Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
PREMIER MINISTRE

S . G . D . S . N
Agence nationale de la sécurité des systèmes d'information
CERTA

Paris, le 10 janvier 2006
N° CERTA-2005-INF-006

Affaire suivie par :
CERTA

NOTE D'INFORMATION DU CERTA

Objet : Filtrage et pare-feux

Gestion du document

Référence	CERTA-2006-INF-001
Titre	Filtrage et pare-feux
Date de la première version	10 janvier 2006
Date de la dernière version	-
Source(s)	Voir Documentation
Pièce(s) jointe(s)	Aucune

TAB. 1 – *Gestion du document*

Une gestion de version détaillée se trouve à la fin de ce document.

1 Introduction

1.1 Définition

La traduction officielle du terme anglais `firewall` est barrière de sécurité ou pare-feu ([JO99]). La définition qui est associée est la suivante :

« Dispositif informatique qui filtre les flux d'informations entre un réseau interne à un organisme et un réseau externe en vue de neutraliser les tentatives de pénétration en provenance de l'extérieur et de maîtriser les accès vers l'extérieur. »

Cette description souffre de quelques erreurs parmi lesquelles le recours aux notions d'intérieur et d'extérieur qui sont des notions étrangères à l'équipement et qui relèvent en fait de choix d'architecture, et un présupposé restrictif sur la politique de sécurité applicable.

On pourra retenir de façon assez large qu'il s'agit d'un dispositif informatique de filtrage de protocoles réseaux (routables) et, par extension, d'un système ou d'un groupe de systèmes permettant d'imposer une politique de sécurité entre plusieurs périmètres réseaux.

1.2 Intérêts et limites du pare-feu

1.2.1 Avantages

– Avec une architecture réseau cohérente, on bénéficie d'une centralisation dans la gestion des flux réseaux.

– De plus, avec un plan d'adressage correct, la configuration du pare-feu est peu ou pas sensible au facteur d'échelle (règles identiques pour 10 comme 10000 équipements protégés).
– L'utilisation de la journalisation offre une capacité d'audit du trafic réseau et peut donc fournir des traces robustes en cas d'incident, si le pare-feu n'est pas lui-même une des cibles.
– Enfin le pare-feu permet de relâcher les contraintes de mise à jour rapide de l'ensemble d'un parc en cas de vulnérabilité sur un service réseau : il est possible de maintenir une certaine protection des équipements non vitaux au prix de la dégradation du service avec la mise en place d'un filtrage.

1.2.2 Inconvénients

– La capacité de filtrage d'un équipement dépend de son intégration dans le réseau mais le transforme en goulet d'étranglement (capacité réseau et ressources du pare-feu).
– De par sa fonction, le pare-feu est un point névralgique de l'architecture de sécurité avec de fortes contraintes de disponibilité. Il existe des solutions permettant la synchronisation de l'état des pare-feu, comme l'élection du routeur avec `VRRP`, ou le système de haute disponiblité `CARP/pfsync` ([MB2004]) développé pour `OpenBSD`, mais beaucoup de configurations reposent encore sur un équipement unique.
– Enfin une bonne gestion d'un pare-feu nécessite la compréhension des protocoles filtrés surtout lorsque les interactions deviennent complexes comme dans les cas FTP, H323,... avec le transport de paramètres de connexion dans le segment de données. De plus il apparaît bien souvent des effets de bord liés aux diverses fonctions (couches réseaux filtrées, traduction d'adresses) et influencées par l'ordre d'application des règles.

2 Principes du filtrage

Selon l'équipement, des informations sont extraites des flux réseaux depuis une ou plusieurs des couches 2 à 7 du modèle OSI, éventuellement corrélées entre elles, et comparées à un ensemble de règles de filtrage. Un état peut être mémorisé pour chaque flux identifié, ce qui permet en outre de gérer la dimension temporelle avec un filtrage en fonction de l'historique du flux.

Les types de filtrage les plus courants sont :
– Liaison (adresse MAC Ethernet,...),
– Réseau (entêtes IP, IPX,... et type/code ICMP),
– Transport (ports TCP/UDP),
– Filtrage adaptatif (« stateful inspection ») ou dynamique,
– Session (« circuit level gateway », « proxys » génériques),
– Application : serveur(s) mandataire(s)/relais applicatifs (« proxys »),
– Dans la pratique une combinaison des types précédents est utilisée : un pare-feu protégeant un serveur `http` fera passer les requêtes clientes à travers un relais applicatif tandis que la réponse serveur ne sera analysée qu'au niveau transport pour mettre à jour l'état des sessions dynamiques.

Par la suite, ce document se limitera au cas le plus répandu, soit IPv4 avec une liaison Ethernet.

2.1 Le filtrage de paquets IP

Il s'agit d'un filtrage réalisé au niveau des couches 2 à 4 dans un routeur – une passerelle –, un pont ou un hôte.

Les critères se basent sur les champs des entêtes des différentes couches ainsi que sur l'interface d'entrée ou de sortie du paquet (figure 1) :
– Couche 2 : adresse MAC,
– Couche 3 :
  – protocole IP (généralement limité au choix accepté/refusé à l'exception des types 1 ICMP, 6 TCP et 17 UDP qui bénéficient d'une meilleure granularité),
  – durée de vie (TTL : typiquement les paquets arrivant à expiration peuvent être éliminés),
  – adresses IPs source et destination,
  – « Flags » et options IP.
– Couche 4 (et ICMP) :
  – ports source et destination (TCP/UDP),
  – « Flags » TCP,
  – type/code (ICMP).

