TCP/IP & sécurité

Préliminaires

La sécurité des hôtes sur un réseau, et donc sur le Net, est un vaste problème.

L'objectif de ce document n'est certes pas d'être une référence à l'usage des spécialistes, mais plutôt un exposé des connaissances de base qui permettent d'entrevoir les dangers encourus par un utilisateur tel qu'un internaute câblé ou "ADSLisé".

Mais que risque-t-on ?

Malheureusement beaucoup. TCP/IP n'est pas un modèle de sécurité et de nombreux spécialistes ont mis à jour des "trous" qui permettent de s'introduire frauduleusement dans les machines des autres. Comme le monde virtuel n'est pas bien différent du monde réel, nous y trouverons aussi des cambrioleurs, des voyeurs, des casseurs etc. avec toutes les nuisances que ça laisse supposer. Il est donc nécessaire de se protéger de ces agresseurs.

Firewall, c'est quoi ça ?

Traduit plus ou moins harmonieusement par "mur pare-feu", les firewalls sont normalement des systèmes dédiés à la sécurité d'un réseau.

Dans l'absolu, un firewall devrait être un dispositif informatique qui s'intercale entre le réseau privé et la connexion Internet. Comme c'est lui qui va prendre les coups, il vaut mieux qu'il soit solide et qu'il soit dédié à cette tâche.

Malheureusement, les choses ne sont pas toujours aussi simples et de nombreuses entorses à cette règle basique sont souvent nécessaires.

- Le coût.
  Installer selon cette règle un firewall pour protéger une machine unique augmente l'investissement dans des proportions considérables (encore qu'un vieux PC type P75 ,16 Mo de RAM, DD de 500Mo et une installation de Linux bien configurée, ça peut se faire pour pas très cher).
- Les services.
  On a souvent besoin d'installer des services qui doivent communiquer directement avec le Net (DNS, SMTP, FTP, HTTP) et dans ces cas là, un firewall pur et dur devient un vrai casse tête. On parle alors de DMZ (Zone démilitarisée), sorte de "purgatoire" un peu protégé dans lequel les serveurs publics jouissent d'une relative sécurité derrière un barrage filtrant, mais non protégés par un firewall plus strict, dont le rôle reste de protéger la partie privée du réseau.

D'autres compromis sont possibles. Le firewall pourra être une machine exposée, donc protégée par des logiciels appropriés, mais elle servira également à d'autres tâches. C'est le cas d'un poste isolé, seulement connecté au Net. C'est aussi le cas de ma configuration où la machine exposée sert

également de passerelle pour le réseau privé, de DNS et de relais SMTP.

Finalement, Le firewall apparaît plutôt comme un ensemble de règles de sécurité pour la configuration de la machine, avec un logiciel de filtrage de paquets (IPChains ou, plus récemment IPTables sous Linux), un logiciel de surveillance (iplog ou snort, par exemple sous Linux), voire un logiciel capable de construire une protection particulière lorsqu'il détecte les prémices d'une intrusion (portsentry, toujours sous Linux).

D'autres logiciels qui cumulent ces fonctions existent dans le monde Windows, nous parlerons un peu de ZoneAlarm et de Look 'n Stop, mais il en existe beaucoup d'autres. Leur principe reste cependant plus ou moins le même, chaque paquet entrant est vérifié et, s'il correspond à certains critères, est bloqué, tracé, accepté etc.

Plan du chapitre

Attaques

Hasard ou nécessité ?

La bonne question à se poser, concernant les problèmes de sécurité n'est pas :

> *"**Est-ce que j'ai des chances** (grandes ou petites) de subir une attaque un jour?"*

Mais :

> *"**Quand** vais-je être la cible d'une attaque?"*

Et la seule réponse pertinente à cette question est :

> *"**A tout moment**. Peut-être justement pendant que tu lis ces lignes :-)"*

Mon propos n'est certes pas d'affoler le lecteur, mais d'essayer de lui faire comprendre que les arguments du type :

> *"Oh, moi, je n'ai rien d'intéressant sur ma machine et je ne vois pas pourquoi un pirate s'ennuierait à essayer d'y pénétrer..."*

Sont du même ordre que de croire que les accidents de voiture sont pour les autres, jamais pour soi.

Les attaques possibles

Une machine informatique est par défaut, d'une grande vulnérabilité, tout simplement, parce que l'informatique est passée des mains d'une poignée de spécialistes à une foule d'utilisateurs et qu'une certaine éthique de l'informaticien y a laissé ses plumes.

Les environnements informatiques n'ont pas été prévus pour être manipulés par des personnes sans scrupules qui cherchent à en exploiter tous les effets pervers.

Je classerai les malfaisants dans trois catégories (c'est une opinion toute personnelle) :

1. *Les amateurs de génocides*
   Ils utilisent des virus pour produire le plus de dégâts possible sans se soucier de savoir quelles sont les cibles qui seront touchées. Ce qui leur importe est la masse atteinte. Les meilleurs écrivent eux-mêmes leurs virus, ce qui fait au moins la preuve de leur compétence technique, les pires exploitent des virus faits par d'autres, sans même savoir comment ils fonctionnent.
2. *Les voyous du Net*
   Bien connus sous le nom de "script kiddies", ceux là cherchent à pénétrer, le plus souvent pour faire de la casse, des hôtes connectés au Net en utilisant des recettes de cuisine élaborées par d'autres. Nous verrons plus loin de quoi il retourne. Peu leur importe la cible, leur amusement consiste à utiliser des méthodes toutes faites pour "ennuyer" leur monde. Ils s'attaquent à la première machine sur laquelle ils trouvent une faille, juste pour le plaisir. Mes logs sont pleins de petits comiques de ce genre, qui cherchent au hasard du Net, une machine qui serait infectée par un cheval de Troie, juste, sans doute, parce qu'ils ont trouvé

quelque part le client qui sait s'y connecter.

3. *Les Hackers*
   Les vrais, ceux qui ont des connaissances très étendues dans les systèmes et pour qui le jeu consiste à rechercher toujours de nouvelles failles. Ceux là sont de véritables techniciens et, même s'ils font parfois des dégâts, forcent plus ou moins le respect par leur grandes connaissances. Ce sont leurs découvertes qui, la plupart du temps, sont exploitées par les script kiddies et autres utilisateurs de virus. Ce sont également leurs découvertes qui contribuent à créer des systèmes de plus en plus solides et fiables. A priori, ce ne sont pas des pirates et leur objectif premier n'est pas de détruire, mais de comprendre.

Les attaques virales et assimilées

Les virus dans les exécutables

Un "virus" est un bout de programme glissé volontairement dans une application dans le but de nuire. Il est possible d'attraper un virus avec n'importe quelle application que l'on a installée et que l'on exécute, ce n'est pas un problème typique d'une connexion permanente. Un virus ne peut être introduit dans sa machine que si l'on exécute une application infectée, application récupérée sur l'Internet ou sur n'importe quel autre support informatique: Disquette, CD ROM etc.

Les macro virus dans les données

Une autre infection, assez semblable, consiste à exploiter les possibilités qu'ont certaines applications d'installer des macro commandes dans les fichiers de données. La suite Microsoft Office qui propose des possibilités, par ailleurs intéressantes, de placer des macros dans les documents Word, Excel et même Powerpoint est une cible de choix. Ces macros sont maintenant écrites en VBA (Visual Basic for Applications) qui est un langage suffisamment puissant pour arriver à faire beaucoup de dégâts avec. Dans un tel cas, il suffit d'ouvrir un document infecté pour mettre le "macro virus" en activité. Autrement dit, même des fichiers de données peuvent être dangereux. Notez que Microsoft a modifié les applications de MS Office de manière à ce qu'elles puissent vous avertir de la présence de macros dans les documents, vous laissant la possibilité de les activer ou non.

Les scripts et applets dans le HTML

Malheureusement, d'autres moyens existent, typiquement venus de l'Internet, dans les pages HTML. En effet, pour rendre les pages HTML plus vivantes, il devient possible d'y insérer des composants actifs. Parmi ceux-ci nous trouvons:

- Les scripts (javascript, vbscript).
  Ce ne sont pas les plus dangereux parce que les langages de scripts offrent rarement des fonctions pouvant être utilisées à des fins vraiment destructrices. Ils disposent cependant de la possibilité de lancer des exécutables locaux, c'est en cela qu'ils peuvent devenir dangereux.
- Les applets Java ou les composants ActiveX.
  Plus puissants, ils sont introduits dans les pages HTML sous la forme de composants compilés (ou pré compilés). Leur contenu n'est pas visible et les outils qui permettent de les construire (Java ou Visual Basic) offrent des fonctions permettant de réaliser des opérations

extrêmement dangereuses.
La grande mode consiste en un mélande des deux. Un message malicieux en HTML pour exécuter un script qui exécute un applet ou un ActiveX, le résultat étant par exemple que votre carnet d'adresse sera utilisé à votre insu pour "spammer" à toutes vos connaissances le même message malicieux ou mieux encore, un autre message, mais contenant le même ver.

- Les "plug-in" qui sont des extensions ajoutées aux navigateurs peuvent également être corrompus.

Parades :

Toutes ces attaques sont plus ou moins prises en charge par des applications anti-virus, les meilleures précautions à prendre sont :

- Installer un "bon" antivirus et effectuer une mise à jour fréquente de la base de données de cet outil. Mais attention, un antivirus est efficace si le fichier exécutable (ou le fichier de données pour les macro virus) est au préalable enregistré sur le disque, pas s'il est lancé (ou ouvert) directement depuis le serveur...
- Ne pas faire une confiance aveugle à son antivirus, sous prétexte qu'il est "bon" et à jour.
- Rester très prudent sur les sources des documents ou applications que l'on rapatrie sur la machine.
- Paramétrer son navigateur pour ne pas laisser exécuter n'importe quel script ou applet dont l'origine n'est pas sûre.
- Éventuellement, prier.

Les intrusions

Décrire dans le détail les méthodes employées serait long, voire fastidieux. Ceux qui sont avides de détails sur la question ont intérêt à se procurer le très instructif "Halte aux Hackers"[1] édité en français chez Eyrolles[2]. Cet ouvrage traite en quelques 600 pages des diverses techniques de piratage, ainsi que les parades possibles. Voyons tout de même en résumé les principaux risques.

Les ports à l'écoute

Lorsqu'un port est ouvert à l'écoute sur un service serveur, c'est une porte ouverte par laquelle un intrus peut entrer. Sur un serveur, on peut entrer avec des outils comme telnet et exploiter des failles de ces logiciels.

Je vous entend me dire "Oui, mais ma machine Windows xx n'est pas un serveur, il n'y a donc pas de ports à l'écoute"... En êtes-vous si sûr?

Vous avez déjà les ports 137, 138 et 139 qui sont ouverts pour que NetBIOS fonctionne (la partie la plus visible étant le voisinage réseau). Surtout, si vous avez le partage des fichiers et des imprimantes activé. Dans ce cas là, vous êtes bel et bien un serveur.

SI vous avez installé PWS (Personal Web Server), nécessaire pour travailler efficacement avec FrontPage, vous avez également le port 80 qui est ouvert (et vous êtes particulièrement en danger).

Vous êtes donc peut-être beaucoup plus serveur que vous ne le pensez...

1 Halte aux Hackers : http://www.eyrolles.fr/php.informatique/Ouvrages/ouvrage.php3?ouv_ean13=9782746402034&xd=ef43061271a0d3ca48c1cc501ebc7bc7
2 Eyrolles : http://www.eyrolles.fr/

Les "backdoors", "trojans" et assimilés

Une porte dérobée n'est pas à proprement parler un virus, dans la mesure où elle ne se multiplie pas. Elle peut cependant s'attraper sensiblement de la même manière, par un cheval de Troie. Un cheval de Troie est une application, d'apparence inoffensive qui installe discrètement une porte dérobée. Elle peut également être inoculée par un pirate qui a réussi une opération de "spoofing", un débordement de pile ou de prise de contrôle à distance sur votre poste, comme nous le verrons plus loin.

Une porte dérobée est en gros un logiciel de contrôle à distance. Il fonctionne comme un serveur, sur un port connu de celui qui a conçu le piège. Un simple scan d'adresses IP sur ce port permet alors de repérer les machines infectées actuellement en ligne. Le mal intentionné peut s'y connecter et faire plus ou moins ce qu'il veut sur la machine distante. Très dangereux, ce genre de saleté peut heureusement être repéré relativement simplement, en prenant la précaution de vérifier périodiquement les ports ouverts sur sa machine. Malheureusement, l'utilisateur a souvent autre chose à faire que de surveiller continuellement les ports ouverts.

Une variante est le "spyware" très à la mode actuellement. Ce n'est pas dangereux à proprement parler, mais ça envoie des informations diverses sur le contenu de votre machine, vos habitudes sur l'Internet etc. à un serveur qui les récupère. Les "spywares" sont souvent implantés dans des logiciels en démonstration ou des "sharewares", de la même manière qu'une porte dérobée.

