
/usr/share/honeyd
/usr/share/doc/honeyd
/usr/include/honeyd
/usr/bin/honeyd
```

Suivant l'utilisation de votre honeypot vous pourriez avoir besoin des paquets suivant :

- farp
- rrdtool

Pour plus d'informations sur les dépendances du paquet Honeyd vous pouvez visiter le site Web de Debian : <http://packages.debian.org/etch/honeyd>

IV - Configuration

IV-A - Fichier de configuration utilisé

Le fichier de configuration utilisé est similaire au fichier installé par défaut.

```
#####
#Configuration du réseau virtuel utilisé
route entry 10.0.0.1
route 10.0.0.1 link 10.2.0.0/24
route 10.0.0.1 add net 10.3.0.0/16 10.3.0.1 latency 8ms bandwidth 10Mbps
route 10.3.0.1 link 10.3.0.0/24
route 10.3.0.1 add net 10.3.1.0/24 10.3.1.1 latency 7ms loss 0.5
route 10.3.1.1 link 10.3.1.0/24

#####
# Création d'un profil template
create template
set template personality "Microsoft Windows XP Professional SP1"
set template uptime 1728650
set template maxfds 35
# For a complex IIS server
add template tcp port 80 "sh /usr/share/honeyd/scripts/win32/web.sh"
add template tcp port 22 "/usr/share/honeyd/scripts/test.sh $ipsrc $dport"
add template tcp port 23 proxy $ipsrc:23
add template udp port 53 proxy 141.211.92.141:53
set template default tcp action reset
# Use this if you are not running honeyd as 'honeyd' user:
# Debian-specific (use nobody = 65534 instead of 32767)
# set template uid 65534 gid 65534

#####
#Création d'un profil default
create default
set default default tcp action block
set default default udp action block
set default default icmp action block

#####
#Création d'un profil router
create router
set router personality "Cisco 1601R router running IOS 12.1(5)"
set router default tcp action reset
add router tcp port 22 "/usr/share/honeyd/scripts/test.sh"
#add router tcp port 23 "/usr/share/honeyd/scripts/router-telnet.pl"
```

```
add router tcp port 23 "/usr/share/honeyd/scripts/telnet/fake.pl"

#####
#On démarre les hotes en spécifiant l'IP et le profil
bind 10.3.0.1 router
bind 10.3.1.1 router
bind 10.3.1.12 template
bind 10.3.1.11 template
bind 10.3.1.10 template
set 10.3.1.11 personality "Microsoft Windows NT 4.0 SP3"
set 10.3.1.10 personality "IBM AIX 4.2"
```

IV-B - Mise en place

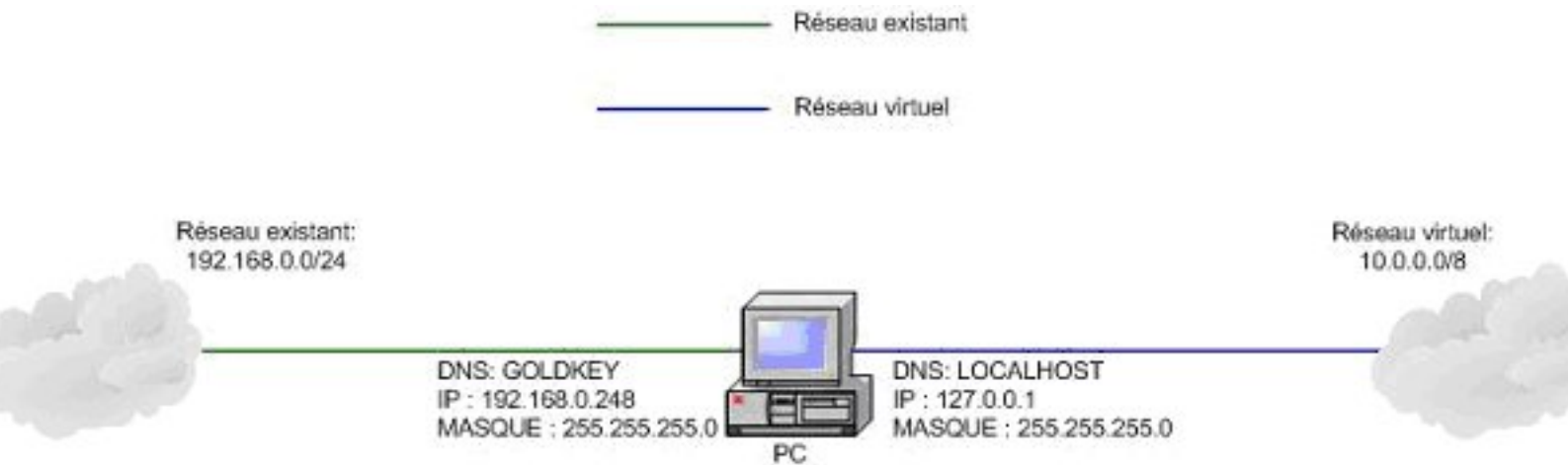
Nous allons utiliser la configuration par défaut proposer par Honeyd dans le fichier honeyd.conf:

