CENTRE NATIONAL
DE LA RECHERCHE
SCIENTIFIQUE

Filtrage IP MacOS X, Windows NT/2000/XP et Unix

Modifications de ce document

- v 1.0 : 06/2002 : Denis Pugnère (Denis.Pugnere@igh.cnrs.fr)
- v 2.0 : 02/2005 : Denis Pugnère (d.pugnere@ipnl.in2p3.fr) :
  - niveaux de filtrages,
  - Win XP SP2

Plusieurs Niveaux de filtrage

- Niveau 2 : adresses adresse M.A.C, par vlan
- Niveau 3 : IP, tunnels IP dans IP
- Niveau 4 : TCP, UDP, ICMP...
- Niveaux applicatifs : en fonction des protocoles (cf RFC), utilisation de proxys

Filtrage niveau 2

- Sont généralement réalisés par éléments actifs du réseau (Commutateurs, Commutateurs routeurs)
  - VLAN (IEEE 802.1Q) :
    - Par port physique
    - Par adresse MAC
    - Par adresse IP
  - Par utilisateur (802.1x)
  - Trames ethernet broadcast
  - ATM
  - PPP

Filtrage niveau 3

- Réalisé sur les routeurs
- Sur les adresses IPv4, IPv6, multicast
  - Anti-spoofing en entrée et en sortie
  - Adresses broadcast .255 .0 (fonction du subnet-mask)
  - RFC 1918 (adresses martiennes)
  - Adresses réservées
- Tunnels IP (GRE, protocole 47)
- VPN, IPsec
- Référence sur les réseaux réservés :
  - http://www.cymru.com/Bogons/index.html

Filtrage niveau 4

- Réalisé sur les routeurs, firewall et serveurs
- Protocoles
  - ICMP (type et code)
  - UDP (port source, destination)
  - TCP
    - Ports (source et destination)
    - Flags (SYN, ASK, RST, PSH, URG, FIN)
  - Routage BGP, RIP, OSPF...

Filtrage niveau applicatif

- En fonction des spécifications des protocoles
- Proxies (mandataires) applicatifs
- Proxies « circuits » : SOCKS
- Utilisation de firewall « statefull inspection »
  - Suivi des connexions déjà établies
  - Diminution complexité des règles de filtrage
- Exemples de proxies applicatifs :
  - ftp,
  - H323,
  - Proxy http (filtrage d'URL, javascript, cookies)

Avantages du filtrage par le système

- Interception du trafic par la pile IP du système –kernel mode- (avant l'envoi du flux vers les applications -user mode-)
- Offre un point central d'interception du trafic pour toutes les applications
- Sous Unix : Remplace avantageusement le filtrage type TCP-WRAPPER (uniquement TCP)

Filtrage sous Windows NT/2000

- Inclus dans le système
- Non activé par défaut
- Activation W NT4 :
  - Propriétés réseau
  - protocole TCP/IP, Propriétés
  - Cliquer sur « avancé… »
  - Cliquer sur "Activer la sécurité"
- Activation W2000 :
  - propriétés d'accès au réseau local
  - sélectionner : protocole internet : tcp/ip
  - Cliquer sur « avancé… »
  - onglet "options"
  - choisir : filtrage TCP/IP, propriétés
  - filtrages : port TCP, port UDP, protocole IP

