S E C U R I N E T S

Club de la sécurité informatique

INSAT

Atelier Honeypot WiFi

Outils:

- Kojoney
- Honeyd

S E C U R I N E T S
Club de la sécurité informatique
I N S A T
www.securinets.com
Tel : 20322191

Atelier Honeypot WiFi

Kojoney

Un Honeypot pour le service SSH

Kojoney

Kojoney est un honeypot pour le service SSH. Avant d'aller plus loin, une terminologie s'impose :

- Un **honeypot** (en français *pot de miel*) est un ordinateur ou un programme volontairement vulnérable destiné à attirer et à piéger les pirates informatiques. Situé devant ou derrière un pare-feu, cet appât laisse croire aux intrus qu'ils se trouvent sur une machine de production « normale » alors qu'ils évoluent sur un leurre. On aura alors tout loisir d'observer leur manière de faire et d'enregistrer leurs méthodes d'attaques
- Le service SSH est un ensemble d'outils, permettant de remplacer les services de login distant (comme Telnet, rlogin, rsh) de manière plus sécurisée, via une procédure d'authentification forte (basée sur des algorithmes de cryptographie) à la connexion, puis un cryptage (plus léger) durant tout le temps de la session. Ce protocole fonctionne avec deux clés associées à chaque machine, une clé publique et une clé privée. Une machine peut, à partir d'un message codé avec sa clé publique, le décoder avec sa clé privée. Les clés privées doivent rester secrètes, contrairement aux clés publiques, qui doivent être communiquées. Ainsi, lors d'une connexion, la machine sur laquelle vous vous connectez envoie sa clé publique. Votre machine code les informations à envoyer, qui sont décodées à l'arrivée par la clé privée.

Les réseaux WiFi quelque soit leur niveau de sécurisation peuvent être craqués à l'aide d'outils tels que aircrack. L'idée de cet atelier est de se servir d'un honeypot WiFi, Kojoney, pour y attirer les pirates au lieu de les laisser intégrer le vrai réseau. Ceux-ci auront l'impression d'avoir craqué le réseau sans fil mais en fait c'est un réseau virtuel qu'ils accèdent.

Installation de Kojoney

Pour installer le logiciel, décompressons le dans notre dossier personnel. Voici la liste des fichiers :

```
root@serveur:/home/foulen/kojoney# ls
AUTHORS    coret_command.py  coret_honey.py     coret_users.py  init.d
KoJoney.e3p  LICENSE        UNINSTALL.sh
ChangeLog  coret_config.py   coret_log.py       docs            INSTALL
kojoney.py   PREREQUISITES  VERSION
COPYING    coret_fake.py     coret_std_unix.py  fake_users      INSTALL.sh
libs         reports
```

Le programme INSTALL.sh permet d'effectuer une installation automatique ainsi que de programmer le lancement automatique au démarrage de la machine :

```
root@serveur:/home/foulen/kojoney# ./INSTALL.sh
Kojoney Honeypot installer.

Press enter to view the license agreement ...
```

```
 <<< NOTE: After read the license agreement press 'q' to exit >>>
 Do you accept the ZPL, MIT and GPL license terms (yes/no) ?
 yes
 All licenses accepted.
 ****************************************
----------------------------------------------------------------------------------------
 Kojoney Honeypot Installer version 0.0.3
 ****************************************
----------------------------------------------------------------------------------------
 Step 1 - Copying files
 (... uninteresting information...)
 Step 2 - Building libraries
 [+] Building and installing [IP-Country]
 [+] Building and installing [Geograpy-Countries]
 [+] Building and installing [Zope Interfaces]
 [+] Building and installing [Twisted extension]
 [+] Building and installing [PyCrypto]
 (... Possibly various warnings. You can ignore these safely...)
 [+] Building and installing [Twisted Conch extension]
 Step 3 - Installing documentation
 [+] Installing man pages
 Step 4 - Changing permissions and creating symbolic links
 [+] Creating symlinks

 Step 5 - Final questions and fun

 Do you want to run it automatically at boot time (yes/no)?
 yes

 ***No run levels were assigned. You need to do this manually.***

 Do you want to run it now (yes/no)?
 yes
 Starting daemon

 Kojoney installation finished.
```

Fonctionnement de Kojoney

Kojoney est un programme qui tourne en tâche de fond et qui intercepte les communications sur le port 22 qui est celui réservé aux communications SSH, il va donc émuler un serveur SSH pour donner un semblant d'accès à la machine et ceci pour tromper notre "pirate".