```
route entry 10.0.0.1
route 10.0.0.1 link 10.2.0.0/24
route 10.0.0.1 add net 10.3.0.0/16 10.3.0.1 latency 8ms bandwidth 10Mbps
route 10.3.0.1 link 10.3.0.0/24
route 10.3.0.1 add net 10.3.1.0/24 10.3.1.1 latency 7ms loss 0.5
route 10.3.1.1 link 10.3.1.0/24
```

Pour faire fonctionner le réseau virtuel ci-dessus nous allons devoir déclarer une route (bien réelle) dans la table de routage pour l'atteindre. La passerelle utilisée pour cette route sera l'interface loopback (localhost) afin de ne pas perturber le réseau existant.

```
route add -net 10.0.0.0 netmask 255.0.0.0 gw localhost
```

Ci-dessous, un schéma présentant la configuration:



V - L'exécution

Pour tester notre configuration nous allons d'abord lancer Honeyd en mode interactif en lançant la commande ci-dessous dans une console :

```
honeyd -d -p /etc/honeypot/nmap.prints -l /var/log/honeypot/honeyd.log -f /etc/honeypot/honeyd.conf -i lo 10.0.0.0/8
```

Détails des paramètres :

- -d lancer en mode interactif

- -p fichier des fingerprints
- -f fichier de configuration

```
gold@deb ~# honeyd -d -p /etc/honeypot/nmap.prints -l /var/log/honeypot/honeyd.log -f /etc/honeypot/honeyd.conf -i lo 10.0.0.0/8
Honeyd V1.5b Copyright (c) 2002-2004 Niels Provos
honeyd[3676]: started with -d -p /etc/honeypot/nmap.prints -l /var/log/honeypot/honeyd.log -f /etc/honeypot/honeyd.conf -i lo 10.0.0.0/8
honeyd[3676]: listening on lo: ip and (dst net 10.0.0.0/8)
```

Ok tout fonctionne...un petit ctrl+C pour arrêter la commande. Nous allons maintenant lancer notre démon Honeyd. Pour cela nous allons modifier le fichier de configuration du démon. Editer le fichier :

```
/etc/default/honeyd
```

Dans un premier temps, il faut modifier la constante RUN afin de pouvoir démarrer le démon :

```
RUN="yes"
```

Puis indiquer l'interface à utiliser et la plage d'adresses IP du réseau :

```
INTERFACE="ath0"
NETWORK=10.0.0.0/8
```

Puis démarrer le démon Honeyd avec la commande

```
/etc/init.d/honeyd start
```

S'il n'y a pas d'erreur vous devez obtenir :

```
Starting Honeyd daemon: honeyd.
```

VI - Les scripts d'émulation de services

Pour émuler un service fonctionnant sur une machine virtuelle, Honeyd permet l'utilisation de scripts. Ceux-ci peuvent être écrits en langage Perl ou même directement en SHELL. Des exemples de scripts sont fournis avec l'installation d'Honeyd. Les différents scripts sont situés dans le répertoire :

```
/usr/share/honeyd/script
```

Pour obtenir d'autres scripts vous pouvez visiter la rubrique "contributions" sur le site Web d'Honeyd : <http://www.honeyd.org/contrib.php>

Cela vous donnera sûrement de bonnes idées.

VII - Les premiers pas

VII-A - Vérification avec une commande ping

Vérifions si un de nos hôtes configurés répond à une commande ping. Pour plus de visibilité, nous allons lancer Honeyd en mode interactif :

```
gold@deb ~# honeyd -d -p /etc/honeypot/nmap.prints -l /var/log/honeypot/honeyd.log -f /etc/honeypot/honeyd.conf -i lo 10.0.0.0/8
Honeyd V1.5b Copyright (c) 2002-2004 Niels Provos
```

```
honeyd[3753]: started with -d -p /etc/honeypot/nmap.prints -l /var/log/honeypot/honeyd.log -f /etc/honeypot/honeyd.conf -i lo 10.0.0.0/8
honeyd[3753]: listening on lo: ip and (dst net 10.0.0.0/8)
```

Dans une autre console nous allons tenter de pinguer un hôte configuré :

