Atouts et limites du modèle de sécurité du pare-feu personnel

Cédric Blancher

Cartel Sécurité
blancher@cartel-securite.fr
http://www.cartel-securite.fr/

Résumé Le développement des accès haut-débit pour les particuliers a ouvert un nouveau marché en matière de sécurité. Les spécificités des accès "grand public" ont conduit à l'émergence d'un nouveau type de produit : le firewall personnel.
Si le concept est intéressant et si ce type outil apparaît aujourd'hui comme un composant incontournable de protection, il est essentiel d'en apprécier les atouts comme les limites. L'article vise donc à présenter ce qu'est un firewall personnel par ses concepts fondamentaux, puis en analyser les faiblesses dans son contexte de fonctionnement. Nous pourrons ainsi en définir les limites pour parvenir à une utilisation éclairée et efficace.

1 Introduction

Le firewall personnel, ou pare-feu personnel[1], vise à offrir aux connexions Internet dites personnelles, à savoir un poste directement relié à Internet, segment de marché jusqu'alors oublié des éditeurs, un logiciel de sécurité réseau simple et efficace.

En effet, jusqu'à l'apparition des premiers produits de ce type, les logiciels et autres équipements de sécurité réseau (i.e. les firewalls) visaient à protéger des réseaux entiers situés en amont. L'utilisation d'un tel dispositif pour protéger un poste isolé supposait la constitution d'un réseau local de manière à placer ce dernier en amont du firewall. La mise en place efficace de ce type de configuration demande d'une part des moyens matériels, donc financiers, et d'autre part des compétences certaines pour configurer convenablement le tout. Ce type d'infrastructure de sécurité n'est donc pas adaptée aux besoins de sécurité des particuliers, c'est-à-dire l'utilisateur moyen d'Internet à domicile. Il convenait donc de trouver autre chose.

2 Les bases du firewall personnel

2.1 Description du contexte

La spécificité des connexions monopostes est leur caractère essentiellement "client". Une telle connexion n'est en effet pas adaptée à une utilisation de type

[1] par soucis de simplicité, nous utiliserons le terme plus courant de "firewall"

“serveur”, ne serait-ce que parce que les adresses IP affectées par les fournisseurs d'accès sont souvent dynamiques. Malgré tout, cette dimension doit être prise en compte, puisque cependant possible.

Ce qui caractérise une utilisation cliente, c'est le manque, voire l'absence, de services accessibles depuis Internet, donc de points d'entrées susceptibles d'être exploités par un pirate pour compromettre le poste. Il en résulte qu'en l'absence de points d'accès directs, la principale menace qui pèse sur notre utilisateur est la réception par des moyens divers (courrier électronique, téléchargements, etc.) de programmes type “chevaux de Troie” conduisant à la mise en place de logiciels intrus, commes des espions (spyware) ou des portes dérobées (backdoor) par exemple.

Le rôle d'un firewall dans ce contexte sera double. Il devra d'abord fermer tous les accès au poste protégé, évitant ainsi l'utilisation distante de services ouverts dans les installations par défaut. Ensuite, il devra empêcher l'envoi ou la réception de données via le réseau par des programmes non autorisés pouvant entraîner des fuites d'information ou la compromission du poste par un intrus qui pourra l'utiliser à des fins détournées. C'est cette dernière tâche qui sera la plus importante et la plus problématique.

2.2 Utilisation d'un système de filtrage de paquets classique

Cette tâche peut être réalisée avec un système de filtrage que l'on qualifiera de classique, c'est à dire s'appuyant uniquement sur les caractéristiques des flux réseau locaux. On va donc filtrer les paquets entrant et sortant sur la base des éléments protocolaires qu'ils contiennent. Il doit être noté que pour une station terminale, ce type de filtrage n'apporte rien par rapport à un filtrage effectué par un équipement tiers, puisqu'il ne s'appuie que sur les caractéristiques des flux observés.