Le fichier coret_fake.py contient certains paramètres que nous pouvons personnaliser pour notre faux serveur SSH :

```
 root@serveur:/home/foulen/kojoney# cat coret_fake.py
 #
 # Use the '#' if you want to emulate root
 #
```

```
FAKE_SSH_SERVER_VERSION = "SSH-2.0-OpenSSH_3.8.1p1"

FAKE_USER_CHAR = "$"

#FAKE_OS = "OpenBSD bigturd 2.5 GENERIC#172 sparc"
#FAKE_OS = "FreeBSD myname.my.domain 3.3-STABLE FreeBSD 3.3-STABLE #8: Fri
Dec 17"

#FAKE_OS =  NetBSD pc164 1.4P NetBSD 1.4P (PC164.v6-intl) #5: Sat Nov 27
18:31:37 CET 1999 root@pc164:/usr/src/sys/arch/alpha/compile/PC164.v6-intl
alpha"
#FAKE_OS = "OpenBSD 2.1 (TWP) #3: Sat Jul 19 18:37:43 CDT 1997
FAKE_HOST="darkstar"
FAKE_OS = "Linux "+FAKE_HOST+" 2.6.9 #1 Wed Jan 5 19:30:39 EST 2005 i686
i686 i386 GNU/Linux"
FAKE_SHELL = "bash-2.0"
FAKE_PROMPT = FAKE_SHELL + str(FAKE_USER_CHAR) + " "
FAKE_USER = "test"

FAKE_W = ("USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU
WHAT",
": Permission denied"
)

FAKE_LS = ("drwxr-xr-x    2 root root   4096 2005-06-06 07:00 bin",
"drwxr-xr-x    3 root root   4096 2005-06-25 16:13 boot",
"drwxr-xr-x   10 root root  14320 2005-07-10 22:19 dev",
"drwxr-xr-x  100 root root   4096 2005-07-11 20:31 etc",
"drwxr-xr-x   10 root root   8192 2005-07-10 01:33 lib",
"drwxr-xr-x    2 root root  49152 2005-05-14 18:47 lost+found",
"drwxr-xr-x    3 root root   4096 2005-07-06 23:11 opt",
"drwxr-xr-x    3 root root   4096 2005-07-06 04:30 oracle",
"drwxr-xr-x    2 root root   4096 2005-07-10 01:33 sbin",
"drwxr-xr-x    2 root root   4096 2005-05-14 18:49 srv",
"drwxr-xr-x   10 root root      0 2005-07-11 00:08 sys",
"drwxrwxrwt   11 root root   4096 2005-07-11 21:17 tmp",
"drwxr-xr-x   14 root root   4096 2005-07-10 15:52 usr",
"drwxr-xr-x   14 root root   4096 2005-06-06 07:02 var",
"lrwxrwxrwx    1 root root     25 2005-06-25 16:13 vmunix -> boot/vmunix-
2.6.5"
)
FAKE_WGET = "--00:32:24--  http://../","           => `index.html'",
"Resolviendo ..... fallo: No se ha encontrado el anfitrion."
FAKE_FTP = "ftp: ..: No address associated with name", "ftp> "
FAKE_USERS_FILE = "/etc/kojoney/fake_users"
FAKE_RM = "rm: Permission denied"
FAKE_TOUCH = "touch: Permission denied"
FAKE_DENIED = "Permission denied"
```

Il nous est aussi possible de rajouter nos propres fausses commandes.

Nous avons la possibilité aussi de permettre au "pirate" l'accès à certains vrais fichiers stockés dans le répertoire /var/log/kojoney en le définissant dans le fichier coret_config.py :

```
...
#
# When an intruder tries to download file with CURL or WGET, will I
download the file? And where?
#
DOWNLOAD_REAL_FILE = True
DOWNLOAD_REAL_DIR  = "/var/log/kojoney/"
...
```