```
gold@deb ~# ping 10.3.0.1
PING 10.3.0.1 (10.3.0.1) 56(84) bytes of data.
64 bytes from 10.3.0.1: icmp_seq=1 ttl=63 time=10.0 ms
64 bytes from 10.3.0.1: icmp_seq=2 ttl=63 time=10.0 ms
64 bytes from 10.3.0.1: icmp_seq=3 ttl=63 time=10.0 ms
64 bytes from 10.3.0.1: icmp_seq=4 ttl=63 time=10.0 ms
64 bytes from 10.3.0.1: icmp_seq=5 ttl=63 time=10.0 ms
64 bytes from 10.3.0.1: icmp_seq=6 ttl=63 time=10.0 ms
64 bytes from 10.3.0.1: icmp_seq=7 ttl=63 time=10.0 ms
```

Honeyd a bien reçu notre ping :

```
gold@deb ~# honeyd -d -p /etc/honeypot/nmap.prints -l /var/log/honeypot/honeyd.log -f /etc/honeypot/honeyd.conf -i lo 10.0.0.0/8
Honeyd V1.5b Copyright (c) 2002-2004 Niels Provos
honeyd[3753]: started with -d -p /etc/honeypot/nmap.prints -l /var/log/honeypot/honeyd.log -f /etc/honeypot/honeyd.conf -i lo 10.0.0.0/8
honeyd[3753]: listening on lo: ip and (dst net 10.0.0.0/8)
honeyd[3753]: Sending ICMP Echo Reply: 10.3.0.1 -> 192.168.0.249
honeyd[3753]: Sending ICMP Echo Reply: 10.3.0.1 -> 192.168.0.249
honeyd[3753]: Sending ICMP Echo Reply: 10.3.0.1 -> 192.168.0.249
honeyd[3753]: Sending ICMP Echo Reply: 10.3.0.1 -> 192.168.0.249
honeyd[3753]: Sending ICMP Echo Reply: 10.3.0.1 -> 192.168.0.249
honeyd[3753]: Sending ICMP Echo Reply: 10.3.0.1 -> 192.168.0.249
honeyd[3753]: Sending ICMP Echo Reply: 10.3.0.1 -> 192.168.0.249
```

VII-B - Vérification avec l'utilisation d'un script

Vérifions si un de nos hôtes configurés répond à l'appel d'un script. Pour plus de visibilité, nous allons à nouveau lancer Honeyd en mode interactif :

```
gold@deb ~# honeyd -d -p /etc/honeypot/nmap.prints -l /var/log/honeypot/honeyd.log -f /etc/honeypot/honeyd.conf -i lo 10.0.0.0/8
Honeyd V1.5b Copyright (c) 2002-2004 Niels Provos
honeyd[3753]: started with -d -p /etc/honeypot/nmap.prints -l /var/log/honeypot/honeyd.log -f /etc/honeypot/honeyd.conf -i lo 10.0.0.0/8
honeyd[3753]: listening on lo: ip and (dst net 10.0.0.0/8)
```

Dans une autre console nous allons tenter d'accéder à un hôte configuré sur le port 23 :

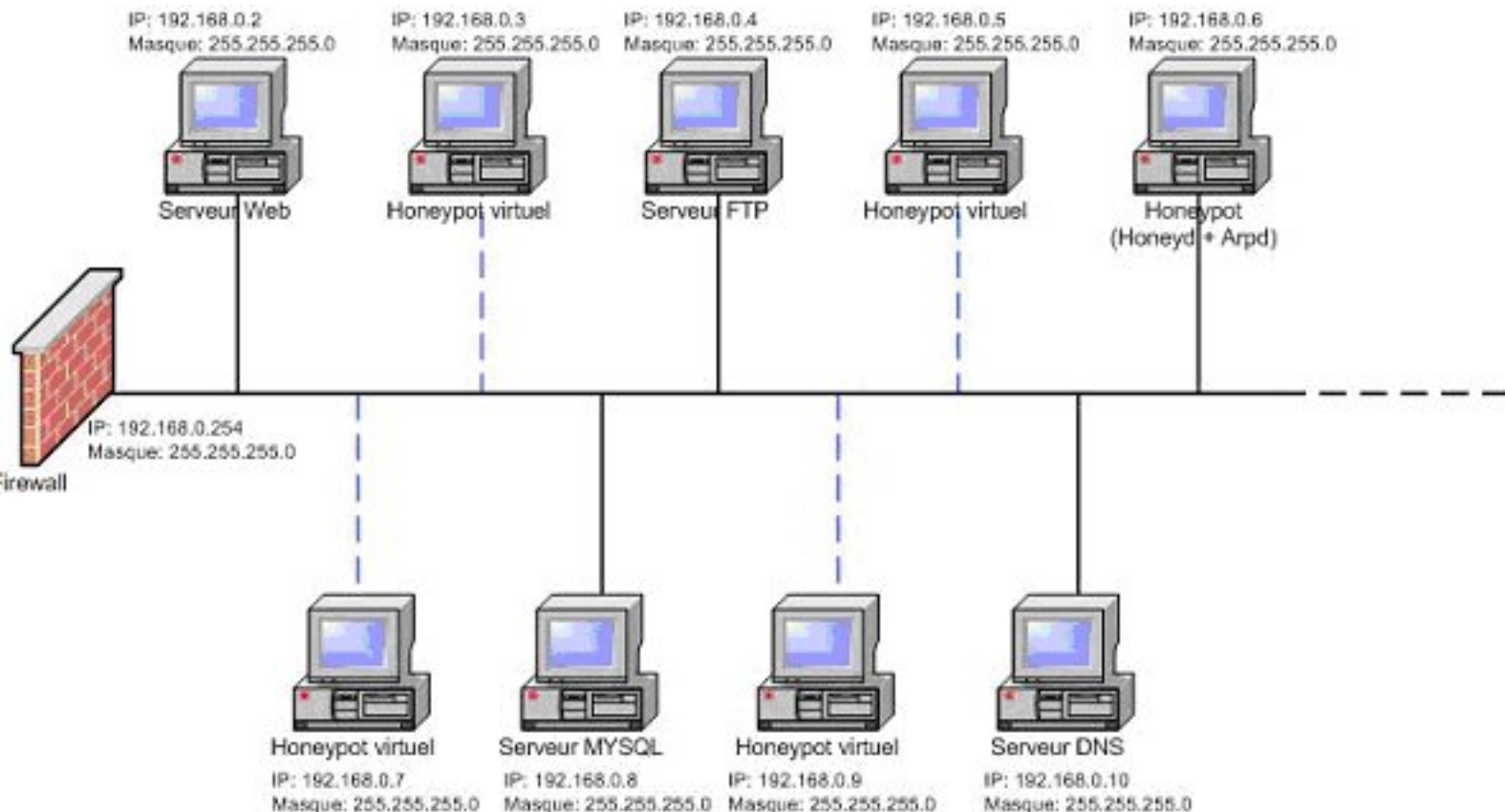
```
gold@deb ~# telnet 10.3.0.1 23
Trying 10.3.0.1...
Connected to 10.3.0.1.
```

Honeyd a bien reçu notre tentative d'accès au port 23 :