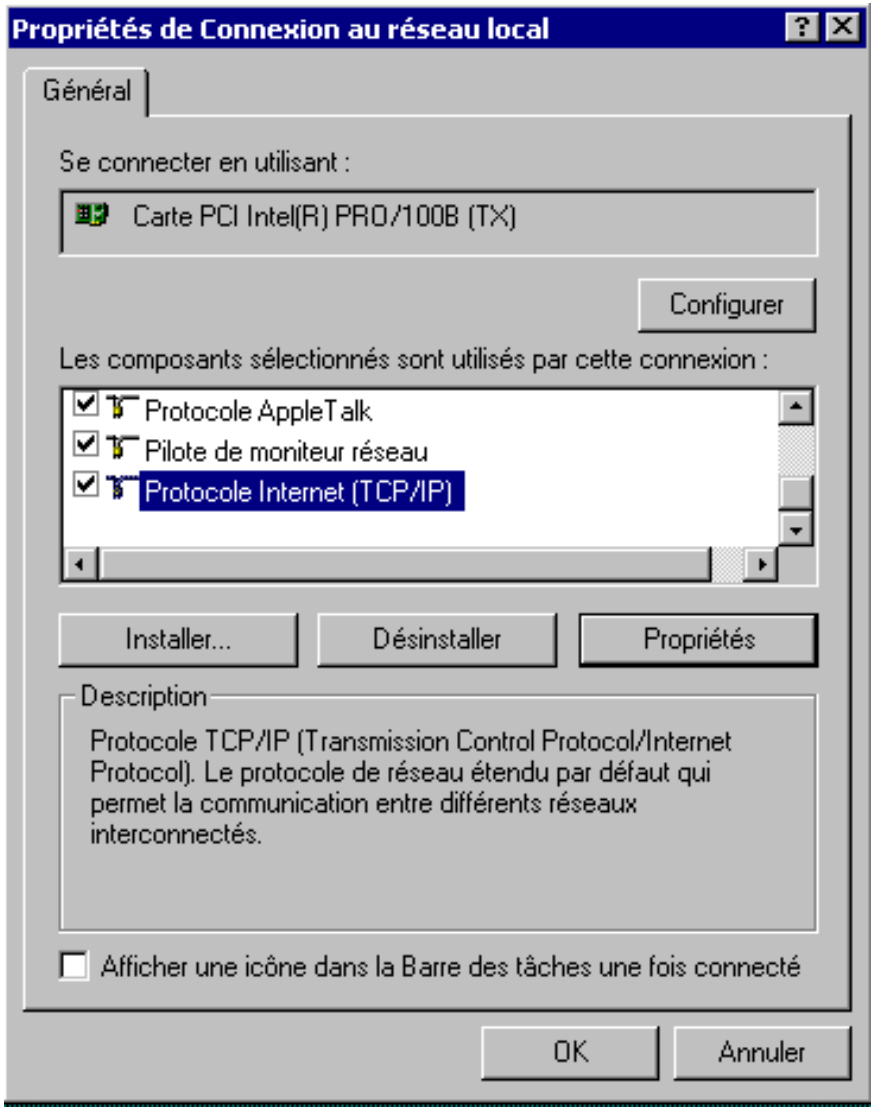
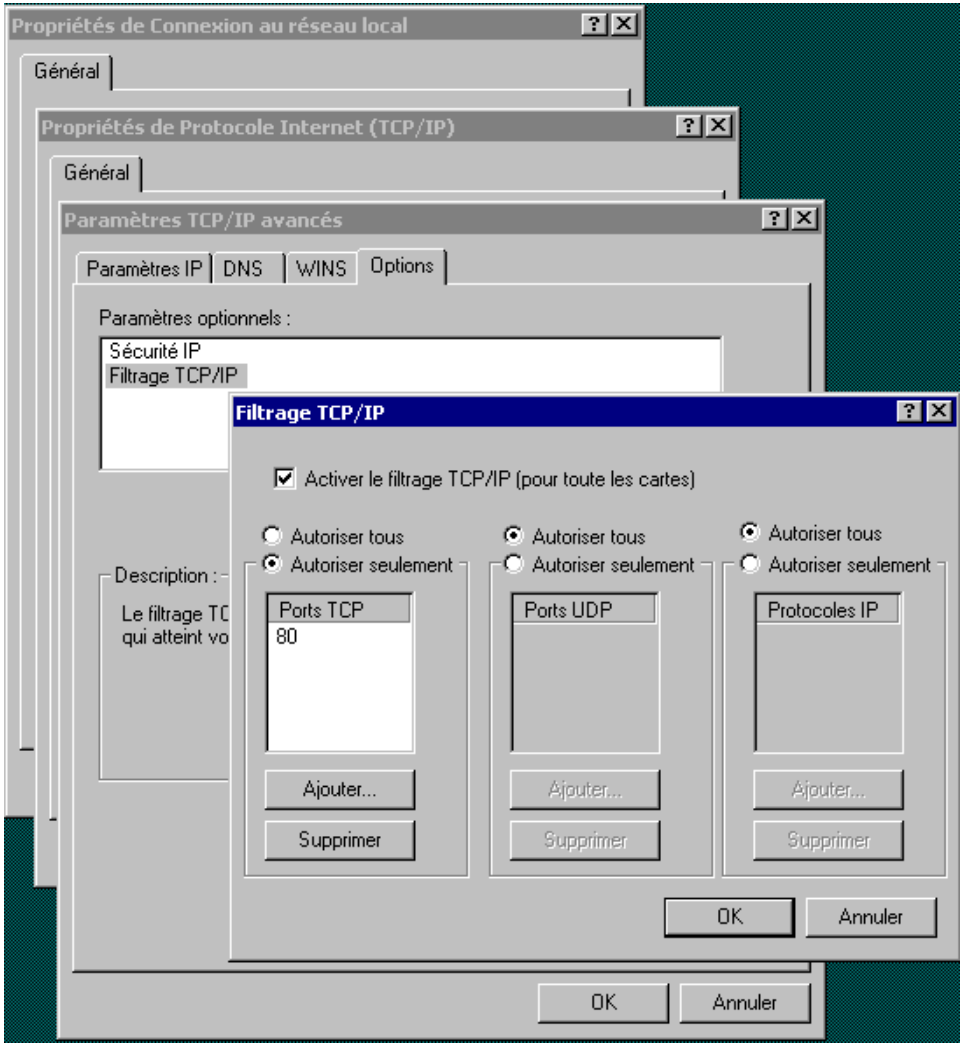


Illustration filtrage sous Windows NT/2000

- Filtrage sur :
  - n° port TCP
  - n° port UDP
  - n° protocole IP
- Sur chaque carte réseau
- pas très souple : « Autoriser tous » ou « Autoriser seulement » ; pas « interdire seulement un port donné »
- Utilisable pour machine en zone semi-ouverte : exemple : serveur http
- Voulez vous redémarrer ?