Parades :

- Contrôler périodiquement les ports ouverts à l'écoute.
- Certains antivirus savent détecter les "trojans", "backdoors" et "spywares" connus

Les défauts logiciels

Il est également possible d'exploiter des failles de sécurité sur des applications serveur "officielles" pour les utiliser comme porte d'entrée, en général par débordement de pile. Il arrive aussi que certains logiciels serveurs comportent des "bugs" ou soient mal configurés et permettent de prendre la main sur une machine, les serveurs FTP mal configurés sont un danger immédiat , mais tout type de serveur peut présenter des failles de sécurité pouvant déboucher sur une prise de contrôle. Le pirate qui réussit l'opération peut alors installer une porte dérobée pour la suite des opérations.

Parades :

- Ne pas installer de serveur inutile.
- Mettre en place toutes les sécurités proposées par les serveurs que l'on a installés et vérifier périodiquement sur les sites des constructeurs de ces logiciels l'apparition de mises à jour de sécurité.
- Filtrer efficacement les accès sur les ports que l'on souhaite laisser ouverts.
- D'une manière générale, se tenir au courant des patchs, service packs et autres correctifs qui sortent et les installer systématiquement.

Les "spoofing", "hijacking" et autres "SYN Flood"

Le jeu consiste à se faire passer pour un autre au cours d'une connexion TCP. Le principe est assez compliqué, mais redoutable s' il réussit. En général, le pirate utilise ces méthodes pour placer une porte dérobée qu'il utilisera par la suite. Vous serez sans doute tout à fait rassurés de savoir qu'il

traîne sur l'Internet des programmes spécialement conçus pour ce genre d'intrusions.

Il est très délicat de se protéger de ces d'attaques.

Parades :

Pas à ma connaissance, sauf si le pirate utilise un "SYN Flood" et que le logiciel firewall sait le détecter. Ici, il faut être préventif et curatif. Comme un spoofing ne s'improvise pas, le mal intentionné a déjà certainement pas mal tourné autour de votre machine. Prise d'empreinte de la pile TCP/IP, scan des ports ouverts et c'est à ce niveau qu'il faut le débusquer et le coincer. Par ailleurs, il se contentera dans cette phase d'installer une porte dérobée ou de créer un compte d'administrateur, choses qui sont détectables si l'on y prend garde en vérifiant périodiquement l'état de son système.

Les blocages

Ce n'est pas à proprement parler une intrusion. C'est assez facile à faire, c'est pas forcément dangereux, mais la machine ciblée se bloque, forçant parfois un "reset" sauvage, on appelle ça un "denial of Service".

Pour ce genre d'attaque, le "méchant" utilise des failles dans le NOS pour bloquer le système distant. Le "ping de la mort" en est un bon exemple. Le jeu consiste à envoyer un "echo request" avec une trame anormalement longue. Certains systèmes y sont sensibles et se bloquent. Le "méchant" n'y gagne rien, il a juste la joie de vous avoir obligé à faire un reset.

Parades :

Un firewall bien configuré arrive généralement à éviter ce genre de problèmes.

ICMP, le proto qui fait peur

Rappelons ici que le protocole ICMP est avant tout destiné au transport d'informations sur le fonctionnement du réseau. Il travaille au même niveau qu'IP. Vous trouverez des détails sur ce protocole ici[3].

ICMP couvre tous les besoins de signalisation des équipements de réseau. Il est clair que l'utilisateur final n'a pas besoin de tous les signaux possibles, c'est cependant une erreur de croire que l'on peut tous les bloquer sans problèmes. Voici quelques éléments de réponse à l'épineuse question: "ICPM oui ou non?".

Le PING

Le ping peut éventuellement être une source de désagréments :

- Il contribue à révéler votre présence sur le Net
- Il peut servir à obtenir un déni de service (blocage de la machine)

Vous pouvez bloquer les "ping request" (signal 8) à l'entrée de votre machine, ça ne prête pas à conséquences.

3 ICMP : http://christian.caleca.free.fr/tcpip/les_protocoles.htm#icmp

Hôte inaccessible

C'est le signal 3. Il sert à indiquer que l'hôte que l'on cherche à joindre ne répond pas. C'est certainement le signal le plus utile pour les clients du Net.

Il ne faut pas bloquer ce signal, du moins en entrée, faute de quoi les couches supérieures du protocole ne pourront pas être informées et ne réagiront pas en conséquence.

Par ailleurs, ce signal intervient dans la découverte du MTU (Maximum Transfert Unit). C'est la taille la plus grosse qu'un paquet peut prendre avant de devoir subir une fragmentation. Si ce processus est mis en oeuvre et que le signal 3 est bloqué, les paquets envoyés risquent d'être systématiquement trop grands et donc systématiquement fragmentés (voire même rejetés parfois). Dans ce cas, les performances de la connexion risquent de devenir déplorables.

Ce signal est-il par ailleurs dangereux ? Pas à ma connaissance.

Horodatage

Ce signal peut donner des indications sur le fuseau horaire sur lequel vous vous trouvez. Son utilisation est assez similaire au ping. Comme vous ne voulez pas forcément donner ce genre d'information à un éventuel pirate, bloquez le (signal 13) A priori, ça ne devrait pas perturber le bon fonctionnement de votre connexion.

TTL Expiré

Information aussi intéressante que "hôte inaccessible". Ce signal (signal 11) est utilisé dans la commande "traceroute".

Il n'est pas utile, voire néfaste de bloquer ce signal.

Redirection nécessaire

Ce signal n'est en principe utile que pour les routeurs. Il peut servir à manipuler à distance la table des routes (commande "route print" sous WIndows, ou "route" sous Linux).

Du fait de ces risques, il vaut mieux le bloquer.

En ce qui concerne les autres signaux, je n'ai trouvé aucune information indiquant s'il valait mieux les bloquer ou non. A mon sens, le signal 17 (requête de masque de réseau) ne perd rien à être bloqué. A essayer pour voir.

Etude de cas : analyse de cas "classiques"

Commençons par voir ce qu'il se passe sur une machine "propre", c'est-à-dire non infectée par des chevaux de Troie, ou toute application destinée à en prendre le contrôle à distance.

Un poste "classique" sous Windows 98

L'expérience est tentée sur deux postes Windows 98, l'un avec le partage de fichiers activé et l'autre non. Nous faisons un scan de ports TCP et UDP avec nmap depuis un poste Linux situé sur le réseau :

- Ports TCP :

```
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
 Interesting ports on michele.maison.mrs (192.168.0.2):
(The 1522 ports scanned but not shown below are in state: closed)
Port       State       Service
139/tcp    open        netbios-ssn


TCP Sequence Prediction: Class=trivial time dependency
                         Difficulty=2 (Trivial joke)
Remote operating system guess: Windows NT4 / Win95 / Win98


Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

- Ports UDP :

```
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
 Warning:  No TCP ports found open on this machine,
OS detection will be MUCH less reliable
Interesting ports on michele.maison.mrs (192.168.0.2):
(The 1446 ports scanned but not shown below are in state: closed)
Port       State       Service
137/udp    open        netbios-ns
138/udp    open        netbios-dgm


Too many fingerprints match this host for me to give an accurate OS guess
Nmap run completed -- 1 IP address (1 host up) scanned in 8 seconds
```

Ce qu'il est intéressant de constater au premier abord, c'est que les mêmes ports sont ouverts, que l'on ait activé le partage des fichiers ou non. La seule chose qui diffère, c'est que le poste sur lequel le partage n'a pas été activé n'est pas visible dans le voisinage réseau.

Les seuls ports ouverts sont ceux utilisés par NetBIOS.

Mon poste de travail Windows 2000

Prenons un autre exemple un peu plus compliqué, mon poste de travail sous Windows 2000. (Je vous rappelle qu'il n'est pas directement connecté à l'Internet, j'ai une passerelle Linux entre les deux :-)

Ce poste est considéré comme une station de travail et en aucun cas comme un serveur. Il ne doit donc théoriquement pas y avoir de ports à l'écoute sur l'Internet.

Sur ce poste, j'utilise Frontpage 2000, j'ai donc un serveur web personnel (proposé par Windows

2000). Voici ce que donne un scan de ports TCP avec nmap depuis ma passerelle Linux :

```
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
 Interesting ports on chris.maison.mrs (192.168.0.10):
(The 1517 ports scanned but not shown below are in state: closed)
Port       State       Service
80/tcp     open        http
135/tcp    open        loc-srv
139/tcp    open        netbios-ssn
443/tcp    open        https
445/tcp    open        microsoft-ds

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=14803 (Worthy challenge)
Remote operating system guess: Windows 2000 RC1 through final release


Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

Comme vous le constatez, il y a 5 ports ouverts en écoute sur cette machine. Parmi ceux-ci, il y en a un bon nombre qui présentent des dangers.

Le port 80

C'est normal, PWS est actif (Personal Web Server). Ce serveur HTTP est utile pour la composition de sites avec Frontpage, pour la mise à disposition de documents sur le réseau privé, en revanche, il pourrait être dangereux de le laisser visible sur l'Internet...

Le port 135

Celui-ci doit présenter quelques dangers...

C'est le serveur RPC (Remote Procedure Call), c'est-à-dire le mécanisme qui permet à distance de déclencher l'exécution de procédures sur ma machine (par un administrateur uniquement). Il est clair que ce port ne doit pas être accessible depuis l'Internet.

Le port 139

Ah, celui là est bien connu! C'est un des mécanismes de service de noms NetBIOS (le voisinage réseau). Absolument rien à faire sur l'Internet...

Le port 443

HTTP "sécurisé" (HTTPS). Ouvert également par PWS.

Le port 445

Celui-ci, c'est une originalité de Windows 2000. Pour le service de noms, Microsoft a toujours utilisé son système WINS, basé sur NetBIOS. Depuis Windows 2000, il existe également un service de noms basé sur un DNS dynamique, qui n'utilise pas NetBIOS. Ce port est ouvert pour ce nouveau service et ne devrait se rencontrer que sur les machines Windows 2000 et suivants.

Passons maintenant à un scan de ports UDP :

```
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
 Warning:  No TCP ports found open on this machine,
OS detection will be MUCH less reliable
```

```
Interesting ports on chris.maison.mrs (192.168.0.10):
(The 1442 ports scanned but not shown below are in state: closed)
Port       State       Service
135/udp    open        loc-srv
137/udp    open        netbios-ns
138/udp    open        netbios-dgm
445/udp    open        microsoft-ds
500/udp    open        isakmp
3456/udp   open        vat


Too many fingerprints match this host for me to give an accurate OS guess
Nmap run completed -- 1 IP address (1 host up) scanned in 9 seconds
```

Nous avons déjà vu le port 135. Les ports 137 et 138 sont des services NetBIOS toujours pour la résolution des noms et les ouvertures de sessions. Nous avons également déjà rencontré le port 445, spécifique à Windows 2000.

Le port 500 est utilisé par HTTPS, pour la négociation de clés de cryptage. Encore un port ouvert par PWS.

Conclusions

Il est clair qu'il y a toujours quelques ports ouverts sur un hôte. Si les ports UDP ne présentent pas trop de dangers encore que...), les ports TCP sont plus inquiétants.

Par ailleurs, certaines machines de marque comme Compaq ou Hewlett Packard installent des dispositifs d'administration distante qui ouvrent également des ports à l'écoute et l'utilisateur ne le sait pas forcément.

Il importe donc de savoir avec le plus de précision possible qu'est ce qui est installé sur sa machine, volontairement ou involontairement.

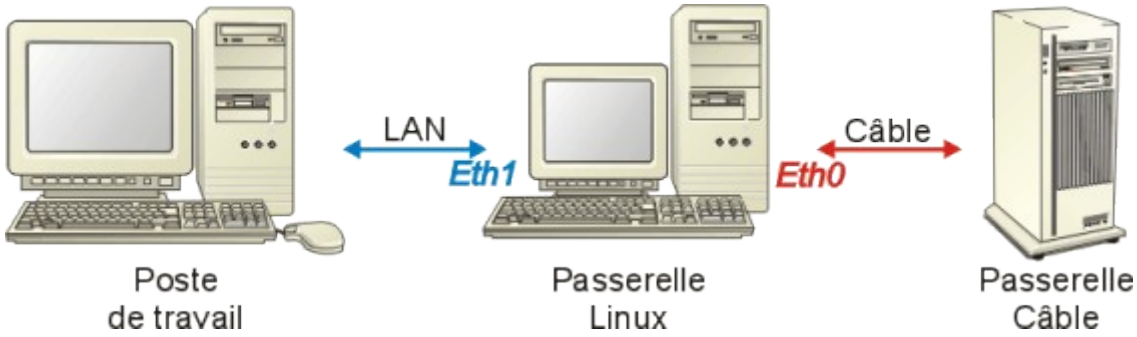
Contrôles possibles

Vérifications "à la main"

Il existe un certain nombre de commandes en ligne qui permettent de vérifier l'état de votre machine vis-à-vis du réseau. Nous allons essayer d'en voir quelques unes et ce que l'on peut en tirer comme informations.