Supposons un poste A relié à Internet. Si on veut restreindre son utilisation du réseau à une utilisation du Web, nous devons mettre en place ce type de règles :

```
Entrée :
   source :           inconnue
   destination :      IPa
   protocole :        TCP
   port source :      80
   port destination : > 1023
   drapeau TCP :      ! SYN

Sortie :
   source :           IPa
   destination :      inconnue
   protocole :        TCP
   port source :      > 1023
   port destination : 80
```

```
   drapeau TCP :      -
```

Pour peu que nous utilisions un firewall à états, nous pouvons simplifier notre jeu de règles à :

```
Sortie :
   source :           IPa
   destination :      inconnue
   protocole :        TCP
   port source :      > 1023
   port destination : 80
```

Le problème principal réside dans le filtrage de la plage des ports non privilégiés (1024-65535). En effet, ce sont les ports de cette plage qui sont utilisés par les applications clientes comme source, sans qu'on puisse dans ce contexte les distinguer entre eux. En effet, n'importe quel port de cette plage peut servir à une application donnée. À l'inverse, n'importe quelle application est susceptible d'utiliser un port donné de cette plage. Les choses se compliquent encore lorsqu'on doit prendre en compte des protocoles applicatifs comme FTP ou H323 qui incluent une négociation de ports pour l'établissement de connexions annexes. Si on considère l'exemple de FTP, dont le mode de transfert de données actif implique la connexion du serveur FTP sur le client, vers un port de la plage non privilégiée, on s'aperçoit que l'utilisation d'une telle application entraine l'autorisation de connexion vers n'importe quel port de cette plage, ce qui réduit considérablement l'intérêt de notre filtrage. En effet, si nous considérons l'utilisation du FTP par notre poste A, sur un firewall à état ne prenant pas en compte les spécificités de FTP, nous aurons un jeu de règles du type :

```
Entrée :
   source :           inconnue
   destination :      IPa
   protocole :        TCP
   port source :      20
   port destination : > 1023

Sortie :
   source :           IPa
   destination :      inconnue
   protocole :        TCP
   port source :      > 1023
   port destination : 21

   source :           IPa
   destination :      inconnue
   protocole :        TCP
   port source :      > 1023
   port destination : > 1023
```

Un tel jeu de règle autorise n'importe quelle machine distante à se connecter à un port non privilégié du poste A pourvu qu'elle utilise le port 20 comme source, et à n'importe quelle application locale de se connecter à un port non privilégié distant. Un filtrage à état prenant capable de gérer les spécificités du protocole FTP apporterait une solution efficace à ce problème. Malheureusement, les firewalls personnels du marché n'incluent pas ce type de fonctionnalité.

2.3 Filtrage par application

La conclusion de ces constatations est que le filtrage de paquets répond difficilement à notre problématique. À moins d'utiliser un système de filtrage à état (stateful) complet, il nous faudra trouver d'autres critères.

L'élément que nous pouvons examiner lorsqu'on filtre des flux locaux est l'application qui les génère ou les reçoit. En effet, si on part du principe qu'on est capable d'associer un comportement donné à une application donnée (i.e. un client FTP fait du FTP, un MUA fait du SMTP, du POP, de l'IMAP, etc.), on est non seulement capable de rendre plus efficace notre système de filtrage, mais aussi d'en simplifier grandement la configuration. Il est important pour la suite de bien noter que ce principe constitue la base du fonctionnement du firewall personnel.

En insérant des points de lecture au sein de la pile TCP/IP du système d'exploitation, nous pouvons savoir par quelle application est émis un paquet, ou à quelle application est destiné un paquet reçu. En établissant une liste des applications autorisées (ou non) soit à initier un flux, soit à se mettre en écoute, nous sommes en mesure d'apporter un élément de réponse probant à notre problème.