Une fausse clé est aussi envoyée pour tromper l'attaquant :

```
 publicKey = 'ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAGEArzJx8OYOnJmzf4tfBEvLi8DVPrJ3/c9k2I/Az64fxjHf9i
myRJbixtQhlH9lfNjUIx+4LmrJH5QNRsFporcHDKOTwTTYLh5KmRpslkYHRivcJSkbh/C+BR3u
tDS555mV'

privateKey = """-----BEGIN RSA PRIVATE KEY-----
MIIByAIBAAJhAK8ycfDmDpyZs3+LXwRLy4vA1T6yd/3PZNiPwM+uH8Yx3/YpskSW
4sbUIZR/ZXzY1CMfuC5qyR+UDUbBaaK3Bwyjk8E02C4eSpkabJZGB0Yr3CUpG4fw
vgUd7rQ0ueeZlQIBIwJgbh+1VZfr7WftK5lu7MHtqE1S1vPWZQYE3+VUn8yJADyb
Z4fsZaCrzW9lkIqXkE3GIY+ojdhZhkO1gbG0118sIgphwSWKRxK0mvh6ERxKqIt1
xJEJO74EykXZV4oNJ8sjAjEA3J9r2ZghVhGN6V8DnQrTk24Td0E8hU8AcP0FVP+8
PQm/g/aXf2QQkQT+omdHVEJrAjEAy0pL0EBH6EVS98evDCBtQw22OZT52qXlAwZ2
gyTriKFVoqjeEjt3SZKKqXHSApP/AjBLpF99zcJJZRq2abgYlf9lv1chkrWqDHUu
DZttmYJeEfiFBBavVYIF1dOlZT0G8jMCMBc7sOSZodFnAiryP+Qg9otSBjJ3bQML
pSTqy7c3a2AScC/YyOwkDaICHnnD3XyjMwIxALRzl0tQEKMXs6hH8ToUdlLROCrP
EhQ0wahUTCk1gKA4uPD6TMTChavbh4K63OvbKg==
-----END RSA PRIVATE KEY-----"""
```

Pour permettre à l'attaquant de pénétrer le système, en général par brute force, nous disposons d'une liste d'utilisateurs virtuels avec de faux mots de passe. Cette liste se trouve dans le fichier fake_users :

```
...
webmaster webmasters
webmaster www
webmaster1 webmaster1
website website
webstyle_bu webstyle_bu
webstyleinternet webstyleinternet
webstyleuk webstyleuk
webtrends webtrends
wednesday wednesday
weisz weisz
wel wel
wel6375 wel6375
weldon weldon
```

```
welfare welfare
welfare119 welfare119
...
test test
guest guest
user user
root root
mysql mysql
webmaster webmaster
linux linux
unix unix
bsd bsd
administrator administrator
fake fake
testuser testuser
alex alex
adam adam
...
```

Exemple de console pour le pirate :

Il faut personnaliser la configuration de Kojoney qui se trouve dans le fichier kojoney.py :

```
foulen@station01:~$ ssh serveur
The authenticity of host 'serveur (192.168.1.254)' can't be established.
RSA key fingerprint is 3d:13:5f:cb:c9:79:8a:93:06:27:65:bc:3d:0b:8f:af.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'serveur,192.168.1.254' (RSA) to the list of
known hosts.
foulen@serveur's password:
Wellcome to Linux alpha-server 2.6.9 #1 Wed Jan 5 19:30:39 EST 2006 i686
i686 i386 GNU/Linux!
foulen@alpha-server$ ls
drwxr-xr-x    2 root root   4096 2006-06-06 07:00 bin
drwxr-xr-x    3 root root   4096 2006-06-25 16:13 boot
drwxr-xr-x   10 root root  14320 2006-07-10 22:19 dev
drwxr-xr-x  100 root root   4096 2006-07-11 20:31 etc
drwxr-xr-x  192 root root   4096 2006-07-11 20:40 home
drwxr-xr-x   10 root root   8192 2006-07-10 01:33 lib
drwxr-xr-x    2 root root  49152 2006-05-14 18:47 lost+found
drwxr-xr-x    3 root root   4096 2006-07-06 23:11 opt
drwxr-xr-x    3 root root   4096 2006-07-06 04:30 oracle
drwxr-xr-x   24 root root   4096 2006-07-11 20:22 root
drwxr-xr-x    2 root root   4096 2006-07-10 01:33 sbin
drwxr-xr-x    2 root root   4096 2006-05-14 18:49 srv
drwxr-xr-x   10 root root      0 2006-07-11 00:08 sys
drwxrwxrwt   11 root root   4096 2006-07-11 21:17 tmp
drwxr-xr-x   14 root root   4096 2006-07-10 15:52 usr
drwxr-xr-x   14 root root   4096 2006-06-06 07:02 var
```

```
lrwxrwxrwx    1 root root     25 2006-06-25 16:13 vmunix -> boot/vmunix-
2.6.5
foulen@alpha-server$ cat /proc/cpuinfo
processor       : 0
vendor_id       : AuthenticAMD
cpu family      : 15
model           : 44
model name      : AMD Sempron(tm) Processor 2800+
stepping        : 2
cpu MHz         : 1607.390
cache size      : 256 KB
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 1
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
cmov pat pse36 clflush mmx fxsr sse sse2 syscall nx mmxext fxsr_opt lm
3dnowext 3dnow up pni lahf_lm ts ttp tm stc
bogomips        : 3217.60
foulen@alpha-server$ who
foulen
foulen@alpha-server$
```