Rappel sur la configuration de test

Pour mieux comprendre la suite, rappelons la configuration avec laquelle ces tests vont être effectués :



Le poste de travail est sous Windows 2000, il s'appelle "pchris". Il est connecté à la passerelle Linux via eth1.

La passerelle Linux est connectée à l'Internet via eth0, elle s'appelle "gateway1". Cette passerelle fait tourner un serveur SAMBA, qui permet d'ajouter les composants NetBIOS pour pouvoir partager des partitions Linux sur un réseau Microsoft. Ce n'est pas une solution exempte de dangers pour la sécurité de la passerelle, mais il est simple de filtrer les ports 137 à139. Par ailleurs, SAMBA permet de définir la plage d'adresses autorisées à accéder aux ressources, ainsi que les interfaces réseau à utiliser. Ce sont des protections supplémentaires à ne pas négliger.

La commande "netstat"

La commande "netstat'" est bien instructive, même si elle n'est pas toujours très lisible. Elle affiche les statistiques de protocole et les connexions réseau TCP/IP en cours.

```
E:\>netstat

Connexions actives

Proto Adresse locale Adresse distante Etat
TCP pchris:1031 gateway1.maison.mrs:netbios-ssn ESTABLISHED
```

Tout à fait normal. C'est la session réseau NetBIOS qui est à l'œuvre. En effet, dans mon voisinage réseau, je vois la passerelle.

Une variante de la commande :

```
E:\>netstat -n

Connexions actives

Proto Adresse locale Adresse distante Etat
TCP 192.168.0.10:1031 192.168.0.250:139 ESTABLISHED
```

Elle donne les mêmes informations, mais sans résoudre ni les adresses IP ni les ports.

Si j'arrête le serveur SAMBA sur la passerelle, cette connexion va disparaître, mais je ne verrai plus gateway1 dans mon voisinage réseau.

Attention !

La liste était simple parce que je ne faisais rien de spécial sur l'Internet. Voici la liste si j'ouvre Internet Explorer sur http://www.altavista.fr :

```
E:\>netstat

Connexions actives

Proto Adresse locale Adresse distante Etat
TCP pchris:1031 gateway1.maison.mrs:netbios-ssn ESTABLISHED
TCP pchris:1296 gateway1.maison.mrs:5901 ESTABLISHED
TCP pchris:1304 212.187.226.53:http TIME_WAIT
TCP pchris:1311 195.154.216.235:http TIME_WAIT
TCP pchris:1318 195.154.216.235:http TIME_WAIT
TCP pchris:1327 212.187.226.53:http TIME_WAIT
TCP pchris:1331 195.154.216.235:http TIME_WAIT
TCP pchris:1333 a213-56-194-51.deploy.akamaitechnologies.com:http ESTABLISHED
TCP pchris:1334 a213-56-194-51.deploy.akamaitechnologies.com:http ESTABLISHED
TCP pchris:1335 212.187.226.41:http ESTABLISHED
TCP pchris:1338 195.154.216.235:http TIME_WAIT
```

Il convient donc d'être prudent dans l'interprétation des résultats de cette commande qui liste toutes les connexions existant à un instant donné. Pour détecter des "connexions indélicates", il faut être certain que votre poste n'a aucune activité "normale" sur l'Internet (Pas de navigateur ouvert, pas de client de messagerie etc.).

La commande "netstat" dispose de quelques options, voici pour ceux qui voudraient un peu

s'amuser avec la documentation extraite de Windows 2000 (je ne garantis pas qu'elle soit entièrement compatible avec Windows 98).

Netstat

Affiche les statistiques de protocole et les connexions réseau TCP/IP en cours. Cette commande est disponible uniquement si le protocole TCP/IP est installé.

netstat [**-a**] [**-e**] [**-n**] [**-s**] [**-p** *protocole*] [**-r**] [*intervalle*] **Paramètres**	
-a	Affiche toutes les connexions et les ports d'écoute. Les connexions serveur ne sont en principe pas affichées.
-e	Affiche des statistiques relatives à Ethernet. Ce paramètre peut être combiné avec l'option **-s**.
-n	Affiche les adresses et numéros de ports sous forme numérique (au lieu de tenter des recherches par nom).
-s	Affiche les statistiques des protocoles respectifs. Par défaut, les statistiques de TCP, UDP, ICMP et IP sont affichées. L'option **-p** peut être utilisée pour spécifier un sous-ensemble des protocoles par défaut.
-p *protocole*	Affiche les connexions du protocole spécifié par le paramètre *protocole* ; ce paramètre peut avoir pour valeur **tcp** ou **udp**. Quand il est utilisé avec l'option **-s** pour afficher des statistiques par protocole, *protocol* peut prendre la valeur **tcp**, **udp**, **icmp** ou **ip**.
-r	Affiche le contenu de la table de routage.
intervalle	Affiche les statistiques sélectionnées de manière répétée avec un *intervalle* (en secondes) entre chaque occurrence. Appuyez sur CTRL+C pour interrompre l'affichage des statistiques. Si ce paramètre est omis, **netstat** n'imprime qu'une seule fois les informations de configuration.

La commande "arp"

Affiche et modifie les tables de conversion des adresses physiques IP employées par le protocole ARP (Address Resolution Protocol). Il s'agit ici d'une conversion au niveau du sous réseau. Le protocole ARP permet, dans un sous réseau, d'établir une relation entre adresse IP et adresse MAC, celle qui est employée par la couche d'accès réseau pour acheminer les données.

Autrement dit, si quelqu'un tente de se connecter sur votre machine alors qu'il est dans le même sous réseau que vous (un autre abonné câble de votre branche), vous le verrez. Mais si la tentative vient d'un autre réseau, vous ne verrez qu'une connexion à la passerelle.

Un exemple

Test de la commande "arp" sur le poste de travail :

```
E:\>arp -a

Interface : 192.168.0.10 on Interface 0x1000003
Adresse Internet Adresse physique Type
```

```
192.168.0.250 00-20-18-61-90-e3 dynamique
```

192.168.0.250, c'est l'adresse de ma passerelle sur eth1.

Même commande sur la passerelle Linux (pour une fois que l'on a les mêmes outils dans les deux environnements) :

```
[root@gateway1 chris]# arp -a
ca-ol-marseille-9-1.abo.wanadoo.fr (213.56.56.1) at 00:D0:79:72:5C:00 [ether] on eth0
chris.maison.mrs (192.168.0.10) at 00:20:18:B9:49:37 [ether] on eth1
```

J'ai deux connexions :

- La première sur 213.56.56.1 par eth0, c'est normal, c'est la passerelle par défaut que le client DHCP a récupéré.
- La seconde sur 192.168.0.10 par eth1, c'est encore normal, c'est mon poste de travail.

Note :

Il est possible que la commande vous réponde :

```
E:\>arp -a
Aucune entrée ARP trouvée
```

Ce n'est pas alarmant, si vous n'avez pas utilisé la connexion depuis un certain temps, la table ARP s'est vidée. Ce qui éventuellement peut être plus dérangeant, c'est lorsque votre table ARP indique des connexions multiples et pas forcément faciles à justifier:

Il vous faudra alors essayer d'identifier ces connexions en fonction de ce que vous êtes en train de faire sur votre machine.

Voici les diverses options de la commande "arp" :

Arp

Affiche et modifie les tables de conversion des adresses physiques IP (Ethernet ou anneau à jeton) employées par le protocole ARP (Address Resolution Protocol). Cette commande est disponible uniquement si le protocole TCP/IP est installé.

arp -a [*adr_inet*] [**-N** [*adr_si*]]
arp -d *adr_inet* [*adr_si*]
arp -s *adr_inet adr_ether* [*adr_si*]

Paramètres

-a	Affiche les entrées ARP en cours en interrogeant TCP/IP. Si *adr_inet* est spécifié, seules les adresses physiques et IP du système spécifié apparaissent.
-g	Identique à **-a**.
adr_inet	Spécifie une adresse IP en notation décimale pointée.
-N	Affiche les entrées ARP pour l'interface réseau spécifiée par *adr_si*.

adr_si	Spécifie, le cas échéant, l'adresse IP de l'interface dont la table de conversion des adresses doit être modifiée. Si elle n'est pas spécifiée, la modification est appliquée à la première interface rencontrée.
-d	Supprime l'entrée spécifiée par *adr_inet*.
-s	Ajoute une entrée dans la mémoire cache ARP pour associer l'adresse IP *adr_inet* à l'adresse physique *adr_ether*. L'adresse physique se compose de six octets hexadécimaux séparés par des tirets. L'adresse IP est spécifiée en notation décimale pointée. L'entrée est permanente, c'est-à-dire qu'elle est automatiquement supprimée du cache à l'expiration de la temporisation.
adr_ether	Spécifie une adresse physique.

Les outils d'audit

Sous Linux, il existe une foule d'outils permettant de tracer le trafic réseau. En général, ces outils sont capables d'afficher ou d'enregistrer (ou les deux) tout ce qui est visible sur une interface réseau, concernant les protocoles TCP, UDP, ICMP et même ARP.

Brutalement, ça ressemble un peu à un sniffer, à part que les trames sont identifiées, mais pas visualisées en entier et que, normalement, seules les trames destinées à l'hôte sont analysées (pas de mode "promiscuité").

Ces outils nécessitent la mise en place de règles de filtrage, sans quoi, ils enregistrent absolument tout ce qui entre depuis le réseau concerné.

Ce ne sont pas à proprement parler des outils de protection, mais ils permettent de contrôler le trafic et constituent un excellent moyen d'apprentissage. Bien configurés, ils peuvent servir d'alarme en cas d'activité jugée suspecte.

Parmi les plus connus sous Linux, il y a ippl, iplog , iptraf et bien sûr Snort et TCPDump.

L'audit ne se résume cependant pas aux traces générées par ce genre d'outils. Les systèmes d'exploitation "sérieux" (Windows NT, Windows 2000, Linux...) construisent des journaux d'audit du système qui permettent de savoir qui s'est connecté sur la machine, quel service a été démarré, utilisé, arrêté... Autant d'informations qu'il ne faut pas négliger de consulter régulièrement.

Protections

Savoir ce qui est installé

La première règle est bien entendu de savoir exactement quels sont les services installés sur sa machine et de n'y laisser que ce qui est strictement nécessaire. Cette méthode, surtout dans le cas d'un réseau local connecté à l'Internet par une passerelle est toutefois assez pénalisante. On peut souhaiter disposer de quelques services sur l'hôte qui sert de passerelle.

Bien entendu, la solution la plus sûre consiste à installer une passerelle qui ne fera que son travail de passerelle et de firewall et d'installer par ailleurs sur le réseau privé un serveur pour les divers services souhaités. Ça augmente tout de même le nombre de machines et la facture EDF. Ça ne résoudra pas non plus certains problèmes pour les entreprises qui souhaitent accéder à certaines de leurs ressources depuis l'extérieur, mais le cas de figure dépasse largement le propos de cet exposé.

Construire une barrière

État des lieux

Une solution de protection consiste à interdire l'accès aux ports inutiles côté Internet. Sur Linux Mandrake 7.x (plus généralement avec un noyau 2.2.x bien compilé), ceci peut se faire avec IPChains. Les noyaux 2.4.x, bien que supportant IPChains, gagneront à exploiter plutôt IPTables, nettement plus évolué.

Pour fixer les esprits, donnons un exemple.