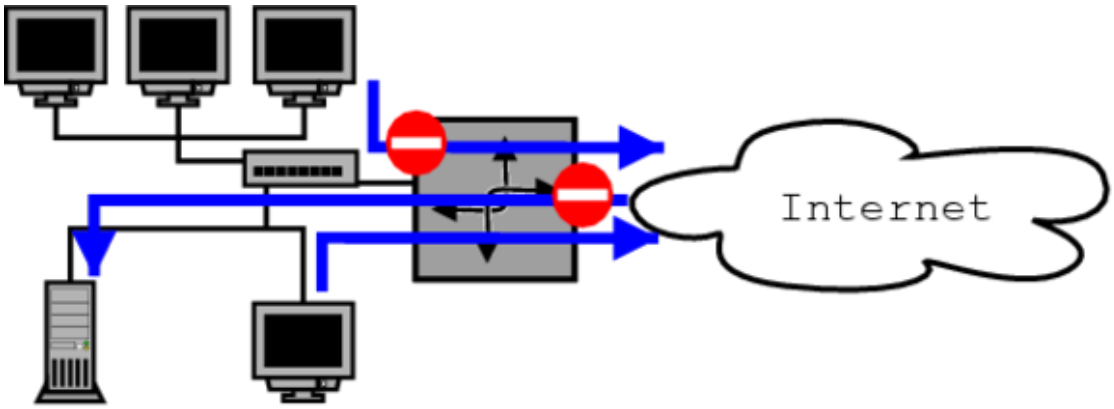

FIG. 1 – *Le filtrage de paquets IP*

Les avantages et inconvénients de ce filtrage sont :

- l'espace noyau est performant mais plus risqué (une faille induit un risque de compromission `administrateur/root`),
- il est facilement adaptable au routage (un routeur est par essence un filtre),
- il est transparent pour l'utilisateur,
- il permet une corrélation adresse source/interface en particulier :
  - « anti-spoof » (les paquets avec une adresse source correspondant à l'adressage `interne` ne peuvent venir de l'`extérieur`),
  - « egress filtering » (vérifier que les adresses sources sortant du domaine `interne` sont cohérentes avec le plan d'adressage).
- cependant il suppose que les applications respectent les ports par défauts assignés par l'IANA (si on laisse ouvert l'accès au port 25/tcp (`smtp`), un serveur `http` écoutant sur ce port au lieu de 80/tcp sera alors accessible),
- en l'absence d'historique, beaucoup de ports doivent rester ouverts pour permettre le passage des paquets en retour. Exemple de règles pour l'accès web (`http`) :

  192.168.10.0/24:1024-65535 =>*:80
  192.168.10.0/24:1024-65535 <= *:80

- il ne gère pas l'abstraction de la résolution de nom (mal adapté au filtrage des bannières insérées depuis `domaine.envahissant.tld`,...),
- la fragmentation IP pose problème (les informations de la couche transport peuvent être : absentes du 1er paquet, réparties sur plusieurs paquets, n'apparaissent pas dans les paquets suivants,...),
- l'adresse et le port source *ne sont pas* des données *fiables*,
- le filtrage des services RPC (« Remote Procedure Call ») est complexe : les ports ne sont pas standardisés et sont assignés dynamiquement (on ne peut se limiter à filtrer le service de mappage, une recherche par balayage des ports restant possible),
- il ne peut prendre en compte les services avec ports dynamiques : `FTP`,...
- il est difficile de filtrer spécifiquement un ou des hôtes si `DHCP` est employé sans précaution,
- il n'y a généralement pas de filtrage des utilisateurs.

2.2 Le filtrage en couches 5 à 7 : les serveurs mandataires

On peut distinguer deux types de serveur mandataire (« proxy », qui agit en lieu et place de son mandant, un serveur ou un client, voir figure 2) :

- les génériques, appelés « Circuit Level Gateway » qui valident la session avant d'ouvrir une connexion (vérifications adresses/ports source et destination, identifiant utilisateur, éventuelle requête `ident`,...) ; il s'agit généralement de SOCKS (la V4 supporte TCP uniquement, et la V5 rajoute l'UDP et le support de protocoles d'authentifications fortes),

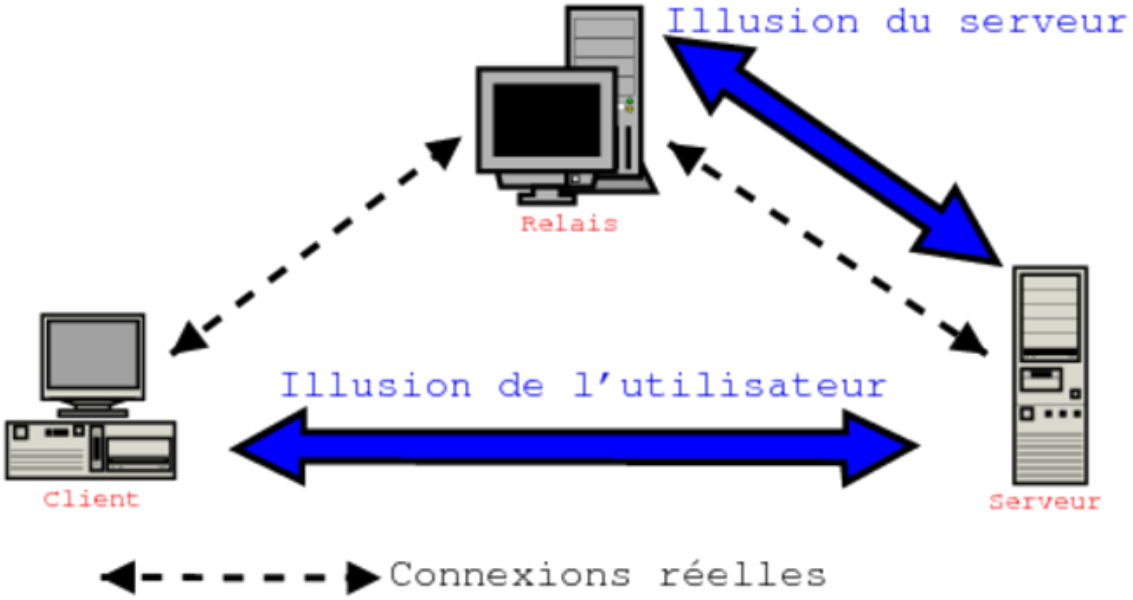

FIG. 2 – *Le serveur mandataire*

– les relais applicatifs, qui ne supportent qu'un protocole de haut niveau particulier (`http`, `smtp`,...), dont les principales caractéristiques sont :
  – relais soumis à la politique de sécurité,
  – ne supporte qu'un sous-ensemble minimal de la RFC de manière à avoir un code réduit et donc moins susceptible d'inclure une faille majeure,
  – peut inclure une fonction cache.

