```
ns      IN      A       212.94.201.10
mail    IN      A       212.94.201.10
mail2   IN      A       212.94.201.11
www     IN      A       212.94.201.11

dns     IN      CNAME   ns
```

EXEMPLE Extrait du fichier /etc/bind/db.192.168

```
; Zone inverse pour 192.168.0.0/16
; admin.falcot.com. => contact pour la zone: admin@falcot.com
$TTL    604800
@       IN      SOA     ns.interne.falcot.com. admin.falcot.com. (
                        20040121        ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL

        IN      NS      ns.interne.falcot.com.

; 192.168.0.1 -> arrakis
1.0     IN      PTR     arrakis.interne.falcot.com.
; 192.168.0.2 -> neptune
2.0     IN      PTR     neptune.interne.falcot.com.

; 192.168.3.1 -> pau
1.3     IN      PTR     pau.interne.falcot.com.
```

DHCP

Présentation

DHCP (*Dynamic Host Configuration Protocol*, ou protocole de configuration dynamique des hôtes) est un moyen de rapatrier automatiquement sa configuration pour une machine qui vient de démarrer et souhaite configurer son interface réseau. De cette manière, on peut centraliser la gestion des configurations réseau et toutes les machines bureautiques pourront recevoir des réglages identiques.

Un serveur DHCP fournit de nombreux paramètres réseau, et notamment une adresse IP et le réseau d'appartenance de la machine. Mais il peut aussi indiquer d'autres informations, telles que les serveurs DNS, WINS, NTP.

C'est encore l'*Internet Software Consortium* qui développe le serveur DHCP (avec *bind*). Le paquet Debian correspondant est *dhcp3-server*.

Configuration

Les premiers éléments à modifier dans le fichier de configuration du serveur DHCP, /etc/dhcp3/dhcpd.conf, sont le nom de domaine et les serveurs DNS. Il faut aussi activer (en la décommentant) l'option *authoritative* si ce serveur est le seul sur le réseau local (tel que défini par la limite de propagation du *broadcast*, mécanisme employé pour joindre le serveur DHCP). On créera aussi une section *subnet* décrivant le réseau local et les informations de configuration

diffusées. L'exemple ci-dessous convient pour le réseau local `192.168.0.0/24`, qui dispose d'un routeur (`192.168.0.1`) faisant office de passerelle externe. Les adresses IP disponibles sont comprises entre `192.168.0.128` et `192.168.0.254`.

EXEMPLE Extrait du fichier /etc/dhcp3/dhcpd.conf

```
#
# Sample configuration file for ISC dhcpd for Debian
#

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
ddns-update-style interim;

# option definitions common to all supported networks...
option domain-name "interne.falcot.com";
option domain-name-servers ns.interne.falcot.com;

default-lease-time 600;
max-lease-time 7200;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# My subnet
subnet 192.168.0.0 netmask 255.255.255.0 {
    option routers 192.168.0.1;
    option broadcast-address 192.168.0.255;
    range 192.168.0.128 192.168.0.254;
    ddns-domainname "interne.falcot.com";
}
```

DHCP et DNS

Une fonctionnalité appréciée est l'enregistrement automatique des clients DHCP dans la zone DNS de sorte que chaque machine ait un nom significatif (et pas automatique comme `machine-192-168-0-131.interne.falcot.com`). Pour exploiter cette possibilité, il faut autoriser le serveur DHCP à mettre à jour la zone DNS `interne.falcot.com` et configurer celui-ci pour qu'il s'en charge.

Dans le cas de **bind**, on ajoutera la directive *allow-update* aux deux zones que le serveur DHCP devra modifier (celle du domaine *interne.falcot.com* et celle de la résolution inverse). Cette directive donne la liste des adresses autorisées à effectuer la mise à jour ; on y consignera donc les adresses possibles du serveur DHCP (adresses IP locales et publiques le cas échéant).

```
allow-update { 127.0.0.1 192.168.0.1 212.94.201.10 !any };
```

Attention ! Une zone modifiable sera changée par *bind*, qui va donc réécrire régulièrement ses fichiers de configuration. Cette procédure automatique produisant des fichiers moins lisibles que les productions manuelles, les administrateurs de Falcot gèrent le sous-domaine *interne.falcot.com* à l'aide d'un serveur DNS délégué. Le fichier de la zone *falcot.com* reste ainsi entièrement sous leur contrôle.