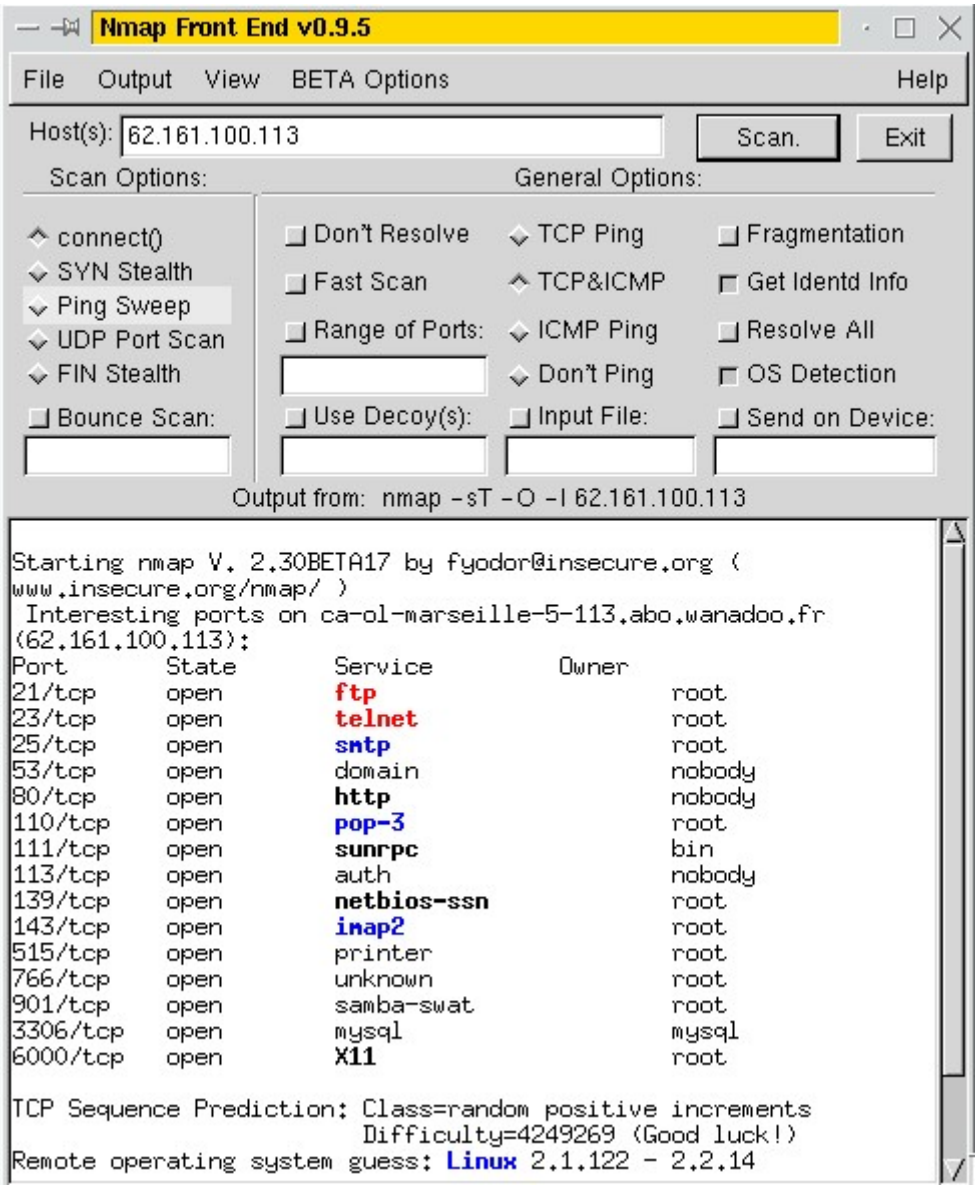
Soit une machine Linux servant de passerelle sur l'Internet. Comme on aime bien jouer avec les diverses applications fournies, on y a installé :

- SAMBA, pour communiquer avec le réseau Microsoft,
- APACHE, pour tester ses pages web sur un serveur classique de l'Internet,
- WU-FTPD, serveur FTP pour pouvoir charger ses pages HTML avec les outils de FrontPage,
- BIND, pour avoir son propre DNS,
- POSTFIX, pour avoir son propre serveur SMTP,
- VNC SERVER pour ouvrir des sessions X depuis les postes Microsoft du réseau privé,
- Et j'en passe...

Et comme on ne veut pas s'embêter, la règle par défaut sur INPUT est ACCEPT.

Croyez-vous que ce soit prudent? Pas du tout bien entendu. Sur une machine exposée à l'Internet, moins on installe de serveurs, mieux ça vaut. Examinons ce que verrait un pirate qui ferait un "scan" de cette machine avec l'un des meilleurs outils dans le genre: nmap (inclus dans les distributions Mandrake).

Démonstration :



De quoi vraiment donner envie de s'y intéresser de plus près !

- 21, c'est WU-FTPD
- 23, Telnet, ah! nous verrons ça...
- 25 C'est Postfix, faudra voir si ce ne serait pas un "open relay" :-)
- 53 Tiens, il y a un DNS, intéressant.
- 80, sans doute apache.
- 139, Ce monsieur a installé SAMBA.

Bon, ça suffit comme ça pour l'instant. Exploitons un peu le sujet...

Telnet, c'est intéressant :

```
Welcome to gateway2.maison.mrs
Linux Mandrake release 7.1 (helium)
Kernel 2.2.16-9mdk on an i586
login:
```

Voilà, une Mandrake 7.1 avec un kernel 2.2.16-9 et la machine s'appelle gateway2.maison.mrs. Ça c'est intéressant. Allons faire un tour sur le DNS du monsieur...

```
E:\>nslookup
Serveur par défaut :  <peu importe>
Address:  <peu importe>

> server 62.161.100.113
Serveur par défaut :  ca-ol-marseille-5-113.abo.wanadoo.fr
Address:  62.161.100.113
```

```
> set q=any
> ls maison.mrs
[ca-ol-marseille-5-113.abo.wanadoo.fr]
 maison.mrs.                    NS     server = gateway2.maison.mrs
 gateway2                       A      192.168.0.253
 remi                           A      192.168.0.12
 michele                        A      192.168.0.2
 chris                          A      192.168.0.10
 gateway1                       NS     server = 192.168.0.250
 daniel                         A      192.168.0.11
 gateway2                       NS     server = 192.168.0.253
```

Et hop! On sait tout du réseau de ce monsieur :-) (Même, si vous avez bien suivi, que ce monsieur dispose sans doute d'une seconde machine du même genre qui s'appelle gateway1).

Il faut dire que c'est quand même très mal, de laisser libre le transfert de zone sur un DNS. Mais j'ai trouvé cette gravissime lacune sur des sites très "officiels".

Reprenons le scénario, mais avec BIND correctement configuré. Ça donne ceci :

```
E:\>nslookup
Serveur par défaut : gateway1.maison.mrs
Address: 192.168.0.250

> server 62.161.100.113
Serveur par défaut : ca-ol-marseille-5-113.abo.wanadoo.fr
Address: 62.161.100.113

> set q=any
> ls maison.mrs.
ls: connect: No error
*** Impossible de fournir la liste du domaine maison.mrs.: Unspecified error
> pchris.maison.mrs
Serveur : ca-ol-marseille-5-113.abo.wanadoo.fr
Address: 62.161.100.113

*** ca-ol-marseille-5-113.abo.wanadoo.fr ne parvient pas à trouver pchris.maison.mrs :
 No response from server
```

C'est déjà mieux, au moins le DNS ne répond plus aux requêtes venant de l'Internet. Comment il faut faire? Dans /etc/named.conf, il faut utiliser les directives "allow-transfert", "allow-query" et même "listen-on" (cf. la doc. de BIND).

Cet exemple est juste donné pour bien montrer que la sécurité passe d'abord par une configuration correcte des serveurs installés...

Mais continuons l'investigation. Voyons le serveur FTP, un petit coup de telnet sur le port 21 :

```
220 gateway2.maison.mrs FTP server (Version wu-2.6.0(1) Wed Jun 28 23:51:34
EDT2000) ready.
```

Oui, c'est bien un wu-ftp, version 2.6.0. Faudra voir ce qu'il y a comme "exploits" là dessus. On va s'arrêter là, mais il y a pas mal d'investigations à faire sur un serveur FTP.

Allez, encore un telnet sur le port 25 :

```
220 gateway2.maison.mrs ESMTP Postfix (Postfix-19991231) (Linux-Mandrake)
```

C'est bien Postfix. (Là aussi, il y aurait encore beaucoup à faire).

Convaincu ? En très peu de temps, le pirate accumule une quantité intéressante d'informations sur votre équipement, autant d'informations qu'il pourra exploiter pour essayer de "casser" votre matériel.

Comme l'objectif de cet exposé n'est pas de faire un cours sur l'intrusion (encore que ce soit le meilleur moyen pour apprendre à mettre en place des parades), on va s'arrêter là.

La situation exposée est d'autant plus absurde, qu'avec IPTables, on peut déjà compliquer passablement le travail du pirate.

Que les utilisateurs de Windows n'abandonnent pas la lecture de ce qui suit. La démonstration se fait avec IPtables, mais le principe reste vrai quel que soit l'OS. Nous verrons plus loin les solutions proposées aux utilisateurs de produits Microsoft.

Construction de barrières

Disons d'abord ce que l'on veut faire en français. D'abord une simple passerelle avec masquage d'adresses du réseau privé :

- En entrée, on accepte tout par défaut (comme c'était dans l'exemple précédent),
- en sortie, on laisse tout passer aussi,
- au travers de la passerelle (le routage entre le Net et le réseau privé), on n'accepte rien, mais on va faire du masquage d'adresse pour tout ce qui vient du réseau privé et va vers le Net.

Ensuite, pour tout ce qui vient du Net, nous bloquerons en UDP comme en TCP les ports :

- 21 tcp, qui est le port de commande FTP,
- 23 tcp, qui est le port Telnet,
- 25 tcp, qui est le port SMTP,
- 110 tcp, qui est le port POP3,
- 111 tcp/udp qui est le port des "Remote Procedure Call",
- 135 à 139 tcp/udp, des ports utilisés par NetBIOS,
- 143 tcp, le port IMAP,
- 6000 à 6009 tcp, les ports utilisés pour le serveur graphique.

```
iptables -P INPUT ACCEPT
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

# Le MASQUERADE...
iptables -t nat -A POSTROUTING -s 192.168.0.0/255.255.255.0 -o ppp0 -j MASQUERADE

# et les interdictions :
iptables -A INPUT -p tcp --dport 21 -i ppp0 -j REJECT
iptables -A INPUT -p tcp --dport 23 -i ppp0 -j REJECT
iptables -A INPUT -p tcp --dport 25 -i ppp0 -j REJECT
iptables -A INPUT -p tcp --dport 110 -i ppp0 -j REJECT
iptables -A INPUT -p tcp --dport 111 -i ppp0 -j REJECT
iptables -A INPUT -p udp --dport 111 -i ppp0 -j REJECT
iptables -A INPUT -p tcp --dport 135:139 -i ppp0 -j REJECT
iptables -A INPUT -p udp --dport 135:139 -i ppp0 -j REJECT
iptables -A INPUT -p tcp --dport 143 -i ppp0 -j REJECT
iptables -A INPUT -p tcp --dport 6000:6009 -i ppp0 -j REJECT
```

Traduit en français, ça veut dire que l'on jette (REJECT) les paquets qui viennent de n'importe où pour aller n'importe où, s'ils ont le malheur de rentrer par ppp0 sur les ports 21, 23, 25, 110, 111, 139, 143 et de 6000 à 6009. C'est bien ce que nous voulions. Refaisons un scan :

```
Starting nmap V. 2.30BETA17 by fyodor@insecure.org ( www.insecure.org/nmap/ )
 Interesting ports on ca-ol-marseille-5-113.abo.wanadoo.fr (62.161.100.113):
(Ports scanned but not shown below are in state: filtered)
Port       State       Service         Owner
1/tcp      unfiltered  tcpmux
2/tcp      unfiltered  compressnet
3/tcp      unfiltered  compressnet
...
80/tcp     open        http                    nobody
...
113/tcp    open        auth                    nobody
...
515/tcp    open        printer                 root
...
...
3306/tcp   open        mysql                   mysql
...


TCP Sequence Prediction: Class=random positive increments
                         Difficulty=4917615 (Good luck!)
Remote operating system guess: Linux 2.1.122 - 2.2.14


Nmap run completed -- 1 IP address (1 host up) scanned in 220 seconds
```

nmap ne s'y trompe pas, il constate qu'il y a des règles de filtrage sur cette machine et essaye de trouver les ports non filtrés. Il va en trouver beaucoup, mais ça ne veut pas dire qu'ils sont ouverts. Ceux qu'il trouve ouverts sont ceux que l'on n'a pas filtrés.

Il est clair que l'on a déjà limité les problèmes. Mais on pourrait encore faire beaucoup mieux . L'objectif ici n'était que de montrer comment l'on peut établir des règles de filtrage de paquets, mais d'expliquer comment un port peut être bloqué par un firewall.

Dans la pratique, il sera probablement plus judicieux de tout interdire, puis de n'ouvrir que ce qui est nécessaire. Voyez à ce propos le chapitre sur Netfilter[4].

Conclusion

Ce type de protection passive offre déjà un bon niveau de sécurité, si l'on a convenablement analysé la configuration de sa machine et placé les bonnes règles. Il y aurait encore à faire sur cette machine, car si l'on a à peu près filtré les ports TCP, qui sont les plus dangereux parce qu'ils permettent un mode connecté, on n'a encore rien fait ni sur UDP, ni sur ICMP. Ces deux protocoles peuvent cependant créer des nuisances parce qu'ils peuvent être utilisés pour bloquer la machine.

Notez qu'il existe un site qui permet de construire des règles IPChains en fonction de critères de protection que l'on choisit dans des tables

http://www.linux-firewall-tools.com/linux/firewall/index.html

Cependant, il est intéressant de placer en plus quelques systèmes qui vont épier le trafic et prévenir, voire réagir, en cas d'activité suspecte avec les "loggers" et les firewalls actifs.