La notion de « reverse-proxy » est parfois employée : elle désigne les relais qui sont mandataires pour un serveur ; le cas le plus courant, donc appelé « proxy », étant de faire écran pour des clients.

Les avantages et inconvénients des serveurs mandataires peuvent être résumés ci-dessous :

– Possibilité de filtrage de contenu (scripts, applets java, ActiveX,...) et sémantique, mais qui n'est pas toujours utilisable dans la pratique, beaucoup de sites recourant massivement à ces technologies sans offrir de présentation alternative,
– interface possible avec un antivirus (protocole d'échange le plus courant : CVP développé par CheckPoint Software),
– authentification possible des utilisateurs,
– masque les adresses des machines clientes,
– ne nécessite pas de fonction de routage,
– processus en espace utilisateur [une vulnérabilité peut être moins critique si les privilèges sont bien gérés, mais vulnérabilités additionnelles du système d'exploitation sous-jacent – pile IP, journalisation,...–] qui peut être plus lent (changements de contexte noyau/espace utilisateur),
– anonymisation des clients ou des serveurs avec les mêmes limitations que le filtrage de contenu,
– nécessite un serveur relais différent par application ou de se limiter à un filtrage générique,
– il faut un processus par connexion (20 utilisateurs dont le butineur réalise 5 connexions simultanées induit 100 processus ou fils),
– ne peut être employé pour des protocoles aux spécifications fermées,
– sont difficilement transparents :
  – configuration spécifique des clients (par exemple des butineurs),
  – bibliothèque client spécifique (SOCKS),
  – « redirection » par filtre de paquets basée sur le port destination.

2.3 Le filtrage dynamique et adaptatif

Ce mécanisme se veut le meilleur des deux mondes précédents en apportant une capacité de filtrage applicative tout en restant au niveau de la couche transport/session.

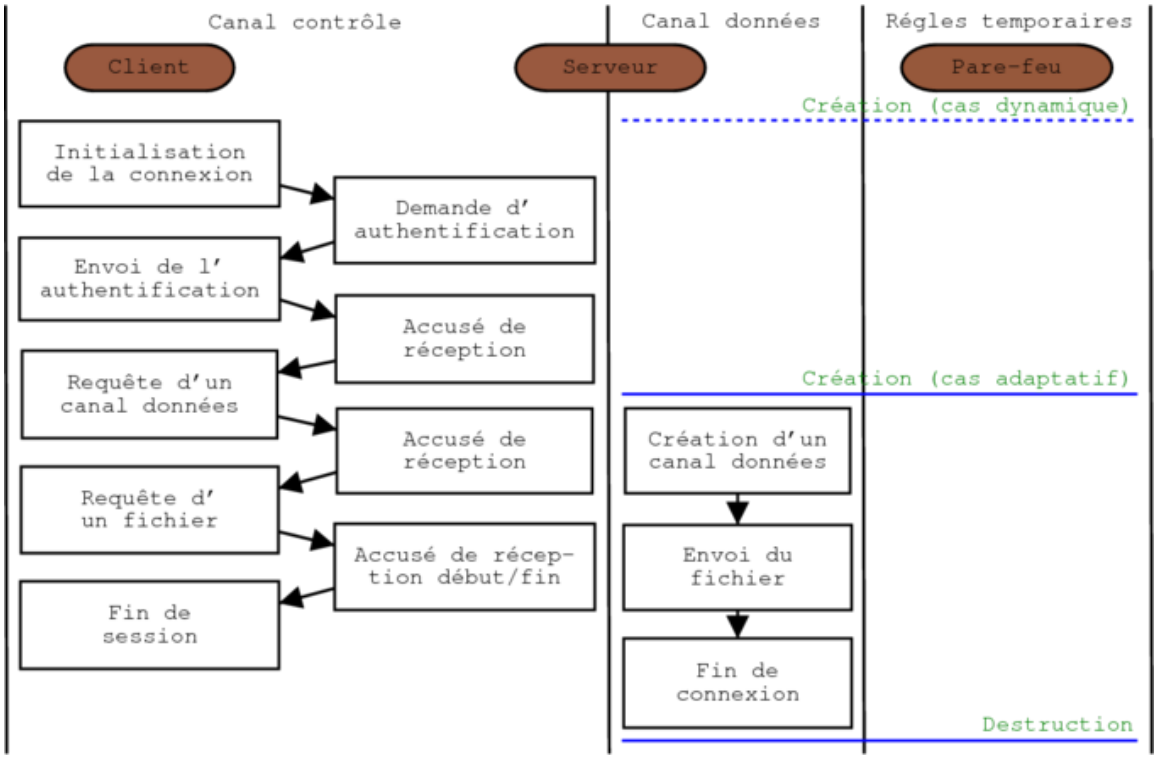

FIG. 3 – *Règles dynamiques et adaptatives pour une session FTP active*

Le filtrage dynamique ajoute la prise en compte de l'historique au simple filtrage de paquet : l'idée de base étant qu'avec un échange client/serveur si un paquet est passé dans un sens il en passera un dans l'autre (commutation de la source et destination du couple IP/port pour les paquets TCP/UDP). Diverses temporisations sont introduites : poignés de main TCP, fermeture de connexion ou de session,.... Ce mode permet de générer à la volée des règles temporaires de filtrage des paquets. Ces dernières disparaissent lorsqu'aucun paquet ne passe pendant un délai configuré ou avec la fermeture de la session en TCP (RST, FIN). En reprenant l'exemple précédent de l'accès au service `http` :

Règle dynamique : 192.168.10.0/24:1024-65535 =>*:80
Initiation d'une connexion : 192.168.10.12:1036 =>10.0.0.1:80
Règle générée temporairement : 192.168.10.12:1036 <= 10.0.0.1:80

Le filtrage adaptatif recherche, en outre, des signatures dans le segment de données des paquets afin de déterminer le type et l'état du protocole applicatif transporté et de procéder ainsi à des vérifications de cohérences (figure 3). C'est dans cette catégorie que l'on peut ranger le terme de « stateful inspection » utilisée par divers éditeurs.