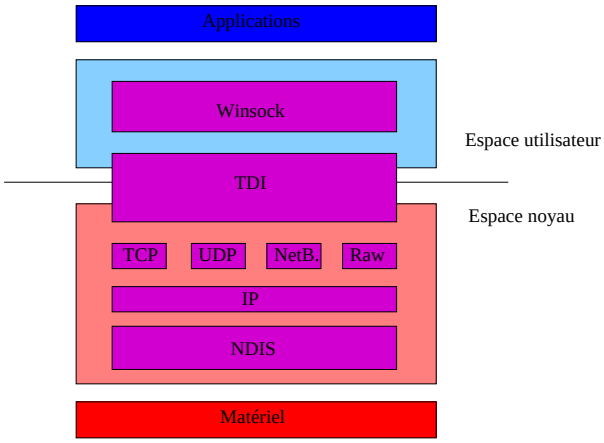
3 Le firewall personnel dans la pratique

Dans la pratique, le firewall personnel est un système dont le rôle principal sera de filtrer les demandes d'ouverture de socket par les applications du système. La plupart des produits entrant dans cette catégorie sont à destination des systèmes d'exploitation grand public, à savoir Microsoft Windows 95, 98, Me, NT, 2000 et XP ou encore MacOS d'Apple. Des produits ciblés pour les entreprises sont aussi proposés, sur les mêmes plateformes.

De tels outils ne sont pas proposés pour les Unix du marché. Certains de ceux-ci disposent de solution de filtrage de paquets performantes, comme Netfilter, IP Filter ou encore Packet Filter. Bien que n'entrant pas dans cette catégorie d'outils, nous considèrerons brièvement le cas de Netfilter, le filtre de paquets Linux, dans la mesure où il inclue des fonctionnalités de filtrage similaires. Mais l'essentiel de cet article portera sur les architectures Windows.

3.1 Caractéristiques principales

Un firewall personnel nécessite l'introduction dans la pile TCP/IP (cf. Tab. 1) de deux points d'entrées (hook).



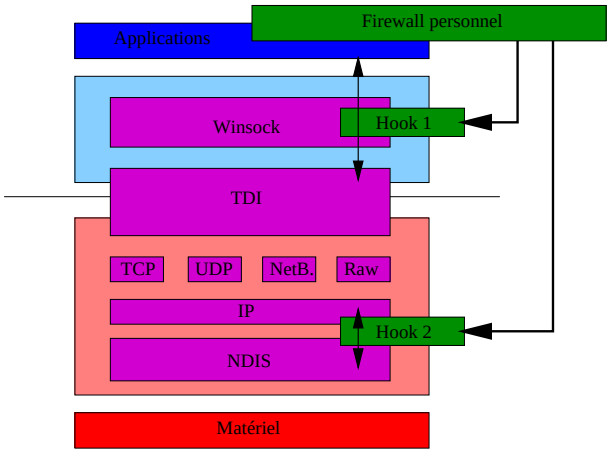
Tab. 1. Pile réseau générique Windows

Le premier point se situe au niveau de l'API Winsock (cf. Tab. 2). Cette API sert d'interface entre l'espace utilisateur, donc les applications, et la couche TDI (Transport Driver Interface) qui sert d'interace d'accès aux protocoles de niveau 4 (typiquement TCP et UDP) en espace noyau. Ce point filtre les demandes d'ouverture de sockets de communications par les applications, quel que soit leur type (TCP, UDP ou RawIP sous Windows XP) et leur mode (client ou écoute).

Le second (Cf. Tab. 2) vient se positionner entre la couche IP et la couche NIDS (Network Driver Interface Specification) qui réalise l'interface unifiée pour les pilotes de périphériques de communication et qui intègre les protocoles de niveau 2 comme PPP, PPPoE ou encore PPTP. C'est le point de filtrage des paquets qui passent de la couche liaison à la couche IP, quel qu'en soit le sens.

La majorité des produits du marché gèrent les états, au moins pour TCP. Ils sont donc capables de savoir si un paquet donné appartient ou non à une connexion déjà établie. Certains ne savent pas gérer les états pour UDP. Dans la mesure ou aucun paramètre ne nous permet de discriminer le sens d'un flux UDP, cette absence est un manque lourd qui entraine des ouvertures importantes dans le jeu de règles. Aucun ne possède de gestion efficace des erreurs ICMP. Si cela ne pose pas vraiment de problème de sécurité, cette mauvaise gestion conduit à des alertes et des refus de messages d'erreur valides que l'outil n'est pas capable d'associer au flux qui les a généré. Il s'en suit parfois des problèmes de connectivité, en particulier pour les découvertes de PMTU. Enfin, aucun ne gère de protocole applicatif faisant intervenir une négociation de flux annexe, comme FTP ou H323.