IPtables sait déjà "logger" les événements qui satisfont aux critères des chaînes, mais il est peut-être plus intéressant d'utiliser des outils spécifiques.

4 Netfilter : http://christian.caleca.free.fr/netfilter/

Les miradors du Net

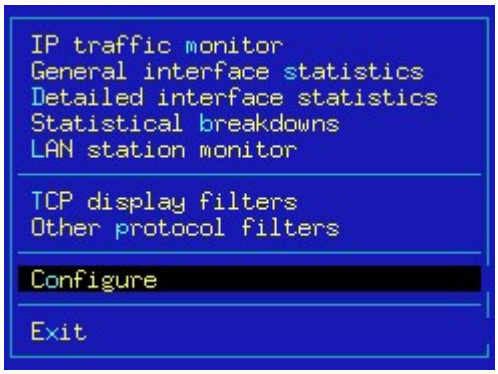
La plupart des attaques commencent par un scan des ports ouverts sur la cible. Des outils particuliers permettent de détecter ce scan, d'en identifier la source par son adresse IP et certains permettent même de monter un firewall "sur mesure" pour bloquer l'intrus. Le scanner n'aura même pas le temps de finir son travail et aura l'impression que la cible a disparu.

Les "loggers"

J'en ai plus ou moins testé trois sous Linux, il en existe beaucoup d'autres, chacun dispose, à mon sens, d'avantages et d'inconvénients. Les trois fonctionnent sur Mandrake 7.1 et 7.2 et se trouvent au format RPM.

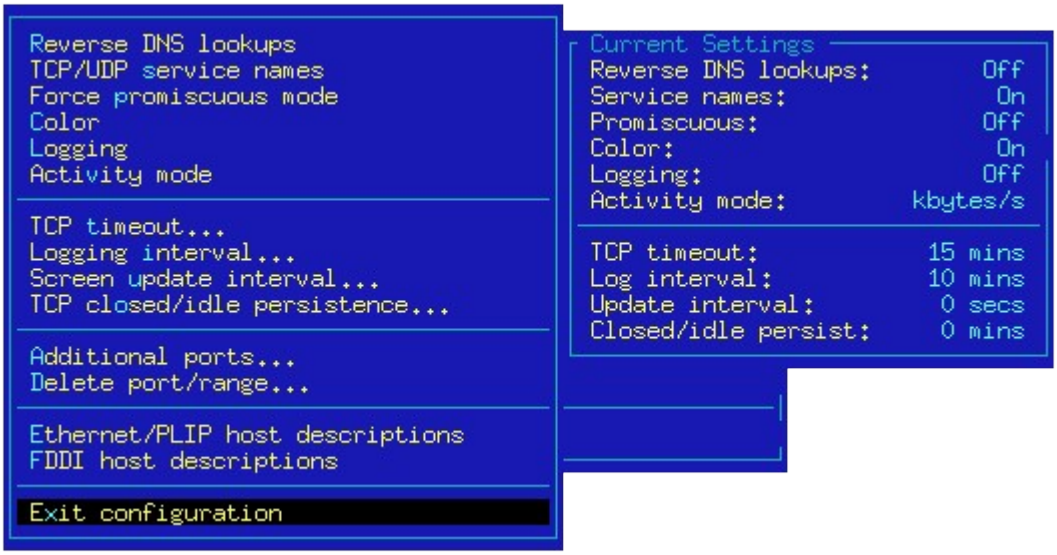
IPTRAF

Il ne figure plus dans la distribution 9.1, mais se trouve toujours au format rpm pour mandrake dans les contributions.

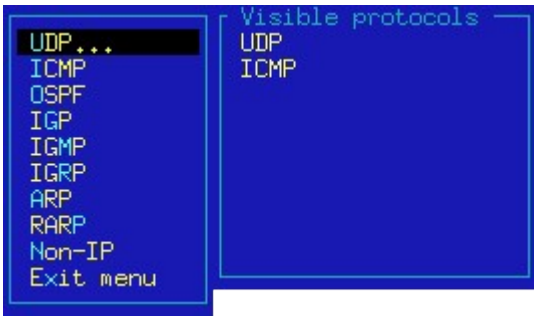


C'est peut-être le plus "convivial" mais pas forcément le plus paramétrable. Il fonctionne bien en mode texte mais présente quelques bugs d'affichage dans une console sous X, suivant la taille de la fenêtre.

Les "anciens" de MS DOS retrouveront avec nostalgie les menus arborescents en mode caractère...

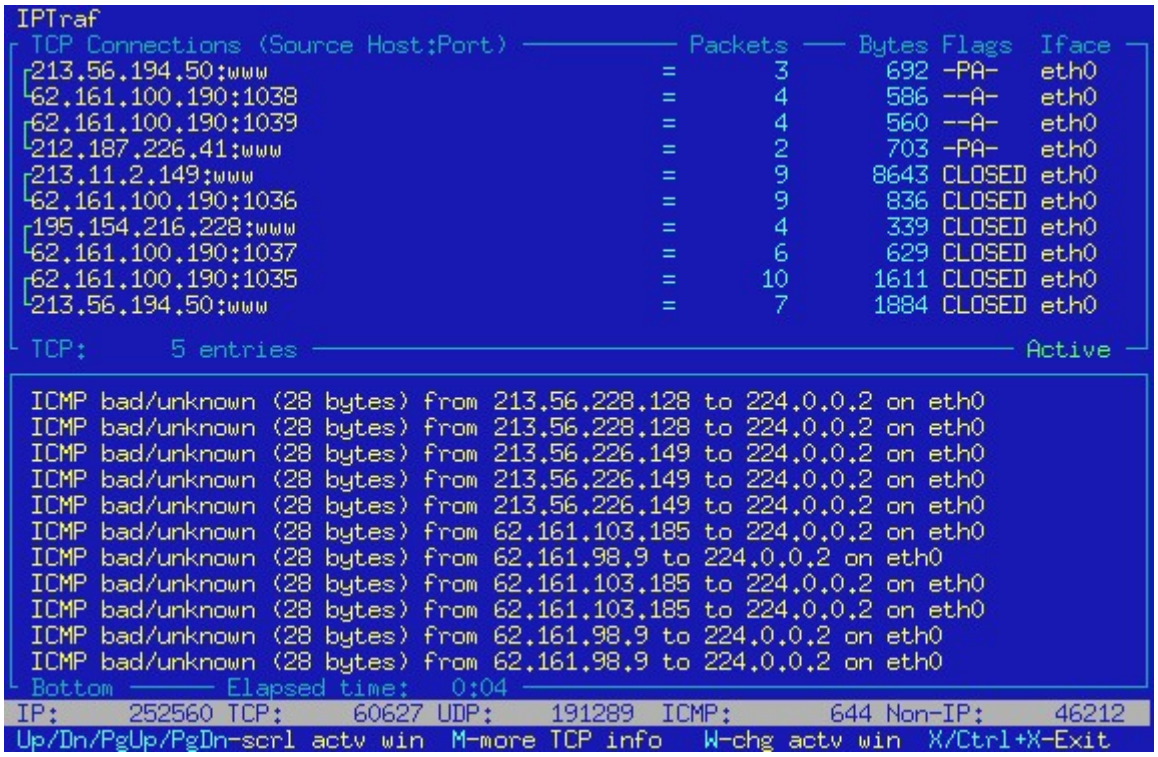


Ici, il est possible de choisir les protocoles à tracer. Malheureusement, s'il est possible de définir des filtres personnalisés pour UDP, l'option n'existe pas pour les autres protocoles.



Il est également possible de choisir l'interface que l'on souhaite tracer

Voici un exemple de trace. Si les connexions TCP sont affichées de manière très lisible, il n'en va malheureusement pas de même pour les autres protocoles.



IPPL

Celui-ci fonctionne de manière différente. Son but principal étant d'enregistrer la trace dans des fichiers, bien qu'il soit possible de diriger ses traces sur une console. Lui aussi, se trouve encore dans les contributions à la version 9.1 de Mandrake.

Sa configuration se fait par le fichier /etc/ippl.conf, je n'ai malheureusement pas trouvé le moyen de limiter son activité à l'interface choisie autrement qu'en donnant son adresse IP, ce qui n'est guère pratique lorsque l'on dispose d'une adresse dynamique.

IPLOG

Assez similaire à IPPL, celui-ci permet d'indiquer l'interface à logger par son nom (eth0 par exemple). Lui aussi se configure par un fichier /etc/iplog.rules. Il est toujours distribué dans la version 9.1.

Une simple manip.

- iplog est démarré :
  `iplog -L -i eth0`
  (-L pour diriger le flux dans la console)
- Le fichier iplog.rules ne contient que la ligne :
  `ignore udp from * sport 53`
  Ma machine ayant un DNS installé, les réponses des serveurs DNS sur le port 53 sont fréquentes, inutile de les tracer.

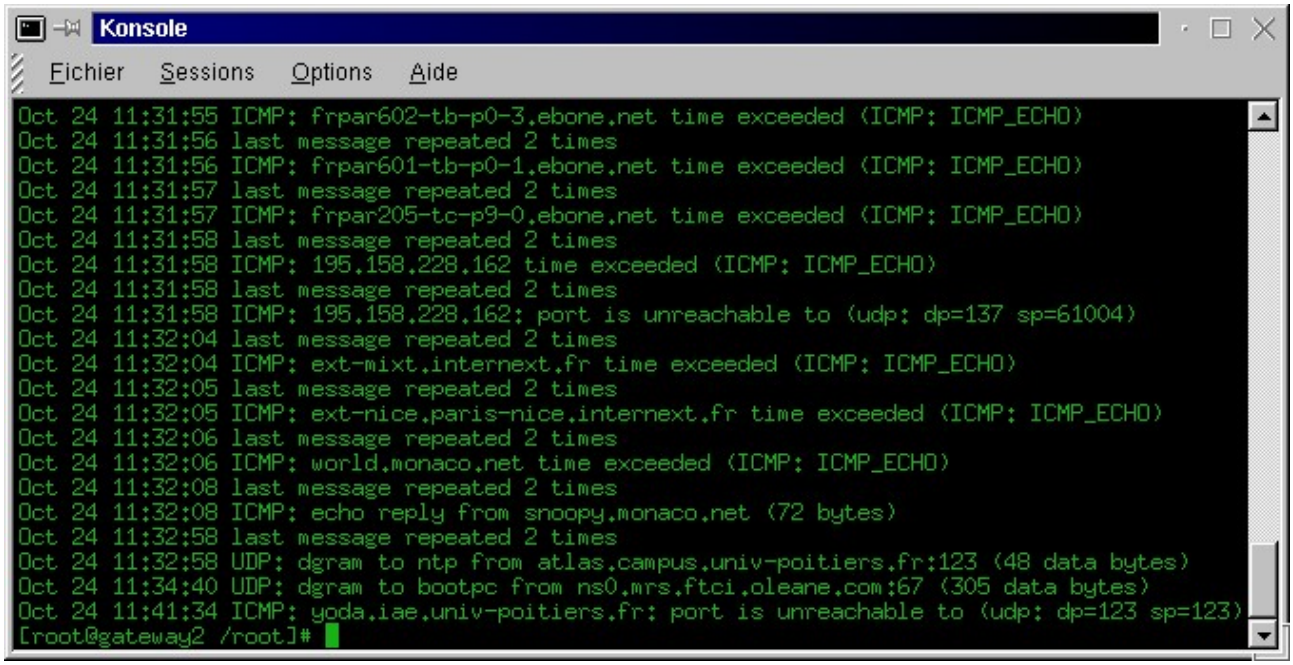
Où la paranoïa nous guette...