Les avantages et inconvénients qui en découlent :

- moins de ports ouverts qu'avec le filtrage de paquet simple,
- analyse du contenu applicatif avec les performances et les risques du mode noyau,
- limitation de l'adaptatif aux protocoles applicatifs connus et documentés,
- à l'inverse du serveur mandataire applicatif, le filtre adaptatif peut être induit en erreur quant à l'état du protocole et donc être source de comportements vulnérables (création de règles dynamiques sous contrôle d'un client distant par exemple).

3 Les autres fonctionnalités

L'évolution des pare-feu a conduit à l'ajout de fonctionnalités dont le domaine peut sembler connexe. Parmi celles-ci ont peut distinguer :

- les réseaux privés virtuels « VPN » : les possibilités proposées vont de la création d'un « extranet » (réseau interne multi-site utilisant des tunnels chiffrants entre sites) à la sécurisation de l'accès aux ressources internes des itinérants ;

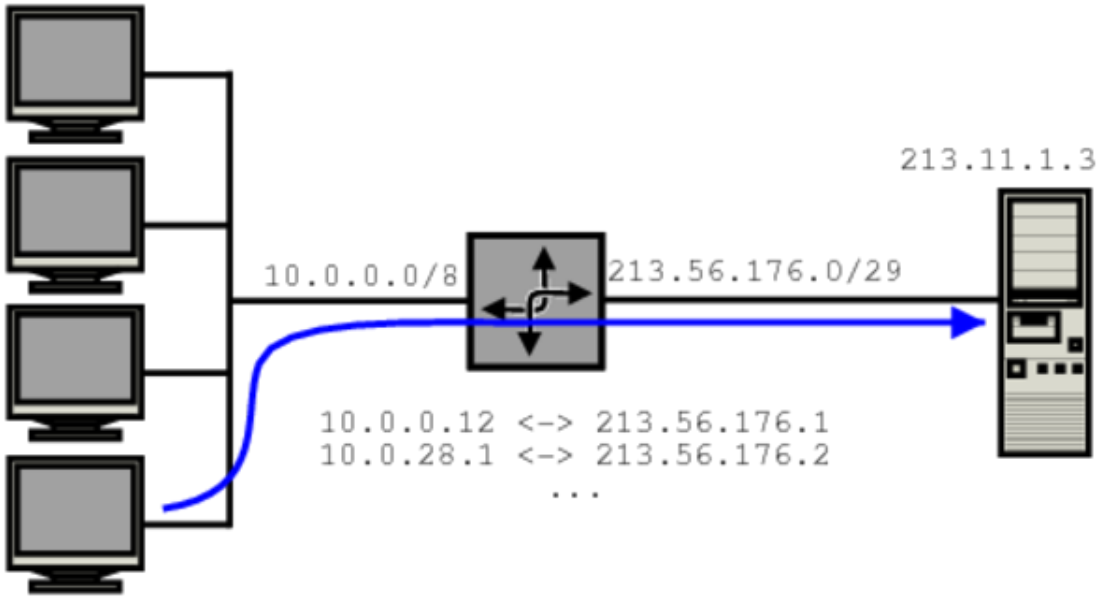

FIG. 4 – *NAT*

- l'authentification : peut être intégrée dans les relais, qui peuvent alors généralement s'interfacer avec les serveurs d'authentification les plus répandus (`Radius`, `SecurID`,...) ;
- la haute disponibilité :
  - élection du routeur : les hôtes étant supposés avoir une adresse IP de passerelle par défaut statique, la redondance est assurée par plusieurs routeurs qui partagent cette adresse et se coordonnent grâce une diffusion « multicast » utilisant le protocole 112/ip. Dans le domaine on trouve VRRP (RFC 3768) évolution du protocole HSRP de Cisco et CARP issu du monde OpenBSD et étendu aux autres BSDs mais qui n'est pas officialisé par l'IANA,
  - synchronisation des tables de filtrage : OpenBSD a développé `pfsync` pour le système de filtrage `pf` ; utilisant toujours du « multicast » mais sur 240/ip également sans officialisation de l'IANA.
- la traduction d'adresse qui consiste à réécrire les champs adresse IP source et/ou destination pour permettre le routage d'adresses privées, répondre à la pénurie d'adresses IPv4, tenter de dissimuler le plan d'adressage interne,...
- enfin certains équipements se proposent d'inclure des filtres du niveau applicatif, comme un antivirus, la recherche de contenus licencieux, une sonde de détection d'intrusion,... Cela se fait généralement au prix d'une consommation de ressources (recherches de signatures) qui peut grever les performances globales, et cela contrevient au principe de minimisation de la taille du code pour minimiser les risques de faille résiduelle.

3.1 Quelques aspects de la traduction d'adresse

3.1.1 « Network Address Translation - NAT »

Le routeur établit et maintient une correspondance entre les adresses internes (généralement privées pour éviter un effet de masquage) et une ou plusieurs adresses publiques. Cette correspondance peut être statique ou bien réalisée dynamiquement par l'équipement. Dans ce cas, le nombre d'hôtes pouvant sortir simultanément est limité par le nombre d'adresses publiques utilisables (figure 4).

3.1.2 « Port Address Translation - PAT»

Pour contourner la limitation qui précède, la correspondance n'est plus établi sur la simple adresse IP mais utilise un couple adresse IP/port (TCP/UDP) ou adresse IP/ID (ICMP). C'est une technique couramment utilisée pour connecter un réseau à l'aide d'une unique machine ayant un compte chez un FAI (figure 5).