Les rapports d'activités de Kojoney

Nous retrouvons bien sur dans le fichier des logs la trace de ma tentative de "piratage" :

```
root@serveur:/var/log# cat honeypot.log
2007/04/04 20:05 CEST [SSHServerTransport,31,192.168.1.2] kex
alg, key alg: diffie-hellman-group1-sha1 ssh-rsa
...
```

Il y a aussi des utilitaires pour l'analyse des données :

- kojreport : génération de rapport avec le fichier des logs
- kojreport-filter : même fonction que le précédent avec la possibilité d'inclure des options de filtre pour affiner le rapport
- kip2country : génération de rapport par pays d'aprés les adresses IP
- kojhumans : génération de rapport avec vérification de l'attaquant (humain ou bot)

SECURINETS
Club de la sécurité informatique
INSAT

Atelier Honeypot WiFi

Honeyd

Sécurinets
Club de la sécurité informatique
INSAT
www.securinets.com
Tel : 20322191

Honeyd

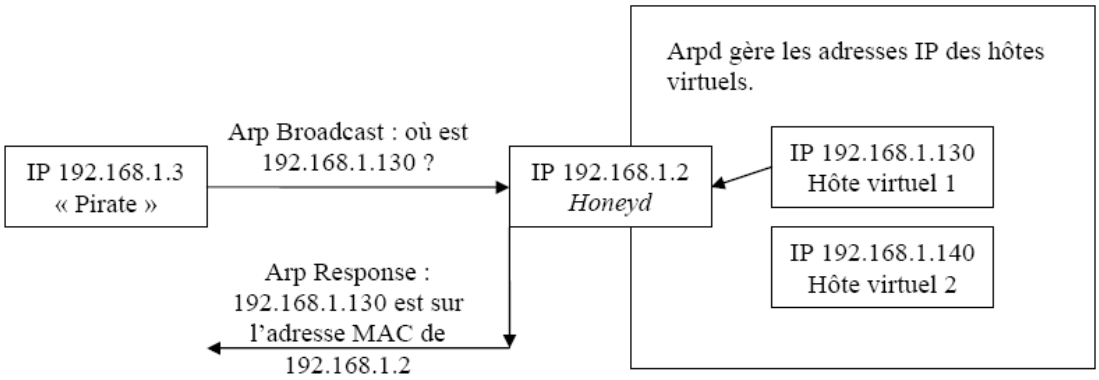
Honeyd est un projet OpenSource développé par Niels Provos. C'est un *honeypot* à faible interaction qui permet de déployer des machines virtuelles sur un réseau dans le but de détecter toute activité illégale sur le réseau. En effet, toute connexion sur une adresse IP d'une machine virtuelle est considérée comme suspecte.

Honeyd fonctionne sous plusieurs environnements (Unix/Linux, Solaris, Windows, BSD), il est capable de simuler sur un réseau, la présence de machines de tous types (serveurs, PC, routeurs,…). Les serveurs peuvent être configurés pour qu'ils paraissent fonctionner sous certains systèmes d'exploitation. De plus pour accroître l'effet de la réalité, ces machines peuvent virtuellement faire tourner des services, qui sont en réalité des scripts écrits en perl, python ou encore en shell-script et ainsi un réseau de production peut être simulé entièrement.

Fonctionnement détaillé de honeyd

*) Outil Arpd

Honeyd associe à ses machines virtuelles des adresses IP qui ne sont pas encore attribuées sur le réseau. Pour gérer les requêtes ARP destinées aux hôtes virtuels, *honeyd* doit être utilisé en collaboration avec l'outil Arpd. Il est chargé de répondre aux requêtes en renvoyant l'adresse MAC de la machine hébergeant l'*honeyd*. *Honeyd* recevra ensuite tout le trafic qui lui est destiné, et pourra répondre si nécessaire.