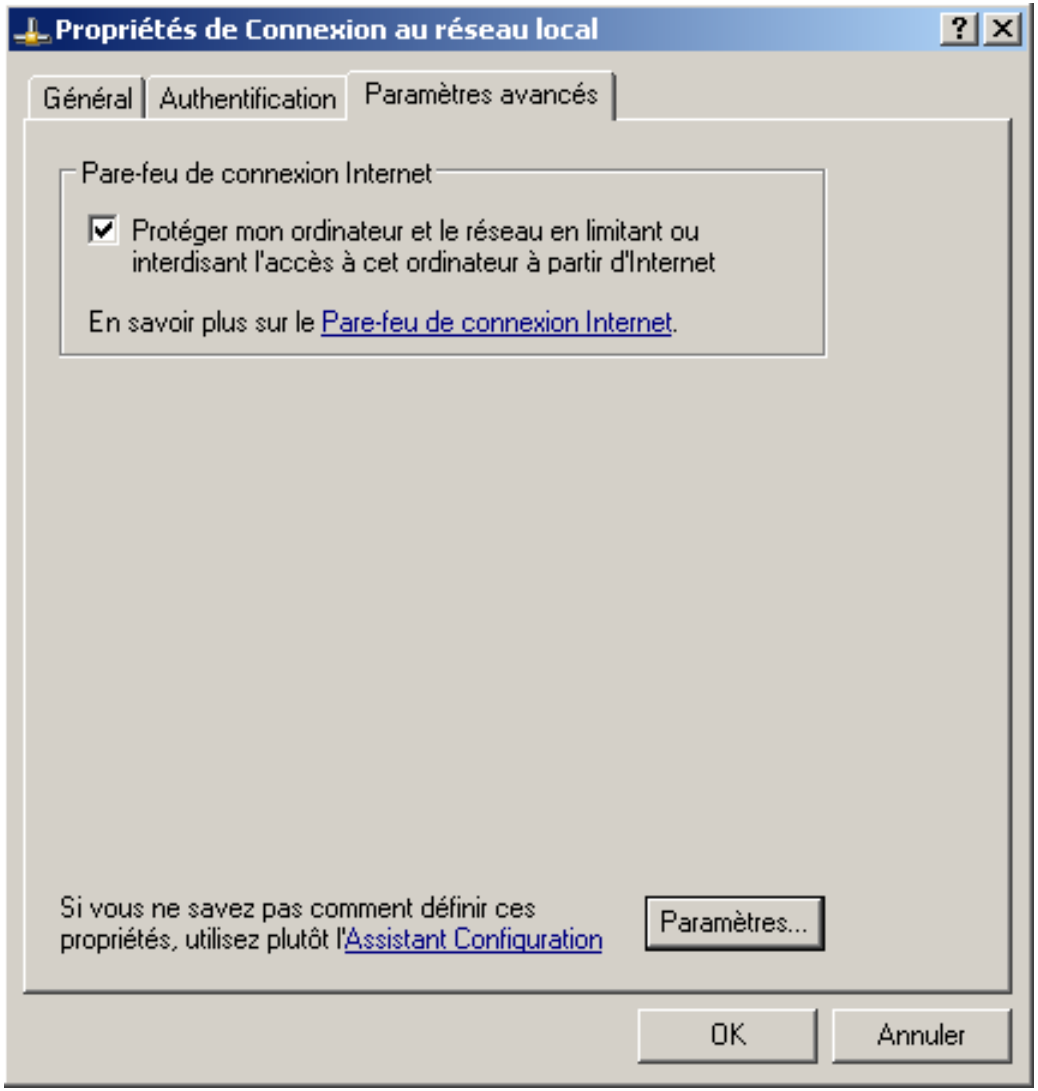


Filtrage IP sous Windows NT/2000

- Remarques et limitations
  - Pas de trace des paquets rejetés
  - Pas de filtrage ICMP (TCP et UDP seulement)
  - Pas filtrage sur les connexions TCP ouvertes (depuis le poste) ?
  - Pas de filtrage par application (ports ouverts dynamiquement)
  - Pas de filtrage par processus
  - Utiliser d'autres produits ?