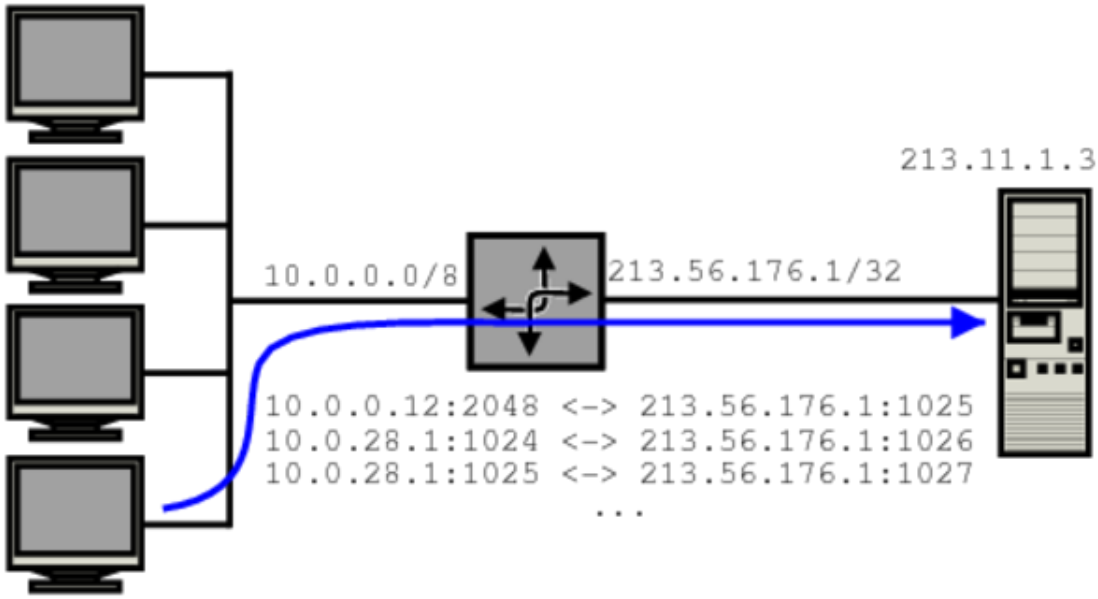

FIG. 5 – *PAT*

3.1.3 Avantages et inconvénients

- Masque le plan d'adressage interne,
- permet de connecter à Internet des adresses privées,
- autorise le changement de FAI sans modification du plan d'adressage interne,
- la « NAT » est utilisable pour faire de la répartition de charge,
- mais nécessite un filtrage adaptatif ou un relais pour les protocoles transportant des éléments de connexion dans les données (FTP, H323,...),
- est incompatible avec l'authentification de l'adresse dans IPSEC,
- est peu adaptée au routage dynamique si ce n'est avec des protocoles de synchronisation des tables comme `pfsync`,
- il y a des effets de bord sur le filtrage : il faut bien appréhender l'ordre dans lequel traduction et filtrage agissent car cela détermine l'adresse IP à prendre en compte. On ne peut, par exemple, transposer directement, à la syntaxe près, les règles `pf` d'OpenBSD qui est orienté interface en règles `netfilter` de Linux qui est orienté routage (voir figure 6).

4 Contournement du filtrage

Les pare-feu sont inadaptés pour filtrer le contenu des protocoles chiffrés de bout en bout comme SSL/TLS ou IPSEC. Certains éditeurs proposent de contourner le problème en réalisant en quelque sorte une attaque par le milieu « sous contrôle » : cette idée est en contradiction avec les principes de conception de ces protocoles et dégrade fortement leur sécurité (le pare-feu devient une IGC mais sans les procédures et la sécurité qui doivent y être associées – le logiciel du pare-feu doit avoir un accès permanent à la clef privée –, mélange des rôles – l'administration réseau devient tiers de certification –,...). Il semble plus raisonnable de déléguer une partie du filtrage sur l'hôte même, après déchiffrement.

Par ailleurs, il est toujours possible d'encapsuler les protocoles les uns dans les autres quels que soient leurs niveaux respectifs :

- création de tunnels (voir [Ce2001]). Un tunnel nécessitant un système d'encapsulation à chaque extrémité, la meilleure protection reste encore de réaliser un compromis entre un contrôle total des logiciels installés sur le parc protégé et l'éducation et la sensibilisation des utilisateurs,
- « web bugs » (images invisibles sur protocoles HTTP/HTTPS/FTP) dans les méls au format HTML, les documents `Office`,...