Tout va à peu près bien. Pour faire un test, je lance un "tracert" depuis ma station de travail :

```
Détermination de l'itinéraire vers www.eme-enseignement.fr [194.79.150.15]
avec un maximum de 30 sauts :

1 <10 ms <10 ms 10 ms gateway2.maison.mrs [192.168.0.253]
2 20 ms 20 ms 20 ms ca-ol-marseille-1-1.abo.wanadoo.fr [62.161.96.1]
3 20 ms 20 ms 30 ms LS-ATM6-0-0-300.Marseille.raei.francetelecom.net[194.250.158.89]
4 20 ms 30 ms 30 ms POS-6-0-0.NCMAR201.Marseille.raei.francetelecom.net [194.51.171.37]
5 20 ms 30 ms 30 ms P0-2.nrlyo101.Lyon.francetelecom.net [193.252.101.74]
6 30 ms 31 ms 30 ms P2-1.ntaub101.Aubervilliers.francetelecom.net [193.251.126.206]
7 30 ms 40 ms 50 ms 193.251.126.154
8 30 ms 40 ms 40 ms P0-0.BAGBB2.Bagnolet.opentransit.net [193.251.128.142]
9 30 ms 40 ms 40 ms P1-0.BOUBB2.Paris.opentransit.net [193.251.128.58]
10 30 ms 30 ms 40 ms P0-0.BOUBB1.Paris.opentransit.net [193.251.128.233]
11 40 ms 30 ms 40 ms frpar602-tb-p0-3.ebone.net [195.158.226.213]
12 30 ms 40 ms 30 ms frpar601-tb-p0-1.ebone.net [195.158.226.197]
13 30 ms 30 ms 40 ms frpar205-tc-p9-0.ebone.net [195.158.228.157]
14 40 ms 40 ms 30 ms 195.158.228.162
15 40 ms 40 ms 40 ms ext-mixt.internext.fr [195.5.253.2]
16 140 ms 150 ms 180 ms ext-nice.paris-nice.internext.fr [194.79.128.78]
17 150 ms 180 ms 130 ms world.monaco.net [194.79.150.1]
18 130 ms 100 ms 170 ms snoopy.monaco.net [194.79.150.15]
Itinéraire déterminé.
```

IPLOG se met alors à remplir la console...



Les "time exceeded (ICMP_ECHO)", c'est parfaitement normal, c'est le fonctionnement de la commande "traceroute" qui veut ça. Comparez la réponse du "tracert" sur mon poste avec les logs ICMP.

Mais il y a des choses plus délicates à expliquer...

- Ce qui concerne 195.158.228.162 par exemple: udp dp=137, sp=61004 ça ressemble à du NetBIOS (dp=137) qui passe le masquage d'adresses (port 61004). Mais cette adresse n'a pas de nom. Elle appartient à "ebone-backbone 4" dixit "sam spade"[5], outil d'investigation bien connu dans le monde Windows.
  Je ne sais pas expliquer cette trace.
- UDP datagram from atlas.campus.univ-poitiers.fr:123, ça, je sais. Je ne vous avais pas dit que ma machine Linux est à l'heure atomique grâce à xntpd. atlas.campus.univ-poitiers.fr est un serveur NTP (Network Time Protocol).
- UDP datagram from ns0.mrs.ftci.oleane.com:67, je sais aussi. C'est notre cher serveur DHCP/DNS. Le bail a été renouvelé.
- ICMP yoda.iae.univ-poitiers.fr:123 unreachable , c'est intéressant, parce que comme ça, je constate que l'un des serveurs NTP que j'essaye d'utiliser ne répond pas.

Snort

C'est probablement le plus puissant, avec TCPdump. Pas testés ici, ils sont nettement plus délicats à manipuler, mais ce sont les meilleurs.

Conclusions

Un outil de log comme iplog est un bon moyen pour contrôler le trafic. Il ne sait cependant tracer que ce qu'il lui est demandé (ce que **vous** lui demandez) et uniquement ce qu'il lui est demandé. Ça

5 Sam Spade : http://www.samspade.org/ssw/

veut dire deux choses importantes :

- Il tracera des événements qui n'en valent peut-être pas la peine. Si vous lui demandez de tracer tout trafic sur une interface, il le fera. Les logs de la journée risqueront peut-être de peser plusieurs Mo et seront inexploitables, autrement que par des filtres qu'il vous faudra écrire.
- Il ne tracera peut-être pas des événements qu'il aurait été important de ne pas rater.

Vous apprendrez certainement beaucoup de choses en passant du temps à essayer de maîtriser ce genre d'outils

Les Coupe feux actifs

Un coupe feu actif, en plus de surveiller les connexions entrantes est capable de détecter une activité réputée frauduleuse et y parer de manière dynamique. Il existe sous Linux au moins une application de ce type : "Portsentry".

Portsentry, le pompier du Net

Portsentry est capable de détecter un "scan" de ports TCP et UDP. La détection peut déclencher plusieurs types de parades :

- Un simple "log" via syslogd. Dans ce cas, Portsentry agit comme un simple "loggeur", un peu léger toutefois, puisqu'il ne sait pas traiter les activités ICMP.
- Une inscription du scanner dans "hosts.deny", pour qu'il soit pris en compte par le "TCP Wrapper".
- L'ajout d'une règle dans la chaîne "input" d'IPtables pour bloquer l'intrus.
- La modification de la route par défaut dans un "black hole" de manière à ce qu'une éventuelle réponse à l'intrus soit dirigée vers nulle part.
- Le déclenchement d'une commande externe pouvant être, par exemple, la désactivation pure et simple de l'interface réseau.

Bien entendu, ces parades sont configurables.

Juste un exemple

- Dans le rôle de l'intrus: 213.56.228.199. Une machine Linux équipée du scanner "nmap".
- Dans le rôle du défenseur, une autre machine Linux, équipée de portsentry. La configuration est à peu de choses près celle qui est donnée par défaut dans le paquetage rpm pour Mandrake. En cas de détection de scan, portsentry va réagir de la manière suivante :
  - Inscription de l'intrus dans /etc/hosts.deny"
  - Ecriture d' une règle de blocage dans IPChains (fonctionne aussi avec IPTables).

Avant l'attaque

Voyons un peu l'allure de la chaîne INPUT (Iptables) :

```
Chain INPUT (policy ACCEPT)
target  prot opt source    destination
REJECT  tcp  --  anywhere  anywhere  tcp dpt:telnet reject-with icmp-port-unreachable
REJECT  tcp  --  anywhere  anywhere  tcp dpt:smtp reject-with icmp-port-unreachable
```

Les protections sont très simplistes, seuls telnet et smtp sont filtrés.

Portsentry est démarré avec la commande "portsentry -atcp" (Entamer ici une étude détaillée de portsentry nous mènerait trop loin.).

Le démarrage de portsentry ajoute ces lignes au fichier /var/log/messages :

```
adminalert: Psionic PortSentry  is starting.
adminalert: Advanced mode will monitor first 1023 ports
adminalert: Advanced mode will manually exclude port: 113
adminalert: Advanced mode will manually exclude port: 139
adminalert: Advanced Stealth scan detection mode activated. Ignored TCP port: 21
adminalert: Advanced Stealth scan detection mode activated. Ignored TCP port: 23
adminalert: Advanced Stealth scan detection mode activated. Ignored TCP port: 25
adminalert: Advanced Stealth scan detection mode activated. Ignored TCP port: 53
adminalert: Advanced Stealth scan detection mode activated. Ignored TCP port: 80
adminalert: Advanced Stealth scan detection mode activated. Ignored TCP port: 110
adminalert: Advanced Stealth scan detection mode activated. Ignored TCP port: 111
adminalert: Advanced Stealth scan detection mode activated. Ignored TCP port: 113
adminalert: Advanced Stealth scan detection mode activated. Ignored TCP port: 139
adminalert: Advanced Stealth scan detection mode activated. Ignored TCP port: 143
adminalert: Advanced Stealth scan detection mode activated. Ignored TCP port: 515
adminalert: Advanced Stealth scan detection mode activated. Ignored TCP port: 820
adminalert: Advanced Stealth scan detection mode activated. Ignored TCP port: 901
adminalert: Advanced Stealth scan detection mode activated. Ignored TCP port: 113
adminalert: Advanced Stealth scan detection mode activated. Ignored TCP port: 139
adminalert: PortSentry is now active and listening.
```

L'attaque a lieu...

Le scan est démarré avec nmap sur l'hôte "pirate". Portsentry s'en rend compte et effectue les opérations suivantes :

- Les lignes qui suivent sont ajoutées au journal /var/log/messages

```
attackalert: SYN/Normal scan from host: ca-ol-marseille-21-199.abo.wanadoo.fr/213.56.228.199
to TCP port:52
attackalert: Host 213.56.228.199 has been blocked via wrappers with string: "ALL:
213.56.228.199"
attackalert: Host 213.56.228.199 has been blocked via dropped route
             using command: "/sbin/ipchains -I input -s 213.56.228.199 -j DENY"
attackalert: SYN/Normal scan from host: ca-ol-marseille-21-199.abo.wanadoo.fr/213.56.228.199
to TCP port:175
attackalert: Host: ca-ol-marseille-21-199.abo.wanadoo.fr/213.56.228.199 is already blocked
Ignoring
attackalert: SYN/Normal scan from host: ca-ol-marseille-21-199.abo.wanadoo.fr/213.56.228.199
to TCP port:32
attackalert: Host: ca-ol-marseille-21-199.abo.wanadoo.fr/213.56.228.199 is already blocked
Ignoring
attackalert: SYN/Normal scan from host: ca-ol-marseille-21-199.abo.wanadoo.fr/213.56.228.199
to TCP port:621
attackalert: Host: ca-ol-marseille-21-199.abo.wanadoo.fr/213.56.228.199 is already blocked
Ignoring
attackalert: SYN/Normal scan from host: ca-ol-marseille-21-199.abo.wanadoo.fr/213.56.228.199
to TCP port:312
attackalert: Host: ca-ol-marseille-21-199.abo.wanadoo.fr/213.56.228.199 is already blocked
Ignoring
```

- La règle suivante est ajoutée dans la chaîne INPUT d'IPChains :

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP       all  --  213.56.228.199  anywhere
   *** L'attaquant est bloqué sur tous les ports, sur tous les protocoles,
   *** quelle que soit la destination
REJECT       tcp   --   anywhere             anywhere    tcp dpt:telnet reject-with icmp-port-
unreachable
REJECT     tcp  --  anywhere        anywhere   tcp dpt:smtp reject-with icmp-port-unreachable
```

- Le fichier /etc/hosts.deny est modifié de la manière suivante :

```
#
# hosts.deny    This file describes the names of the hosts which are
#               *not* allowed to use the local INET services, as decided
#               by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind you that
# the new secure portmap uses hosts.deny and hosts.allow.  In particular
# you should know that NFS uses portmap!
#