Filtrage IP sous Windows XP SP1

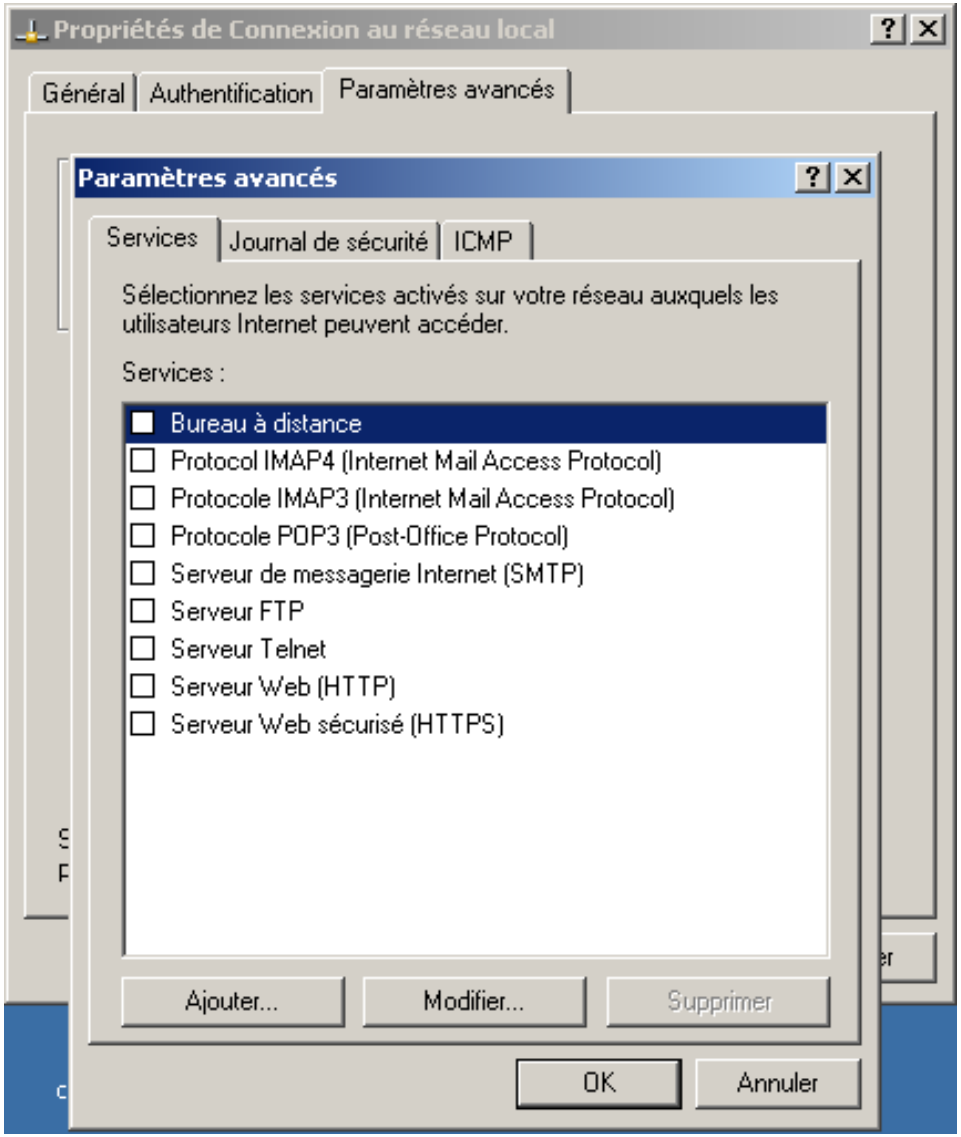
- 2 systèmes de filtrage
  - « à la W2000 »
  - « à la WXP » : Internet Connection Firewall
- ICF : Apparu avec WXP :
  - Propriété de connexion au réseau local
  - Onglet « Paramètres avancés »
  - Paramètres…



ICF de Windows XP (et XP SP1)

- Selon Microsoft : à mémoire d'état : « stateful inspection » : laisse passer le trafic entrant issu d'une connexion initiée de l'ordinateur
- Services prédéfinis
- Possibilité d'ajouter des filtres sur de nouveaux services
- Possibilité d'enregistrer dans un journal les paquets rejetés (journal de sécurité)…
- Possibilité de filtrer les paquets TCP, UDP et ICMP

- Pas de filtrage applicatif (ports ouverts dynamiquement)
- Pas de filtrage par processus