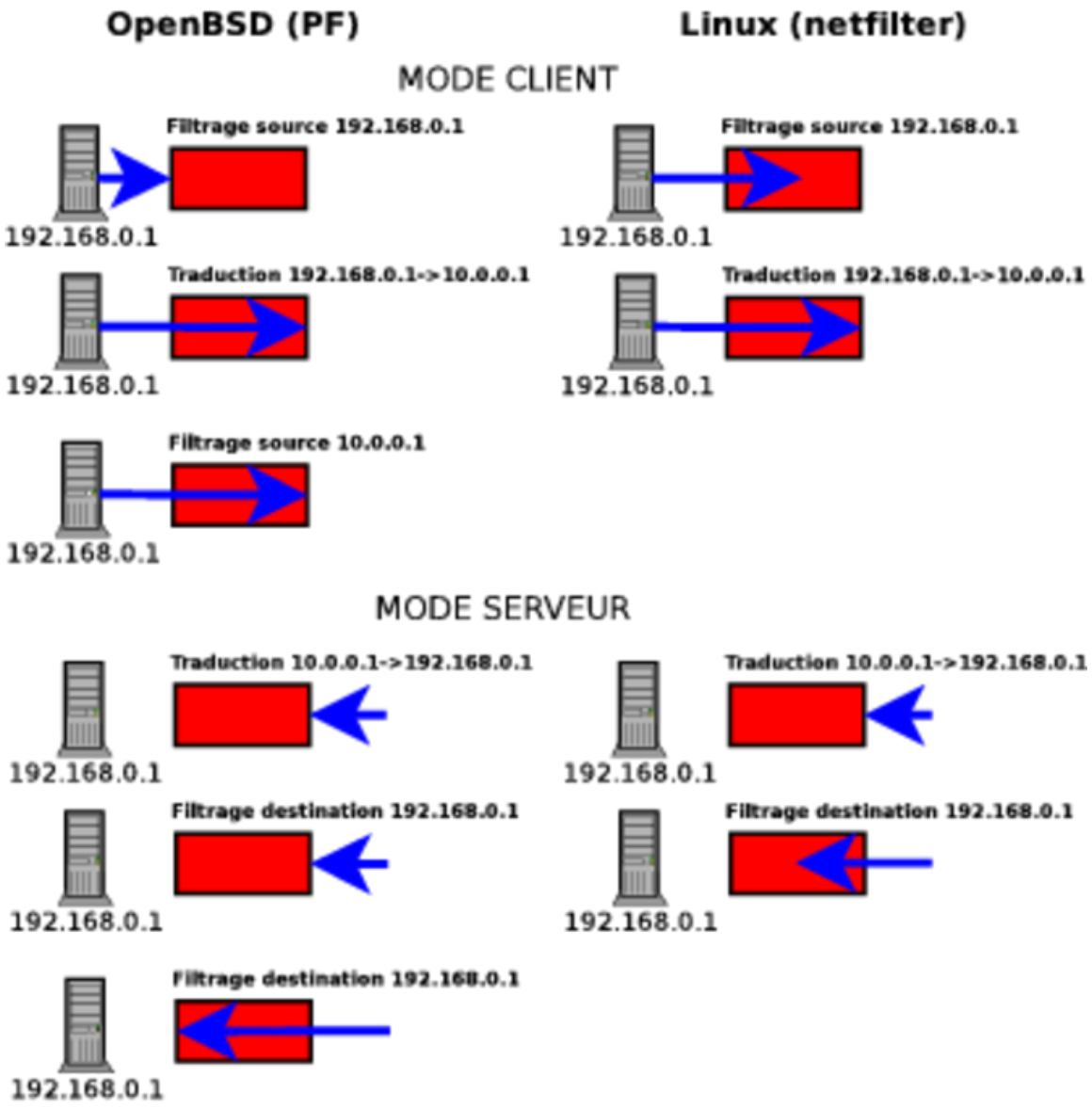

FIG. 6 – *Comparaison du couple traduction/filtrage entre netfilter et pf*

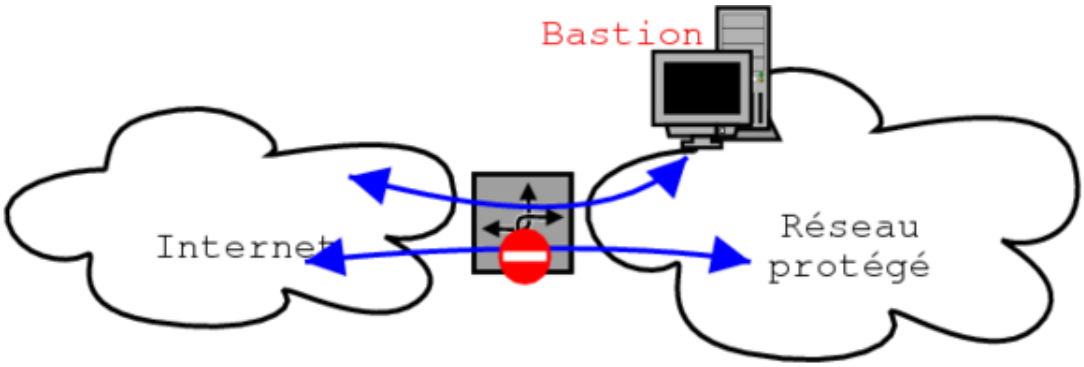

FIG. 7 – *Bastion*

– espiongiciels, chevaux de Troie utilisant `HTTP` ou inversement contournant le pare-feu local en injectant directement les paquets en couche 2,
– configuration spéciales pare-feu : lecteurs multimedias sur port 80/tcp, logiciels de téléphonie sur IP, « bouncers » par exemple IRC,
– standards visant à contourner la fonction du pare-feu : SOAP (qui peut être résumé rapidement à une interface XML pour des RPCs dans un transport HTTP). Aujourd'hui un simple filtre de paquet suffit pour bloquer l'accès aux RPCs, demain la diffusion de SOAP imposera l'emploi d'un « proxy » applicatif HTTP sophistiqué.

5 Quelques organisations classiques

5.1 L'hôte filtré

Seul un hôte, spécifiquement sécurisé, mis à jour et audité (bastion), peut accéder à l'extérieur grâce à un routeur filtrant en coupure. Les utilisateurs doivent s'authentifier dessus pour accéder aux ressources externes (figure 7).

5.2 Sous-réseau filtré

Variante du cas précédent ou l'accès au bastion est contrôlé par un second routeur filtrant. Aucun trafic ne traverse directement le sous-réseau (figure 8).

5.3 Relais isolé

Variante du bastion ou seul le protocole applicatif du relais peut être invoqué (figure 9).

5.4 Serveur relais en passerelle

Un hôte hébergeant des relais pour les applications que l'on souhaite supporter est mis en coupure et son routage désactivé. On s'assure ainsi que seuls les protocoles des relais sont transmis (figure 10).

5.5 La zone démilitarisée (« DMZ »)

C'est un sous-réseau qui n'appartient ni au réseau interne protégé, ni pour pour autant à Internet. C'est typiquement le(s) sous-réseau(x) où l'on place les serveurs publics et les relais. Ceux-ci étant susceptibles d'être compromis, le cas idéal voudrait qu'ils aient chacun leur zone démilitarisée de manière à pouvoir contenir toute intrusion (figure 11).

6 Déploiement

Le pare-feu est un équipement de bordure. Il n'a d'intérêt qu'en coupure d'un périmètre parfaitement *identifié* et *contrôlé*.