L'exemple de fichier de configuration de serveur DHCP de la section précédente comporte déjà les directives nécessaires à l'activation de la mise à jour du DNS : il s'agit des lignes `ddns-update-style interim;` et `ddns-domain-name "interne.falcot.com";` dans le bloc décrivant le réseau.

Détection d'intrusion (IDS/NIDS)

snort (du paquet Debian éponyme) est un outil de détection d'intrusions : il écoute en permanence le réseau pour repérer les tentatives d'infiltration et/ou les actes malveillants (notamment les dénis de service). Tous ces événements sont enregistrés puis signalés quotidiennement à l'administrateur par un message électronique résumant les dernières 24 heures.

Son installation demande plusieurs informations. Il faut ainsi y préciser la plage d'adresses couvertes par le réseau local : il s'agit en réalité d'indiquer toutes les cibles potentielles d'attaques. On précisera également l'interface réseau à surveiller. Il s'agit en général d'*eth0* pour une connexion Ethernet, mais on pourra aussi trouver *ppp0* pour une connexion ADSL ou RTC (Réseau Téléphonique Commuté, ou modem classique).

Le fichier de configuration de **snort** (/etc/snort/snort.conf) est très long et ses abondants commentaires y détaillent le rôle de chaque directive. Il est fortement recommandé de le parcourir et de l'adapter à la situation locale pour en tirer le meilleur parti. En effet, il est possible d'y indiquer les machines hébergeant chaque service pour limiter le nombre d'incidents rapportés par **snort** (un déni de service sur une machine bureautique n'est pas aussi dramatique que sur un serveur DNS). On peut encore y renseigner les correspondances entre adresses IP et MAC (il s'agit d'un numéro unique identifiant chaque carte réseau) pour détecter les attaques par *ARP-spoofing* (travestissement d'ARP), qui permettent à une machine compromise de se substituer à une autre (un serveur sensible par exemple).

ATTENTION **Rayon d'action**

snort est limité par le trafic qu'il voit transiter sur son interface réseau : il ne pourra évidemment rien détecter s'il n'observe rien. Branché sur un commutateur (*switch*), il ne surveillera que les attaques ciblant la machine l'hébergeant, ce qui n'a qu'un intérêt assez limité. Pensez donc à relier la machine employant **snort** au port « miroir », qui permet habituellement de chaîner les commutateurs et sur lequel tout le trafic est dupliqué.
Pour un petit réseau doté d'un concentrateur (*hub*), le problème ne se pose pas : toutes les machines reçoivent tout le trafic.

B.A.-BA **Dénis de service**

Une attaque de type « déni de service » a pour seul objectif de rendre un service réseau inexploitable. Que cela soit en surchargeant le serveur de requêtes ou en exploitant un bogue de celui-ci, le résultat est toujours le même : le service en question n'est plus fonctionnel, les utilisateurs habituels sont mécontents et l'hébergeur du service réseau visé s'est fait une mauvaise publicité.

ALTERNATIVE **Un autre NIDS : `prelude`**

Le NIDS *snort* (*Network Intrusion Detection System*, ou système de détection d'intrusions sur le réseau) est très répandu, mais il compte depuis peu un concurrent moins éprouvé que lui : **prelude**, qui jouit d'une architecture plus modulaire. Un serveur (le *manager* du paquet *prelude-manager*) y centralise les alertes détectées par des capteurs (*sensors*) de plusieurs types. Le paquet *prelude-nids* propose un capteur qui, comme **snort**, surveille le trafic réseau. Le paquet *prelude-lml* (*Log Monitor Lackey*, ou laquais de surveillance de journaux système) surveille quant à lui les fichiers de *logs*, à l'instar de **logcheck**, déjà étudié.