ICF suite…

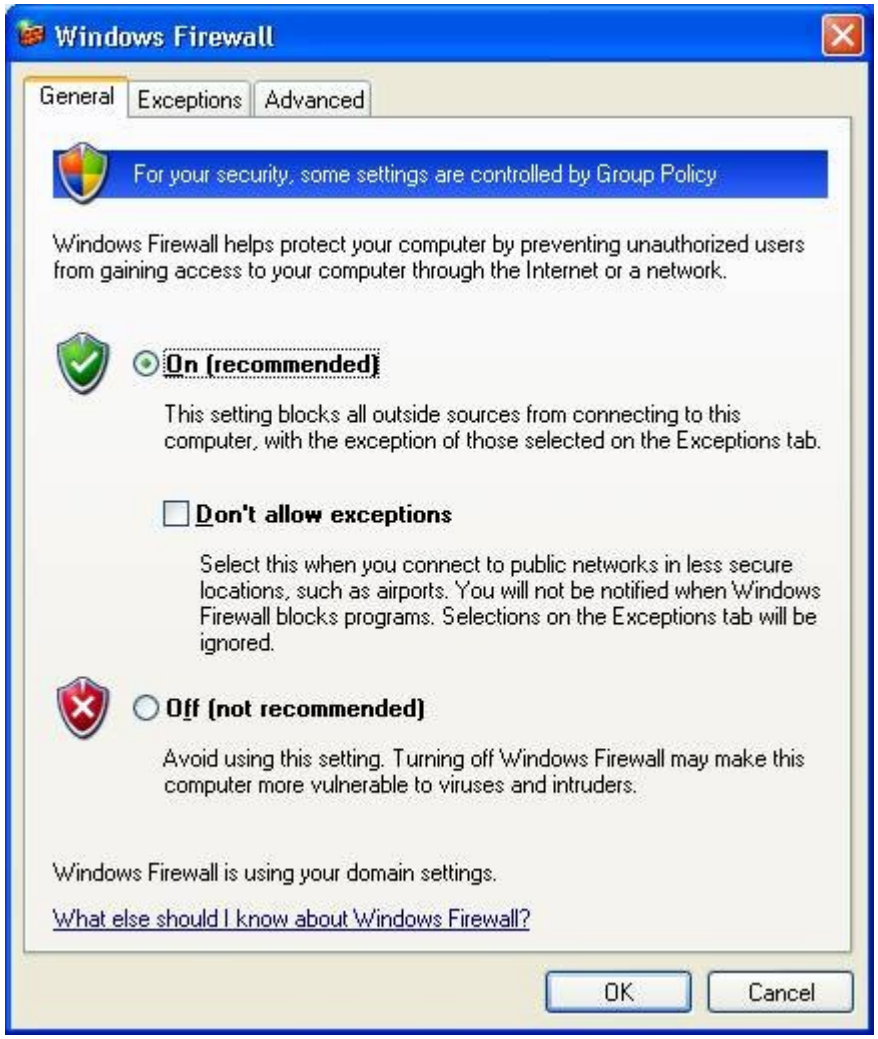
- ICF et partage de connexion Internet (ICS)
  - ICF sert de firewall pour tous les ordinateurs du réseau local
  - MS Exchange et Office2000/Outlook : problème car utilise les RPC (c'est le serveur qui avertit les clients d'un nouveau message)
- ICF et journalisation
  - Peut enregistrer traces sur trafic autorisé et trafic stoppé
  - Taille maximale du journal paramétrable
  - Format (Extended Log File Format) défini par le W3C

Windows XP SP2

- Remplacement d'ICF par « Windows Firewall »
- Rappel : Sous XP SP1 : ICF désactivé par défaut
- Sous XP SP2 :
  - Windows Firewall activé par défaut sur toutes les interfaces
  - Possibilité d'appliquer des filtres globaux sur toutes les connexions
  - Exeptions :
    - basées sur les applications (nom de l'exécutable)
    - basées sur les adresses réseau (IPv4, IPv6, réseau IP local)
    - Possibilité de bloquer temporairement toutes les exeptions
  - Intégration au « group policy »
- Ajout d'une option (-b) à netstat : proccessus écoutant sur les ports
  `netstat -b` <=> `netstat -o` + `tasklist /svc`
- Commande netsh : affichage de l'état du firewall (règles...)