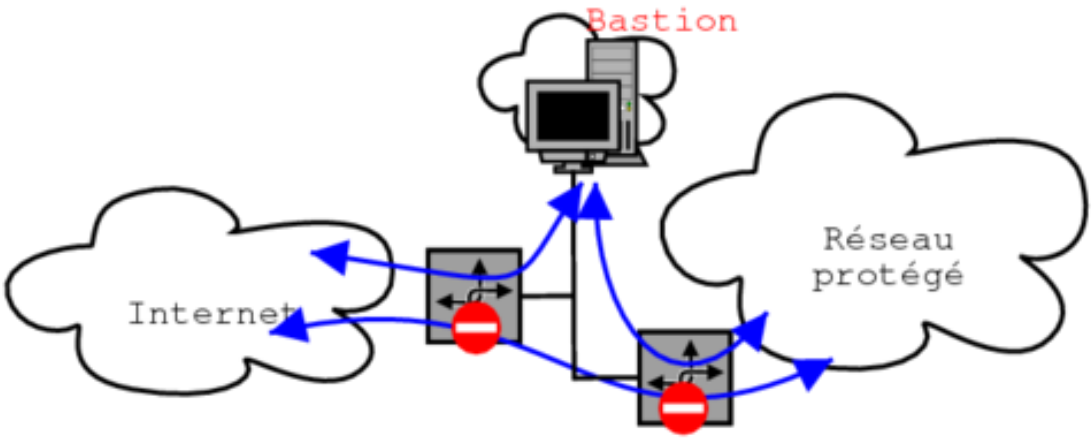

FIG. 8 – *Sous-réseau filtré*

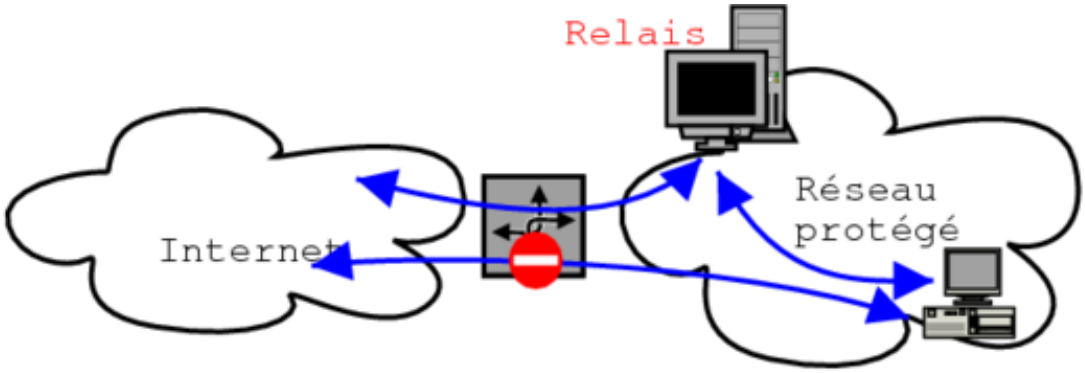

FIG. 9 – *Relais isolé*

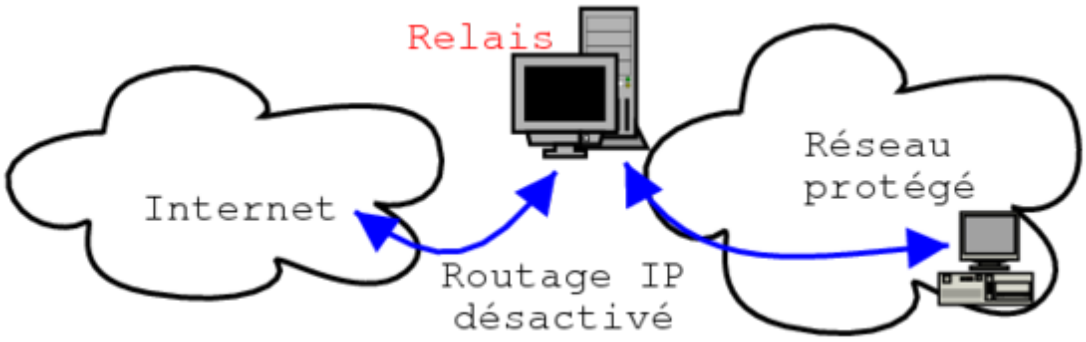

FIG. 10 – *Relais en passerelle*

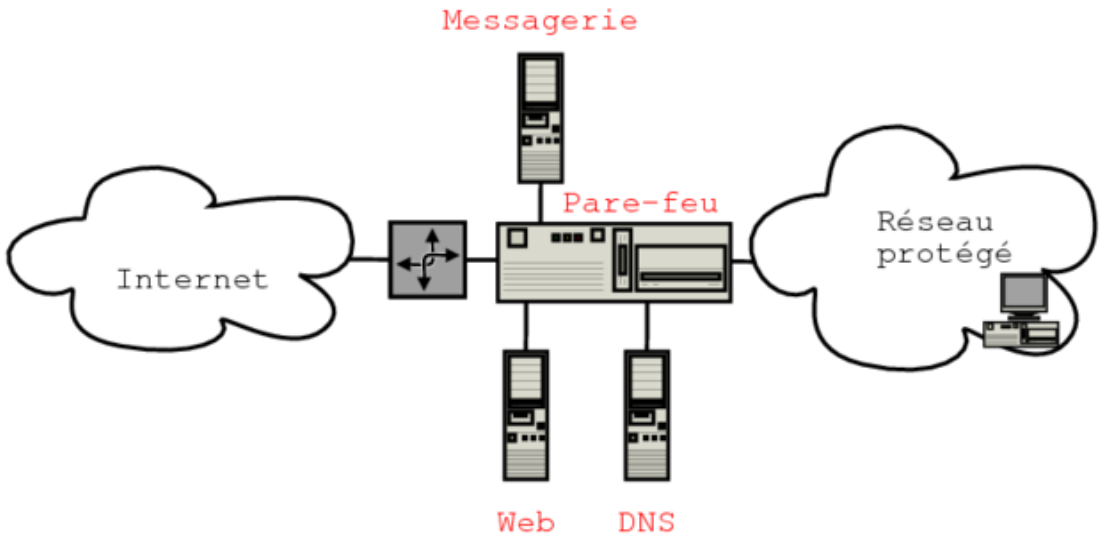

FIG. 11 – *Une DMZ par serveur ou relais*

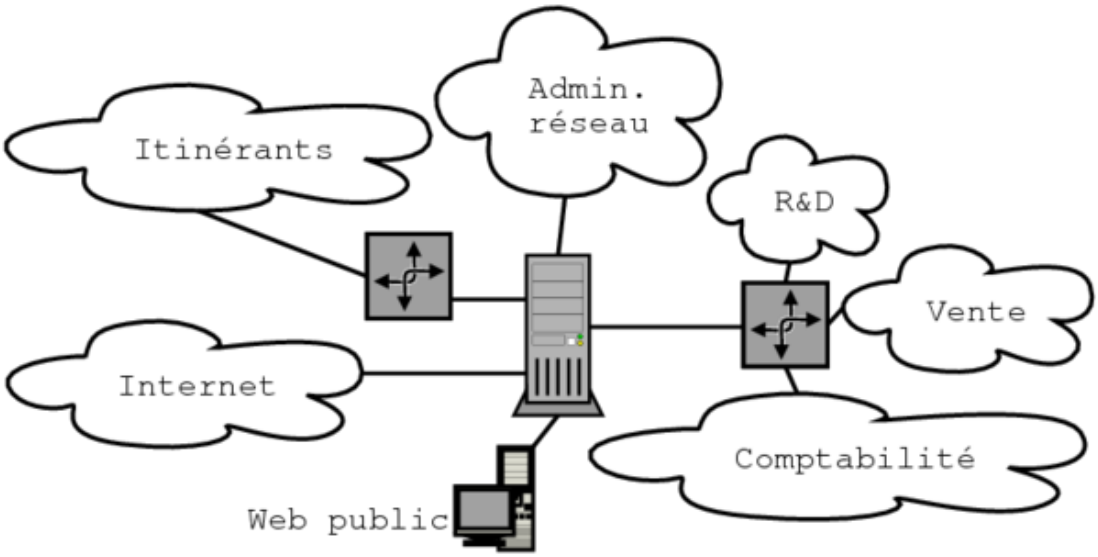

FIG. 12 – *Le cloisonnement de réseau*

De nombreux problèmes viennent altérer l'intégrité du périmètre :

- les équipements télémaintenus et connectés au réseau, telles certaines photocopieuses/imprimantes,
- les technologies sans fil,
- le déploiement de VPNs vers des hôtes dont l'intégrité ne peut être garantie (prestataires de services, itinérants,...)
- les hôtes nomades.

Devant la variété des cas possibles une solution peut être de recourir au cloisonnement de réseau ([Be1999], [Sc2000]). Il s'agit de filtrer au plus près, sur la passerelle ou le commutateur. Cela permet d'appliquer des niveaux de sécurité variables, ne nécessite pas des équipements très performants puisque chaque noeud doit gérer moins de règles. Cependant ce n'est utilisable dans la pratique qu'à l'aide d'un système d'administration centralisé (figure 12).

7 Généralités sur le filtrage TCP/IP

La politique par défaut devrait être de tout interdire puis d'ouvrir des ports en fonction du besoin. L'inverse, la fermeture en cas de problème, n'est pas une mesure de prévention.

Une attention devra apportée au mode arrêté du pare-feu (activation ou non du routage) qui peut varier pour un même produit selon les systèmes d'exploitation.

L'« anti-spoof » est une bonne pratique dans la mesure où le plan d'adressage est simple et l'« egress filtering » a l'avantage de permettre d'identifier de façon certaine le réseau protégé comme source d'un problème si l'un des hôtes venait à être compromis.

Idéalement le pare-feu devra défragmenter au niveau IP, ce qui empêchera les techniques d'évasion par recouvrement. Cependant les mêmes techniques de dissimulation peuvent être utilisées au niveau des segments TCP, phénomène qui ne peut être combattu qu'à l'aide des serveurs mandataires.