```
gold@deb ~# honeyd -d -p /etc/honeypot/nmap.prints -l /var/log/honeypot/honeyd.log -f /etc/honeypot/honeyd.conf -i lo 10.0.0.0/8
Honeyd V1.5b Copyright (c) 2002-2004 Niels Provos
honeyd[3753]: started with -d -p /etc/honeypot/nmap.prints -l /var/log/honeypot/honeyd.log -f /etc/honeypot/honeyd.conf -i lo 10.0.0.0/8
honeyd[3753]: listening on lo: ip and (dst net 10.0.0.0/8)
honeyd[3753]: Sending ICMP Echo Reply: 10.3.0.1 -> 192.168.0.249
honeyd[3753]: Connection request: tcp (192.168.0.249:1584 - 10.3.0.1:23)
honeyd[3753]: Connection established: tcp (192.168.0.249:1584 - 10.3.0.1:23) <-> /usr/share/honeyd/scripts/telnet/fake.pl
```

VIII - Pour aller plus loin

La configuration jusqu'ici a été conçue pour ne pas interagir avec le réseau existant. D'où l'utilisation de l'interface loopback (localhost) pour le routage vers notre réseau virtuel. Cependant grâce au démon Arpd vous pouvez interagir avec votre réseau existant. En effet celui-ci permet d'écouter les requêtes ARP et d'y répondre afin de simuler des machines inexistantes sur le réseau.



⚠ Attention cependant à son utilisation dans un réseau utilisant le protocole DHCP. Il est possible qu'Arpd interfère avec le serveur DHCP en permettant une réponse d'Honeyd au ping envoyé par le serveur DHCP pour déterminer si une adresse IP est libre sur le réseau.

Pour installer Arpd si cela n'a pas été fait au chapitre 1) taper la commande

```
apt-get install farpd
```

Le fichier de configuration du démon Arpd se situe dans le répertoire :

```
/etc/default/farpd
```

N'oubliez pas de modifier les constantes suivantes :

```
INTERFACE="ath0"
NETWORK="192.168.0.0/24"
```

IX - Conclusion

Honeyd est un honeypot à faible interaction complet et particulièrement souple grâce a son système de script. Attention tout de même car Honeyd n'a pas été conçu pour fonctionner dans un environnement de production mais plutôt dans un domaine de recherche afin d'améliorer la sécurité d'un réseau.