L'Arpd cherche l'adresse IP demandée dans la table ARP que l'on a spécifiée. Si l'adresse figure dans cette table, il renverra l'adresse MAC de l'*honeyd*.

*) Association d'un hôte virtuel à une adresse IP

Les hôtes virtuels sont décrits dans le fichier *honeyd.conf* selon un modèle contenant des informations telles que :

Personality : correspond au type d'OS que la machine virtuelle est sensée héberger. Exemple:

```
create win2k #création d'une machine virtuelle
set win2k personality "Microsoft Windows 2000 SP3" #définition du type d'OS
```

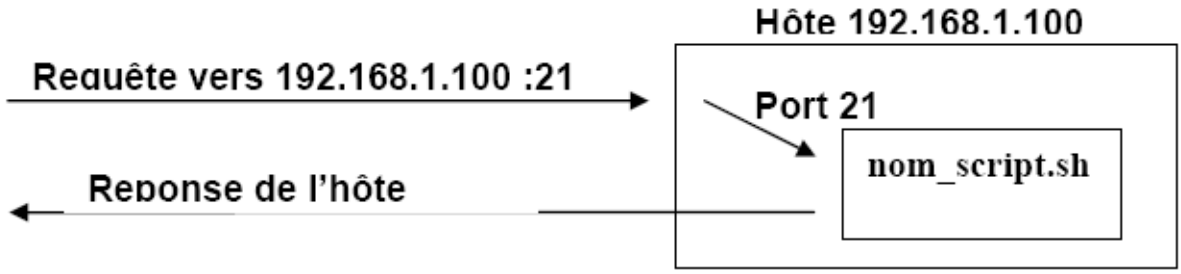
L'état des ports : ouverts ou non qui peuvent être associés à des scripts simulant des services. Exemple:

```
add win2k tcp port 21 "sh nom_script.sh $param1 $param2 …"
```

Les hôtes sont associés à une adresse IP grâce à la commande bind qui associe la machine virtuelle à une adresse IP

Exemple : `bind 192.168.1.100 win2k`

Avec ce modèle, les paquets en direction de l'hôte ayant l'adresse IP 192.168.1.100 et sur le port 21 seront dirigés vers le script shell associé qui se chargera de répondre.



Installation de honeyd

La version de honeyd pour Windows étant en binaire, nous avons opté pour celle relative à linux car cette dernière est opensource, ce qui signifie qu'on a accès aux sources et ainsi nous pouvons modifier et créer des nouveaux scripts simulant des services nouveaux.

Le paquetage d'installation honeyd contient les fichiers nécessaires à sa compilation, mais en plus honeyd a besoin de trois librairies pour son fonctionnement :

- libevent : fournit une API permettant d'associer une fonction de call-back à un événement donné (time-out, signal, événement sur un descripteur de fichier,…)

- libpcap : permet la capture de paquets, est utilisée dans de nombreux sniffers.

- libdnet : offre une interface pour différentes fonctionnalités liées au réseau.

très simple. Il suffit d'exécuter les commandes suivantes sur la console en tant qu'administrateur (root) :

```
./configure
Make
make install
```

L'installation de honeyd s'effectue aussi de la même façon. Normalement, il n'y a pas de problème si les librairies ont bien été installées.

Configuration de honeyd

Honeyd est configuré à travers un fichier de configuration personnalisable (honeyd.conf) où toutes les machines virtuelles sont décrites l'une après l'autre.

Pour que honeyd puisse lire et écrire, le propriétaire de tous les fichiers du répertoire honeyd doit être mis à la valeur : 'nobody' par la commande :

```
#chown –R nobody honeyd_kit
```

Ensuite, un répertoire pour les fichiers log doit être créé et avec la valeur 'nobody' :

```
#mkdir /var/log/honeyd
#chown nobody /var/log/honeyd
```

Avant d'exécuter honeyd, l'outil Arpd nécessite d'être lancé.

```
# ./arpd 192.168.1.0/24 (par exemple)
```

Les options d'appel de *honeyd* sont :

-f : fichier de configuration : *honeyd.conf*
-x : fichier contenant les empreintes d'OS : *xprobe2.conf*
-p : fichier contenant les empreintes : *nmap.prints*
-a : fichier contenant les associations d'empreintes : *nmap.assoc*
-l : fichier des logs paquets : */var/log/honeyd*
-i : interface réseau

Exemple d'activation de *honeyd* pour la plage d'adresses:

```
# ./honeyd -f honeyd.conf -p nmap.prints -x xprobe2.conf -a nmap.assoc -l /var/log/honeyd 192.168.1.130 - 192.168.1.253.
```

Pour pouvoir utiliser honeyd et arpd, il est nécessaire que vous soyez l'administrateur et que le câble réseau soit bien branché. Arpd détecte si le câble est branché ou non. Sinon, si vous travaillez sur une interface sans fil (le cas de notre atelier) il faudrait spécifier le nom de l'interface réseau active par l'intermédiaire de l'option - i de honeyd.