Les messages ICMPs peuvent servir à collecter des informations mais participent aussi au bon fonctionnement du protocole IP. On pourra se référer à [Th2003] pour plus détails. On retiendra que le type 3 code 4 (« icmp unreachable fragmentation needed ») est utilisé pour decouvrir la taille maximum des paquets transmissibles et que son filtrage dégrade les performances. Par ailleurs, le type 11 code 0 (« icmp time exceeded in transit ») est utile pour le routage mais est également utilisé pour détecter les ports filtrés par un équipement ; une solution peut être de filtrer ce type uniquement sur les routeurs terminaux.

8 Documentation

Références

[Be1999] Steven M. Bellovin, AT&T, Distributed Firewalls, novembre 1999
http://www.research.att.com/ smb/papers/distfw.html

[Ce2001] CERTA, Tunnels et pare-feu une cohabitation difficile
http://www.certa.ssi.gouv.fr/site/CERTA-2001-INF-003/index.html

[ch2003] Patrick Chambet, Edelweb, Linux Magazine France HS 12, Firewalls et applications Web : architecture et sécurisation, novembre 2002

[Co2005] COAST, Université de Purdue, Internet Firewalls
http://www.cerias.purdue.edu/coast/firewalls/

[CR2000] Matt Curtin et Marcus Ranum, Internet Firewalls : Frequently Asked Questions, version 10.4, 26 juillet 2004
http://www.interhack.net/pubs/fwfaq/

[Gr2003] Robert Graham, FAQ : Firewall Forensics (What am I seeing ?)
http://web.archive.org/web/20041118093912/http://www.robertgraham.com/pubs/firewall-seen.html

[JO99] Commission des télécommunications, Glossaire informatique des termes relatifs à l'informatique, Journal Officiel, 16 mars 1999
http://www.culture.gouv.fr/culture/dglf/cogeter/16-03-99-internet.html

[MB2004] Ryan McBride, Firewall Failover with pfsync and CARP
http://www.countersiege.com/doc/pfsync-carp/

[Sa2005] SANS Institute, Firewalls & Perimeter Protection
http://www.sans.org/rr/catindex.php?cat_id=21

[Sc99] Hervé Schauer, Hervé Schauer Consultants, The future of the firewall, 18 août 1999
http://www.hsc.fr/ressources/presentations/df/index.html.en

[Sc2000] Hervé Schauer, Hervé Schauer Consultants, Distributed Network Security, 23 mars 2000
http://www.hsc.fr/ressources/presentations/dns/index.html.en

[Sm2001] Gary Smith, SANS Institute, A Brief Taxonomy of Firewalls – Great Walls of Fire, 18 mai 2001
http://www.giac.org/practical/gsec/Gary_Smith_GSEC.pdf

[Th2003] Rob Thomas, ICMP Packet Filtering v1.2
http://www.cymru.com/Documents/icmp-messages.html

[WCP2002] John Wack, Ken Cutler, Jamie Pole, NIST, Guidelines on Firewalls and Firewall Policy
http://csrc.nist.gov/publications/nistpubs/800-41/sp800-41.pdf

Gestion détaillée du document

10 janvier 2006 version initiale.