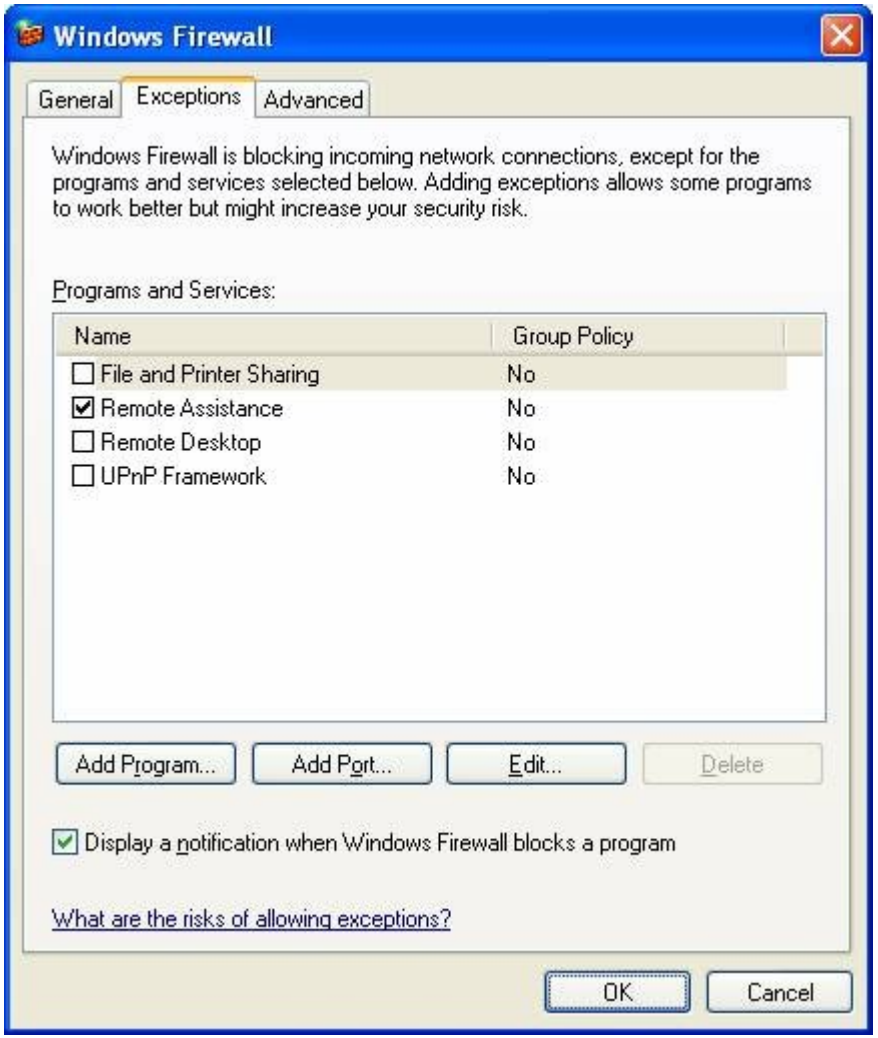
  netsh firewall show state enable

Windows XP SP2 : onglet général



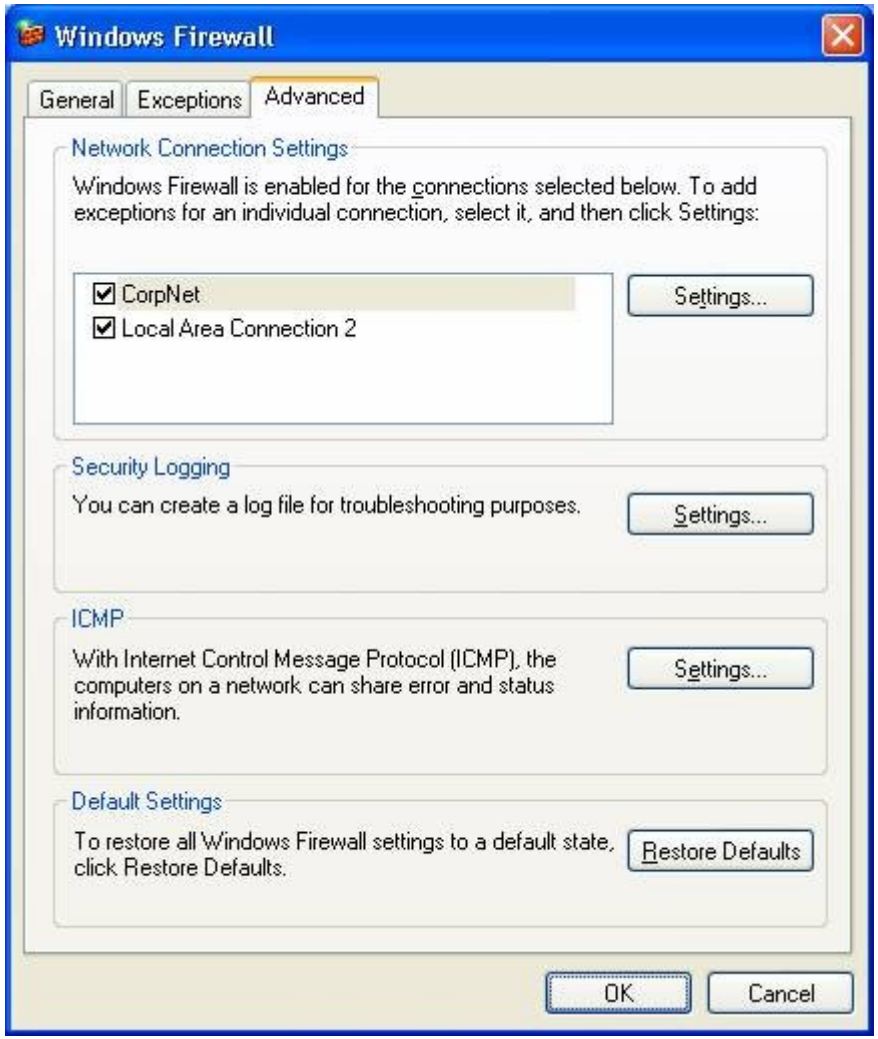
- "On" : tout le trafic entrant est bloqué, sauf pour les exeptions
- "Don't allow exceptions" : tout le trafic entrant est bloqué
- "Off" : « Windows firewall » est désactivé

Windows XP SP2 : onglet exeptions



- On peut activer/désactiver
  - Un programme
  - Un port
  - Une combinaison des 2

Windows XP SP2 : onglet Advanced



- « Network connection settings » :
  - On peut activer/désactiver la protection sur une interface particulière
  - Dans settings : filtrage ICMP
- « Security logging » : traçage des connexions : refusées et/ou acceptées

- Pas de filtrage des connexions sortantes !

Filtrage Unix

- Linux : plusieurs implémentations
- *BSD via ipfilter
- Solaris , AIX , HP-UX, IRIX ... via ipfilter ou bien netfilter
- IRIX, fourni avec le système : ipfilterd

Filtrage Linux

- Linux : plusieurs implémentations en fonction du noyau
  - Ipfwadm : noyau 2.0
    - 3 destinations pour un paquet : accept, deny, reject
    - Directions : in, out, both
    - Règles : input, output, forwarding, masquerading
    - Filtrage sur @IP/port/interface source ou destination
    - Masquage (masquerading) des connexions sur @IP/port/interface source ou destination
  - Ipchains : noyau 2.2
    - 6 destinations par défaut (accept, deny, reject, masq, redirect, return), création d'autres destinations
    - Support port forwarding
    - Création de chaînes (maintenance plus facile)
    - Support QOS, support négation (!)