Routing Topologies

Pour aller au-delà d'un simple pot de miel, l'utilisation de routing topologies permet de réaliser un véritables réseau de pots de miels (ou honeynet). Grâce à l'utilisation de configuration spécifique, honeyd est alors capable de simuler un réseau entier, quelle qu'en soit sa capacité (il est à noter qu'au delà d'un certain nombre d'hosts, les capacités et performances du démon sont très réduites).

Afin de créer l'architecture réseau de nos rêves avec honeyd, nous avons alors à notre disposition trois commandes :

- *route entry*, qui permet de définir un routeur officiant comme étant le point d'entrée dans le réseau. Deux solutions s'offrent alors à nous : soit on définit ici l'adresse IP réelle de la machine, et celle-ci se comporte alors comme étant elle-même le routeur d'entrée dans le HoneyNet, soit on utilise là aussi une adresse IP virtuelle (on devra alors encore faire appel à ARPD);
- *route link*, avec laquelle on spécifie la plage d'adresses accessibles via le routeur ;

- *route add net*, qui sert à définir un nouveau routeur, et le sous-réseau auquel il donne accès.

Outre la disposition des différents hôtes virtuels faisant offices de routeurs (auxquels nous pouvons -devons ?- bien sûr associer un modèle), honeyd permet à l'utilisateur d'améliorer le réalisme de son dispositif en introduisant des paramètres de latence et de perte de paquets.

Voici en outre un exemple concret de fichier *honeyd.conf :*

```
# Création d'un hôte Microsoft Windows XP Pro SP1
create windows

# Personnalité
set windows personality "Microsoft Windows XP Professional SP1"
set windows uptime 1728650
set windows uid 65534 gid 65534

# Ouverture de ports spécifiques
# Plusieurs ports spécifiques au monde Microsoft sont ouverts afin de coller
# au plus à la réalité
add windows tcp port 135 open # msrpc
add windows udp port 137 open # Browsing Requests
add windows udp port 138 open # Browsing Responses
add windows tcp port 139 open # netbios-ssn
add windows tcp port 445 open # microsoft-ds
add windows udp port 445 open # microsoft-ds

# Actions par défaut sur les ports
set windows default tcp action reset
set windows default udp action reset
set windows default icmp action open

# Création d'un serveur web sous Linux :
create serv_web

# Personnalité
set serv_web personality "Linux Kernel 2.4.3 SMP (RedHat)"
set serv_web uptime 354136546
set serv_web uid 65534 gid 65534

# Services
add serv_web tcp port 80 "scripts/HoneyWeb-0.4/HoneyWeb-0.4.py" # Web
add serv_web tcp port 22 "sh scripts/ssh.sh" # Serveur SSH

# Actions par défaut sur les ports
set serv_web default tcp action reset
set serv_web default udp action reset
set serv_web default icmp action open

# Routeur CISCO - IOS 11.1(24)
create router
```

```
# Personnalité
set router personality "Cisco 7206 running IOS 11.1(24)"

# Services
add router tcp port 23 "perl scripts/cisco/router-telnet.pl" # Serveur Telnet

# Actions par défaut sur les ports
set router default tcp action reset
set router default icmp action open

# Machine par défaut
create default
# Toute tentative de connexion est bloquée (l'hôte n'existe pas)
set default default tcp action block
set default default udp action block
set default default icmp action block
```

Notre parc informatique virtuel se compose de PC sous Windows XP Pro SP1. Nous avons un routeur CISCO ayant une patte vers une DMZ et d'autres pattes pour différents sous réseaux.

Conclusion

Le honeypot est une solution qui a l'air d'être efficace pour contrer les pirates et même mieux c'est-à-dire les piéger. Toutefois, son utilisation reste limitée car placer un honeypot sur Internet est une activité illégale et aujourd'hui presque toutes les sociétés sont connectées à Internet...