ALL: 213.56.228.199
```

Bilan

L'attaquant n'aura même pas le temps de finir son scan. il aura l'impression que la cible n'existe plus, tout simplement.

N'est-ce pas joli? Attention toutefois, si la modification de /etc/hosts.deny n'est pas volatile (écriture dans le fichier), il n'en va pas de même pour IPChains. Sans précautions particulières, si la machine est arrêtée, la règle sera perdue. Il existe cependant une parade grâce, par exemple, à la possibilité d'exécuter une commande externe en cas de détection d'attaque. Il suffit de sauvegarder la règle INPUT pour pouvoir la recharger au prochain démarrage. Ceci dit, les attaques venant le plus souvent d'adresses dynamiques, il n'est pas forcément nécessaire de rendre la règle définitive. Par ailleurs, si l'on agit sur IPChains, il n'est pas nécessaire de conserver l'option de modification de /etc/hosts.deny.

Notez tout de même que nous n'avons pas couvert les risques émanant :

- D'un scan UDP.
  C'est généralement moins dangereux, mais pas à négliger tout de même. Portsentry peut également être démarré pour surveiller les ports UDP, mais le fonctionnement est plus délicat et peut provoquer des réactions parasites.
- D'une attaque ICMP
  Portsentry ne sait rien faire contre ça. Il ne vous reste de ce côté qu'à écrire des règles correctes.
- **D'une attaque sans qu'il y ait de scan préalable**

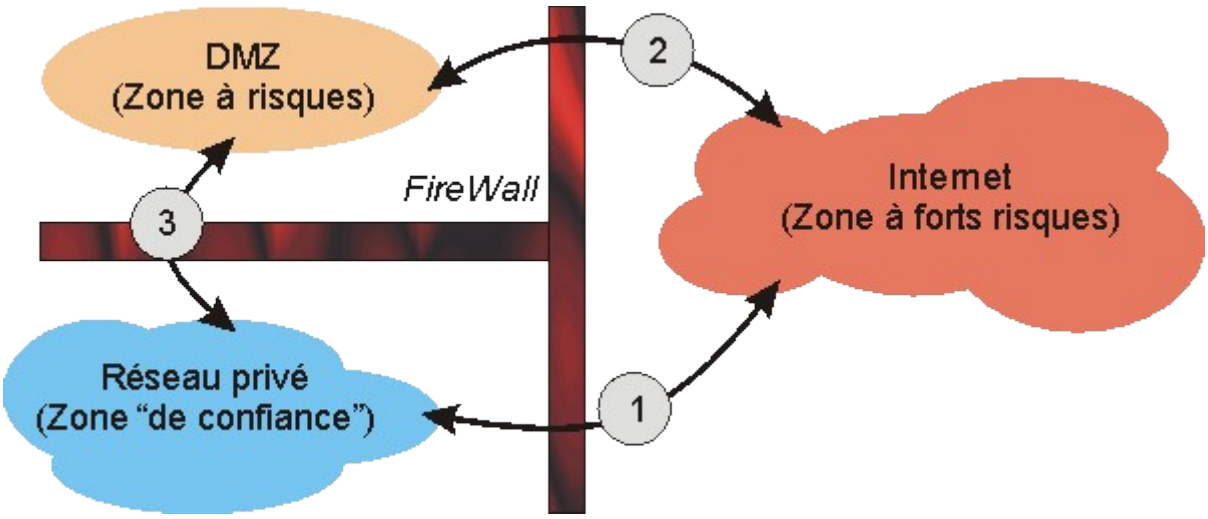
Portsentry n'est donc hélas pas un outil miraculeux, il n'en existe d'ailleurs pas.

Malheureusement, cet outil ne semble plus être fourni avec Mandrake 9.1

Les Firewalls

Dans la page précédente, nous avons fait du "firewalling" sans le savoir. Pour donner quelques indications de plus, il est nécessaire de se pencher d'un peu plus près sur les diverses méthodes de construction d'un "firewall".

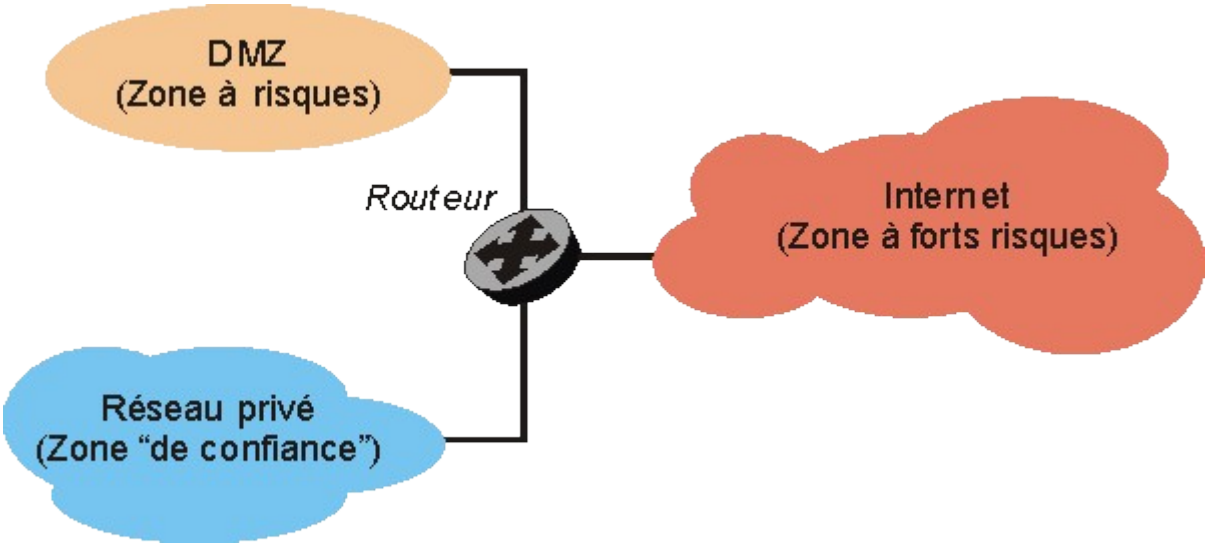
D'abord, c'est quoi, un "firewall" ?



Typiquement, une installation complète contient :

- Un réseau privé, dont on considère (souvent à tort) qu'il ne sera pas utilisé pour attaquer notre système informatique. Dans cette zone, il n'y a que des clients du réseau et des serveurs qui sont inaccessibles depuis l'Internet. Normalement, aucune connexion, au sens TCP du terme, aucun échange, au sens UDP du terme, ne peuvent être initiés depuis le Net vers cette zone.
- Une "DMZ" (Zone DéMilitarisée), qui contient des serveurs accessibles depuis le Net et depuis le réseau privé. Comme ils sont accessibles depuis le Net, ils risquent des attaques. Ceci induit deux conséquences :
    - Il faut étroitement contrôler ce que l'on peut faire dessus depuis le Net, pour éviter qu'ils se fassent "casser" trop facilement,
    - Il faut s'assurer qu'ils ne peuvent pas accéder aux serveurs de la zone privée, de manière à ce que si un pirate arrivait à en prendre possession, il ne puisse directement accéder au reste du réseau.

Les trois types de communications marquées 1,2 et 3 sur l'illustration seront donc soumis à des règles de passage différentes. Le dispositif qui va permettre d'établir ces règles de passages s'appelle un firewall. Techniquement, ce pourra être un logiciel de contrôle installé sur un routeur.



Nous aurons, par exemple :

- Le réseau privé dans une classe d'adresses IP privées, comme 192.168.0.0,
- la DMZ dans une autre classe privée comme 192.168.1.0 ou, si l'on a les moyens, dans un morceau de classe publique, de manière à pouvoir accéder à ces machines depuis le Net sans translation d'adresse,
- une adresse publique attribuée au routeur sur le Net par le FAI (Fixe ou dynamique. Dans ce type d'installation, il vaut mieux qu'elle soit fixe).

Le routeur cumule ici deux fonctions :

- Le routage proprement dit, pour permettre aux paquets de passer d'une zone à l'autre alors que ces zones ne sont pas situées dans le même réseau IP,
- Le filtrage du trafic entre les diverses zones, c'est la fonction de FireWall.

Bien entendu, qui peut le plus peut le moins, il peut ne pas y avoir de DMZ, si l'on n'a pas besoin d'exposer de serveurs sur le Net.

Sur des petites structures, la DMZ peut être implantée sur le routeur lui-même. En effet, un routeur peut très bien n'être rien d'autre qu'un serveur avec plusieurs interfaces réseau et un logiciel de routage. Ce n'est bien entendu pas une solution très "propre" au sens de la sécurité.

Les trois passages

Revenons à notre schéma initial.

1- Entre le réseau privé et le Net

Toujours typiquement, ce sont les clients du réseau (les utilisateurs) à qui l'on va donner des possibilités d'accéder au Net comme par exemple le surf ou la messagerie. Toutes les requêtes partent du réseau privé vers le Net. Seules les réponses à ces requêtes doivent entrer dans cette zone. Les accès peuvent être complètement bridés (les clients du réseau privé n'ont aucun droit d'accès vers le Net, ça nuit à leur productivité. Seul le patron y a droit). Ou alors, les utilisateurs ne pourront consulter qu'un nombre de sites limités, dans le cadre de leurs activités professionnelles exclusivement. Très généralement, cette zone est construite sur une classe d'adresses privées et

nécessite donc une translation d'adresse pour accéder au Net. C'est le routeur qui se chargera de cette translation.

2- Entre la DMZ et le Net

Ici, nous avons des serveurs qui doivent être accessibles depuis le Net. Un serveur Web, un serveur de messagerie, un FTP... Il faudra donc permettre de laisser passer des connexions initiées depuis l'extérieur. Bien entendu, ça présente des dangers, il faudra surveiller étroitement et ne laisser passer que le strict nécessaire.

Si l'on dispose d'adresses IP publiques, le routeur fera un simple routage. Si l'on n'en dispose pas, il devra faire du "port forwarding" pour permettre, avec la seule IP publique dont on dispose, d'accéder aux autres serveurs de la DMZ. Cette technique fonctionne bien sur un petit nombre de serveurs, mais devient très vite un casse-tête si, par exemple, plusieurs serveurs HTTP sont présents dans la DMZ.

3- Entre le réseau privé et la DMZ

Les accès devraient être à peu près du même type qu'entre la zone privée et le Net, avec un peu plus de souplesse. En effet, il faudra :

- Mettre à jour les serveurs web,
- Envoyer et recevoir les messages, puisque le SMTP est dedans,
- Mettre à jour le contenu du FTP (droits en écriture).

En revanche, depuis la DMZ, il ne devrait y avoir aucune raison pour qu'une connexion soit initiée vers la zone privée.

Les divers types de FireWall

Bien. Maintenant que nous savons à peu près ce que l'on peut et ce que l'on ne peut pas faire entre les diverses zones, voyons comment construire ce firewall.

Il y a déjà deux moyens différents de s'y prendre. Soit l'on travaille au niveau TCP et UDP en s'intéressant aux adresses IP des sources et des cibles, ainsi qu'aux ports employés, nous ferons du filtrage de paquets, soit l'on travaille au niveau de l'application (HTTP, SMTP, FTP). Nous ferons alors du "proxying".

Rien d'ailleurs n'interdit de faire les deux.

Si l'on travaille au niveau des paquets, il y a encore deux méthodes, l'une triviale et l'autre plus fine. Pour comprendre la différence entre les deux, ce n'est pas facile. Disons qu'une connexion entre un client et un serveur peut engendrer plusieurs connexions sur des ports différents.

Sans aller très loin, une simple consultation de page web suffit à expliquer ce qu'il se passe. Si le client envoie bien la requête toujours sur le port 80 du serveur, il attend en revanche la réponse sur un port qu'il va choisir aléatoirement, généralement en dessus de 1024. Comme le port de réponse est aléatoire, ça va être difficile de laisser passer les réponses sans ouvrir tous les ports au dessus de 1024 en entrée vers la zone privée.

Pour être efficace, il faut être capable d'assurer un suivi de la connexion, en analysant la requête

initiale pour découvrir le port sur lequel le client recevra la réponse et agir dynamiquement en fonction. C'est ce que l'on appelle le suivi de connexion ("conntrack" en jargon informatico-anglo-saxon).

Le Firewall par filtrage simple de paquets ("stateless")

C'est donc le plus trivial. Il ne sait que vérifier si les "sockets" (Couple adresse IP : port) source et destination sont conformes ou non à des autorisations de passage. Dans la pratique, si la machine cliente d'IP 217.146.35.25 doit pouvoir accéder au serveur web 124.8.3.251 et, bien entendu, recevoir ses réponses, nous dirons en français :

- Les paquets venant de 217.146.35.25 vers 124.8.3.251, port 80 passent.
- Les paquets venant de 124.8.3.251, port 80 vers 217.146.35.25, ports >=1024 passent.

Si nous voulons étendre à tous les serveurs web possibles, nous ne pourrons plus tenir compte de l'IP des serveurs, ça aura pour conséquence que tout le Net pourra accéder à 217.146.35.25 sur tous les ports au dessus de 1024.

Si nous voulons que toute la zone privée puisse accéder à tous les serveurs web du Net, alors, tout le réseau privé sera accessible sur les ports au dessus de 1024.

L'exemple est minimaliste, les IP des clients sont des IP publiques, il n'y a pas de translation d'adresse mise en oeuvre, ce qui rend les risques maximums. Dans la pratique, le cas est rarissime. Il ne sert ici qu'à montrer les limites de ce filtrage simple. Notez qu'avec de tels systèmes, la translation d'adresse est de toutes manières rendue impossible. Pour faire de la translation d'adresse, il faut être capable de suivre les connexions.

Le Firewall par suivi de connexion ("statefull")

Ici, nous mettons en oeuvre le suivi des connexions. Par un procédé que nous conserverons obscur pour le moment, le firewall va savoir si une connexion est :

- NEW.
  C'est à dire qu'elle est créée. Par exemple, un client envoie sa première requête vers un serveur web.
- ESTABLISHED.
  Cette connexion a déjà été initiée. Elle suit dans les faits une connexion NEW que l'on a déjà vu passer.
- RELATED.
  Ce peut être une nouvelle connexion, mais elle présente un rapport direct avec une connexion déjà connue.
- INVALID.
  Un paquet qui a une sale tête, un paquet qui n'a rien à faire là dedans, un paquet qui sent l'ail.

Là, ça va devenir plus facile. Nous dirons en français :

- Toutes les connexions NEW qui viennent du réseau privé et qui vont sur le NET port 80 sont acceptées. Tous les clients du réseau privé peuvent interroger tous les serveurs web du Net.
- Toutes les connexions RELATED et ESTABLISHED qui viennent du Net port 80 sont

autorisées à rentrer. Les serveurs peuvent répondre.

- Toutes les connexions RELATED et ESTABLISHED qui sortent du réseau privé vers le Net sont autorisées à sortir. Les connexions peuvent se poursuivre, même si elles ouvrent de nouvelles connexions.
- Toutes les connexions INVALID, d'où qu'elles viennent sont jetées à la poubelle, on n'aime pas l'odeur de l'ail.

Certains suivis de connexions ne sont pas simples à faire et nécessitent des algorithmes spécifiques, comme principalement le FTP.

Netfilter, avec son interface IPTables, dans les noyaux 2.4 de Linux savent très bien travailler de cette manière.

Les FireWalls applicatifs

Ici, on ne travaille plus au niveau du transport, mais au niveau de l'application, c'est à dire au niveau du protocole applicatif, comme HTTP, FTP ou autre. Un tel "FireWall" est en réalité un serveur "Proxy", c'est à dire que le client s'adressera **toujours** à lui, quelle que soit la cible finale et n'acceptera de réponse **que de sa part**. Le Proxy reformule la requête pour son propre compte vers le Net et, lorsqu'il reçoit la réponse, la transmet au client comme si c'était lui qui répondait directement.

En gros, ça veut dire que le proxy est le seul à accéder au Net et qu'il est donc le seul à risquer de subir des attaques.

Cette méthode, si elle est simple à mettre en oeuvre avec des clients HTTP ou FTP, qui sont pratiquement tous prévus pour fonctionner dans ce cadre, peut poser des problèmes avec des applications qui ne le prévoient pas, comme c'est souvent le cas pour les clients de messagerie.