Filtrage Linux (suite)

- Iptables/netfilter : noyau 2.4
  - destinations : ACCEPT, DENY, REJECT, TOS, MIRROR, MASQ, MARK, DNAT, SNAT, REDIRECT),
  - Support filtrage paquets mal formés ou non standards
  - Support des 6 drapeaux TCP : SYN, ACK, FIN, URG, RST, PSH
  - Support filtrage par adresse MAC
  - Support de plusieurs ports (source ou destination) dans une seule règle
  - Stateful inspection (suivi de connexion)
  - Les règles peuvent préciser le type de connexion : NEW, RELATED, INVALID, ESTABLISHED, RELATED+REPLY
  - Support étendu des traces (logs) : message personnalisé,
  - limitation du nombre (rate limiting) contre les deni-de-service

- Interfaces graphiques ou générateurs de règles :

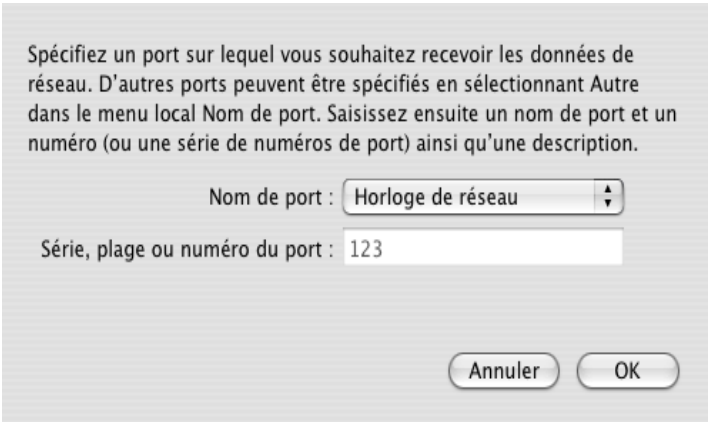
  firestarter, shorewall, fwbuilder, guarddog, narc, smoothwall, gShield...

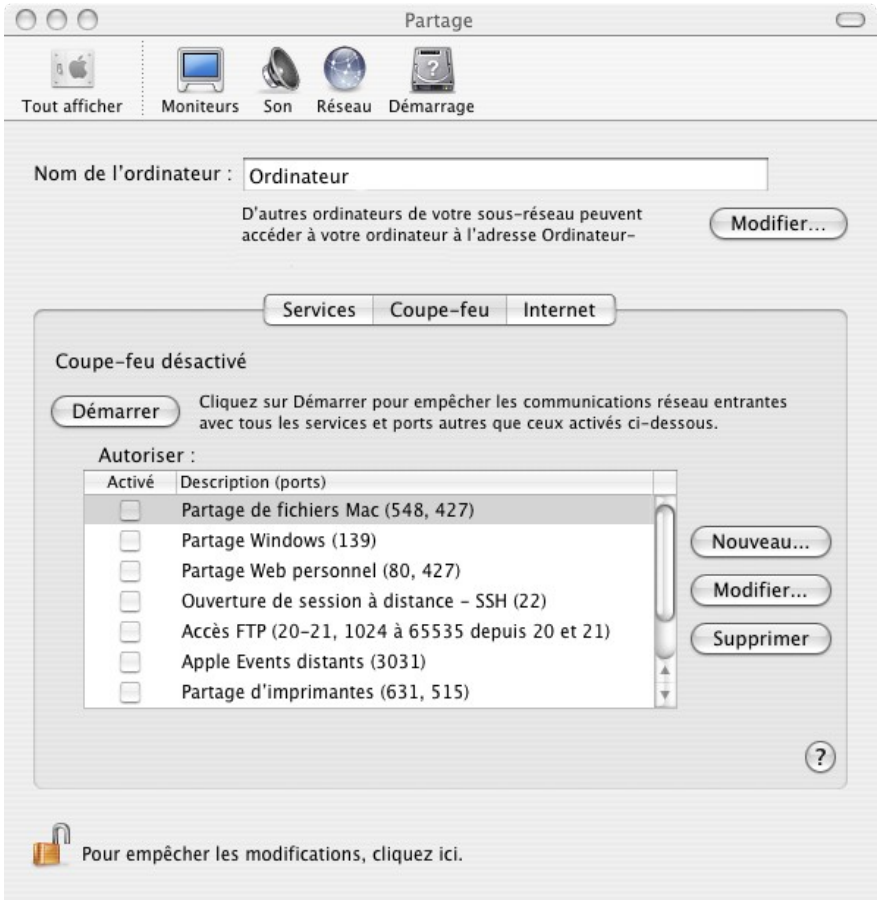
Filtrage IP sous MacOS

- MacOS …/7/8/9 : pas de filtrage IP au niveau système

- MacOS X : firewall intégré au système, application graphique :
- Services pré-déterminés

- Possibilité d'ajouter des autorisations :





Filtrage IP sous MacOS X

- Commentaires sur le « coupe-feu » intégré au système et à l'nterface graphique :
  - Pas de choix sur les protocoles : TCP, UDP, ICMP
  - Pas de traces dans un fichier (connexions refusées, connexions autorisées)
  - Pas de filtrage des connexions sortantes

Ipf (MacOS X): fonctionnement

- MacOS X : ipf (ipfilter) issu des *BSD
  - Ouvrir un shell
  - Nécessité d'être root (commande su ou sudo) ou administrateur
  - Commandes Unix : exemple : bloquer « Appleshare IP » (référence article Apple TIL 'Ports Numbers used by AppleShare IP')

```
# ipfw -f flush
# ipfw add 200 reject tcp from any to any 548
# ipfw add 201 reject tcp from any to any 427
```

- Liste des règles :

```
# ipfw  list
```

- Effacer les règles :

```
# ipfw -f flush
```

- Ajouter les règles :

```
# ipfw add ...
```

Syntaxe : <rule-number> <allow or deny> <protocol> from <source> to <destination> <options>

- Effacer les règles :

```
# ipfw delete ...
```

- Exemple de règle par défaut :

```
# ipfw add 65535 deny ip from any to any
```

- Règles prises en compte de 0 à 65535 (par ordre croissant).
- Le traitement de la chaîne des règles s'arrête quand la règle courante corresponds au paquet courant

Ipf (MacOS X) : exemples

- Accès au serveur web local depuis localhost, tout autre trafic vers le serveur web est stoppé :

```
# ipfw -f flush
# ipfw add 100 allow tcp from 127.0.0.1 to any 80
# ipfw add 101 reject tcp from any to any 80 in via en0
```

  - To any : fonctionne quelle que soit l'adresse IP (destination) de la machine (utile quand ip dynamique)
  - Autoriser accès Appleshare IP depuis un seul client (192.168.0.140) :

```
# ipfw -f flush
# ipfw add 301 allow tcp from 192.168.0.140 to any 548
# ipfw add 302 allow tcp from 192.168.0.140 to any 427
# ipfw add 303 reject tcp from any to any 548
# ipfw add 304 reject tcp from any to any 427
```

- Accès au service telnet depuis un ensemble d'adresses IP :

```
# ipfw -f flush
# ipfw add 400 allow tcp from 192.168.0.0:255.255.255.0 to any 23
# ipfw add 401 reject tcp from any to any 23
```

- NB : Trafic vers autres services autorisé

Ipf (MacOS X) : exemple de script complet restrictif

```
#!/bin/sh

IPFW=/sbin/ipfw
# interface ethernet : en0, ppp : ppp0
INTERFACE=en0

# efface toutes les regles
${IPFW} -f flush
# interface Local loopback est ouverte
${IPFW} add 1000 allow ip from any to any via lo0
${IPFW} add 1001 deny all from any to 127.0.0.0/8
# autorise le trafic sortant
${IPFW} add 2000 pass tcp from any to any out via ${INTERFACE}
# autorise connexions TCP deja ouvertes
${IPFW} add 3000 pass tcp from any to any established
# Autorise les fragments IP de passer
${IPFW} add 4000 pass all from any to any frag
# Autorise les requètes et réponses DNS
${IPFW} add 5000 allow udp from any to any 53 out via ${INTERFACE}
${IPFW} add 5001 allow udp from any 53 to any in via ${INTERFACE}
# Arrete tout trafic hors ICMP
${IPFW} add 8000 deny tcp from any to any via ${INTERFACE}
${IPFW} add 8000 deny udp from any to any via ${INTERFACE}
```

- Très restrictif
  - Gène les flux FTP en mode actif
  - Gène MS Netmeeting et Quicktime/RealPlayer (mode RTP)

Ipf (MacOS X) : exemple de script complet moins restrictif

```
#!/bin/sh

IPFW=/sbin/ipfw
INTERFACE=en0
# efface toutes les règles
${IPFW} -f flush

# Autorise le trafic sortant
${IPFW} add 2000 pass tcp from any to any out via ${INTERFACE}

# Autorise les connexions TCP deja ouvertes
${IPFW} add 3000 pass tcp from any to any established

# Autorise les fragments IP a passer
${IPFW} add 4000 pass all from any to any frag

# Stoppe le trafic vers les services sur les ports r&serves
${IPFW} add 8000 deny tcp from any to any 1-1023 in via ${INTERFACE}
${IPFW} add 8000 deny udp from any to any 1-1023 in via ${INTERFACE}
```

- Autorise trafic :
  - ftp actif
  - Udp
  - Icmp

Bibliographie

- using ipfw with Mac OS X by Stefan Arentz
  http://wopr.norad.org/articles/firewall/
- Netfilter : http://netfilter.samba.org
- Ipfilter : http://coombs.anu.edu.au/~avalon
- Iptables tutorial :
  http://iptables-tutorial.frozentux.net/iptables-tutorial.html
- Windows XP SP2 : « The cable guy 01 et 02/2004 » :
  http://www.microsoft.com/technet/community/columns/cableguy/cg0104.mspx
  http://www.microsoft.com/technet/community/columns/cableguy/cg0204.mspx