Avantages et inconvénients

Reprenons à l'envers.

Le Proxy

Ca pourrait apparaître comme la solution ultime, puisqu'il y a effectivement une barrière entre le Net et la zone protégée. Sauf que ces logiciels sont complexes et qu'ils peuvent contenir des bugs que les pirates, tôt ou tard, découvriront et exploiteront pour passer quand même. De plus, ils consomment pas mal de ressources et nécessitent des machines relativement musclées.

Le FireWall "StateFull"

C'est pas mal, c'est souple à paramétrer, ça consomme peu de ressources, mais le suivi de connexion est délicat et, s'il y a un bug dans le système, ça ouvre des portes aux pirates. Les premières versions de Netfilter en ont présenté beaucoup, surtout sur les protocoles délicats comme FTP.

Le Firewall "StateLess"

C'est très simple, ça consomme très peu de ressources, il n'y a quasiment aucun risque de bug, mais

ça ouvre trop de portes pour qu'un protocole applicatif ne risque pas de rester planté à un moment ou à un autre.

Vous le voyez, il n'y a pas de solution miracle.

Et avec, ça va mieux ?

Oui, bien sûr, un firewall est indispensable. Pour autant, se croire à l'abri parce que son firewall a passé les tests de sécurité classiques relève de la plus pure utopie. Pourquoi ?

Il n'existe pas de firewall inviolable, tout simplement. Pour plusieurs raisons :

- Nous l'avons vu, les systèmes puissants comme un firewall statefull ou un proxy renferment des logiciels complexes qui ne peuvent pas être totalement exempts de failles. Il suffit au pirate d'en trouver une.
- Les règles de filtrage placées sur ces outils sont obligatoirement un compromis entre la sécurité maximale et une certaine liberté d'usage de l'Internet. Là où il y a compromission, il y a aussi faiblesse.
- Les proxys travaillent sur les couches hautes, on peut les percer en passant sur les couches basses.
- Les filtres de paquets travaillent sur les couches basses, on peut les percer en exploitant les failles des couches hautes.

Nous pouvons multiplier les barrages en les diversifiant, mais ça ne procure pas une protection absolument certaine. Pour obtenir une sécurité maximale, (mais jamais absolue), il faut travailler à tous les niveaux :

- Configurer ses serveurs avec le plus grand soin, en éliminant tous les risques connus à leur niveau,
- s'assurer que l'on n'a pas dans un coin un serveur qui tourne sans qu'on le sache. Ne riez pas, le cas est assez fréquent,
- protéger le tout avec un système de filtrage efficace, bien adapté à ses besoins,
- surveiller avec le plus grand soin tout se qu'il se passe, aussi bien sur les serveurs que sur les filtres, pour détecter le plus rapidement possible toute activité anormale,
- avoir perpétuellement à l'esprit qu'il y a forcément quelque part une faille que l'on ne connaît pas, et qu'un pirate peut trouver.

Vous le voyez, la sécurité informatique oblige à devenir complètement paranoïaque. Mais réfléchissez bien... Dans la vie, c'est pareil. Si vous voulez vivre avec un risque zéro, vous aurez les mêmes problèmes :)

Sueurs froides

- Votre réseau est bien à l'abri, derrière tous les murs pare-feu de la création, avec une surveillance de tous les instants sur les points d'accès au Net. Mais dans votre réseau, il y a des utilisateurs qui se servent de portables. Lorsqu'ils rentrent chez eux, ils connectent leur portable au Net sans aucune sécurité et se gavent de virus, de backdoors et autres spywares.

Le lendemain, ils reviennent se connecter sur votre beau réseau tout propre.

- Un proxy travaille dans les couches hautes. On peut le casser en passant par les couches basses ou en exploitant des failles logicielles. Donc, on ajoute aussi un filtre de paquets. Mais le filtrage de paquets peut plus ou moins être contourné, si l'on arrive à installer dessus une autre pile IP, parallèle à celle qui est utilisée par le filtre. Une sorte de super backdoor qui permettra de construire un routeur pirate à côté du votre, très étroitement surveillé, et qui ne s'apercevra de rien.

Et les bons vieux Windows...

Revenons à une situation plus commune, celle de l'internaute privé, qui ne dispose pas d'un réseau de type entreprise chez lui. Tout au plus.

Il faut distinguer deux cas qui n'ont pas beaucoup de rapports :

- Ceux qui utilisent les services d'un petit routeur NAT du commerce, on en trouve maintenant à moins de 150 €. En général, ce type d'équipement propose déjà pas mal de fonctions de filtrage de paquets,
- ceux qui utilisent une machine connectée directement au Net et qui peut éventuellement partager la connexion avec un ou deux autres postes. Les dernières versions de Windows (Me, 2000 et XP) proposent un moyen simple de le faire, Mac OS X également. Dans ces cas, il est clair que la machine exposée devra disposer d'un minimum de protections. Si Mac OS X propose un filtre de paquets performant, hérité de ses origines BSD, les diverses version Windows ne sont pas très performantes dans ce domaine, bien qu'au moins 2000 et XP proposent un filtrage de paquets.

Windows NT, 2000, XP

Les systèmes "professionnels" de Microsoft. N'en déplaise aux détracteurs de ces produits, NT n'est pas une passoire, il est même (je vais me faire des ennemis) beaucoup plus sûr qu'une version Linux installée sans précautions. Pourquoi? Pour au moins la raison suivante: Windows NT, sauf dans sa version "terminal server", ne sait pas exécuter des tâches à distance, sauf pour le groupe des administrateurs, qui peuvent en lancer quelques unes (mais pas des moindres, il est vrai). Un minimum de précautions reste toutefois incontournable :

- Ne laissez pas le compte d'administrateur par défaut avec un mot de passe vide :-) (C'est évident, mais combien l'ont fait?)
- Au moins Windows 2000 et Windows XP permettent de changer le nom de l'administrateur. Dans leur version Française, l'administrateur s'appelle "Administrateur" (comme root sur Linux). Changez ce nom.
- Adoptez pour ce compte (et ceux d'éventuels autres administrateurs) un mot de passe "fort" :
    - 7 caractères minimum; 14, c'est mieux
    - Utilisez une combinaison de caractères sans signification (recherches par dictionnaire), avec des majuscules, des minuscules, des caractères de ponctuation et même éventuellement des caractères non imprimables
- Désactivez NetBIOS sur TCP/IP sur l'interface connectée au Net.
- N'installez pas de serveur inutile. IIS est dangereux, l'agent SMTP de l'option pack aussi, le DNS également.
- Proscrivez dans la mesure du possible tout logiciel de prise de contrôle à distance.
- bloquez les ports 135 à 139 aussi bien en UDP qu'en TCP, ça peut se faire sur NT, 2000 et XP sans logiciel supplémentaire, dans la configuration de la pile IP.

- installez les derniers Service Packs.
- vérifiez chez Microsoft l'apparition de nouveaux patchs de sécurité.
- vérifiez fréquemment que vous n'avez pas installé un "backorifice" ou un "netbus" par inadvertance. (Ctrl+Alt+Suppr permet de visualiser les tâches et processus en cours. C'est pas toujours très compréhensible, mais on y gagne beaucoup à analyser la liste des processus en cours).
- Eventuellement, installez un antivirus sérieux.

Avec ces précautions, vous devriez déjà être relativement à l'abri.

Windows 9x et Me

Là encore, je risque de me faire d'autres ennemis: Vous risquez plus du côté des plantages à répétition que du côté des pirates... Moyennant tout de même quelques précautions.

Ces systèmes d'exploitation n'installent par défaut aucun serveur, donc aucun port d'écoute, si ce n'est bien entendu les ports 137 à 139. Par ailleurs, les outils d'administration distante ne sont pas installés par défaut, ce qui enlève encore un point faible par rapport à leurs grands frères NT, 2000 et XP. Heureusement d'ailleurs, parce que si c'est plus difficile de rentrer dans une machine de ce type, une fois que l'on est dedans, il n'y a plus aucune barrière, puisqu'il n'y a jamais d'authentification de l'utilisateur.

- N'installez pas de logiciels de contrôle à distance (netmeeting par exemple).
- N'installez pas PWS (Personal Web Server), sauf si vous en avez absolument besoin. Dans ce cas, ne le laissez actif que le temps où vous en avez besoin et déconnectez-vous si possible de l'Internet. C'est contraignant, mais PWS est trop dangereux sur ces OS.
- Ne partagez pas de ressources. Si c'est absolument nécessaire, faites-le sur un répertoire et certainement pas sur la racine de votre disque système. De plus, mettez-y un mot de passe "fort" aussi bien pour l'écriture que pour la lecture.
- Là aussi, installez les derniers service packs et patchs de sécurité ***pris directement chez Microsoft et absolument pas ailleurs.***
- Installez ***absolument*** un antivirus ***du commerce, pas n'importe quoi que vous avez téléchargé n'importe où*** sous prétexte qu'il est gratuit. Au mieux, vous installez un guignol, au pire, c'est un cheval de Troie et maintenez régulièrement sa base de donnée à jour ***à partir du site officiel du fournisseur***. (Ces directives sont d'ailleurs également vraies pour tous les autres systèmes).

En revanche, vous ne pourrez pas grand chose contre les attaques destinées à bloquer votre machine. Dans ce cas, seul un logiciel de surveillance pourra limiter les risques.

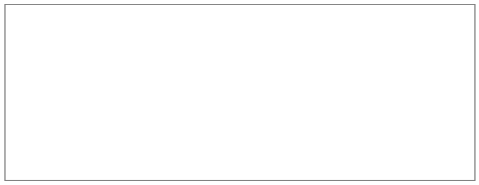
Les logiciels de surveillance

Il en existe un grand nombre, plus ou moins efficaces, plus ou moins simples à configurer, plus ou moins gratuits.

Nous n'allons pas parler de tous, mais de deux, gratuits ou pas chers. Je n'entrerai pas dans les

détails de ces deux logiciels, ma configuration ne nécessitant pas l'usage de ces outils. Ceux qui voudront préparer à leur sujet des pages d'explications plus détaillées seront les bien venus. Ces deux logiciels sont réputés fonctionner sous WIndows 95, Windows 98, Windows NT et Windows 2000.

Zone Alarm.



Zone Alarm[6] est gratuit dans sa version personnelle. Bien qu'en anglais, son principal avantage est qu'il se configure assez simplement. Son inconvénient est également qu'il se configure assez simplement, c'est à dire qu'il n'offre pas de possibilités très fines.

Il n'en demeure pas moins un outil sérieux et efficace dans la plupart des cas.

Look 'n Stop

Look 'n Stop est écrit par un français, que vous rencontrerez peut-être sur news://nzn.fr.cable.wanadoo sous le pseudonyme de XunK. C'est également l'auteur du fameux Stat 'n Perf[7], le compteur de trafic IP indispensable pour les câblés sous Windows.

La version Beta est, à l'heure où j'écris ces lignes, gratuite pour un usage personnel.

Vous trouverez toutes les informations nécessaires sur son site officiel[8].

A mon sens, cet outil parait plus intéressant dans la mesure où il est finement configurable par l'utilisateur. Le seul inconvénient, mais en est-ce vraiment un en matière de sécurité, est qu'il nécessite de comprendre un peu ce que l'on fait. Il dispose tout de même de règles par défaut déjà établies pour les usages courants.

Conclusions

Je tiens à m'excuser auprès des possesseurs de Mac. (Apple), je n'ai aucune compétence dans ce domaine. Mais des outils analogues doivent exister.

La protection d'une machine connectée à l'Internet sous Windows (comme sous n'importe quel OS d'ailleurs) passe d'abord par des principes de prudence et de sécurité de bon sens, encore faut-il être suffisamment averti des problèmes potentiels et des règles élémentaires d'hygiène en matière de réseaux et j'espère que cet exposé vous aura aidé à y voir un peu plus clair dans ce domaine.

6 Zone Alarm : http://www.zonelabs.com/
7 Stat 'n' Perf : http://www.soft4ever.com/StatnPerf/Fr/index.html
8 Look 'n' Stop : http://www.soft4ever.com/LooknStop/Fr/index2.htm

En plus de cela, l'usage d'un outil de surveillance reste tout de même conseillé, mais afin de ne pas sombrer dans la paranoïa, il convient de se renseigner sur les risques, les mécanismes des réseaux pour pouvoir faire le tri dans les alertes, entre celles qui sont réellement dangereuses et celles qui ne le sont pas.