La méthode de configuration est sensiblement la même pour tous ces outils : il s'agit d'une méthode d'apprentissage. Chaque fois qu'une application demande à accéder au réseau, le firewall vérifie sa présence dans une liste de contrôle



Tab. 2. Positionnements des hooks

recensant les applications faisant l'objet d'une décision. S'il la connait, il applique la décision associée. S'il ne la connait pas, il demande à l'utilisateur ce qu'il doit faire. La nouvelle application est alors introduite dans la liste de contrôle. Cette approche présente deux avantages. D'une part, elle permet de simplifier grandement la tâche de configuration, rendant l'outil prêt à l'utilisation, sans aucune spécification initiale. D'autre part, ce mécanisme sert de système de remontée d'alerte : tout accès inconnu est notifié à l'utilisateur. Il en va souvent de même pour les paquets reçus. Chacun fait l'objet d'une notification jusqu'à ce qu'une règle, plus ou moins précise, soit établie par l'utilisateur pour le traiter.

Les applications sont reconnues par le chemin absolu de l'exécutable correspondant. On évite ainsi qu'un binaire portant le nom d'une application autorisée puisse initier des flux réseau. En outre, ce chemin est assorti d'une somme de contrôle, basée sur MD5, pour éviter le remplacement d'un exécutable valide par une version vérolée contenant du code malicieux.

On peut classer ces produits en deux catégories, selon les fonctionnalités de filtrage qu'ils offrent.

3.2 Produits basiques

Les produits basiques ne permettent que l'autorisation ou l'interdiction d'applications. Une fois une application autorisée, elle peut initier n'importe quel type de flux. Ce type d'outil est le plus simple à utiliser puisqu'il ne demande pratiquement aucune connaissance de la part de l'utilisateur. C'est cette facilité d'utilisation qui place souvent ce type d'outil en tête des comparatifs.

D'un point de vue purement technique, cette approche suppose que l'utilisateur connaisse, au moment où il autorise une application donnée, tous les types de flux qu'elle peut être amenée à générer ou recevoir. On retrouve là le prin-

cipe de base évoqué plus haut. Comme nous le verrons plus loin, cette condition est loin d'être remplie, posant dès lors un potentiel problème de sécurité. En effet, il est des applications auxquelles nous ne pouvons pas faire aveuglément confiance. Nous verrons plus loin que Internet Explorer, par exemple, peut être utilisé, de manière légitime ou non, par des applications tierses pour initier des flux réseau. C'est une de ses nombreuses fonctionnalités. Un trojan pourrait tout à fait exploiter ce type d'interface pour passer outre les restrictions imposées par le firewall personnel en profitant de celles dont jouit ce navigateur.

3.3 Produits avancés

Les produits avancés permettent d'associer à une application le type de flux qu'elle est autorisée à initier ou recevoir. Cette approche est certes plus fine et efficace en ce qu'elle permet de contraindre le type d'utilisation d'une application donnée, mais rend la configuration plus difficile en ce qu'elle réclame des connaissance techniques de la part de l'utilisateur.

Cependant, les applications versatiles ou mettant en œuvre des protocoles incluant des flux négociés demandent des autorisations extrêmement larges qui rendent ce type de restrictions inutiles en l'absence d'un filtrage à états complet comme vu plus haut. Considérons un navigateur. Un tel outil propose au minimum de support de HTTP et de FTP en mode passif. La seule autorisation de ce dernier protocole lui donne le droit de se connecter, en TCP, à tout port non privilégié de n'importe quel adresse (on ne connaît pas à priori les adresses de serveurs qu'on consulte) depuis n'importe quel port non privilégié local. Autant dire que ce navigateur peut faire ce qu'il veut sur le réseau. C'est un problème épineux si on considère qu'Internet Explorer tombe dans cette catégorie et qu'il peut être amené à servir de relai pour d'autres applications...

3.4 Linux/Netfilter

Netfilter [1], le système de filtrage des noyaux Linux, n'est pas un firewall personnel. Il inclue cependant une concordance, "owner", lui permettant d'intégrer des éléments de l'espace applicatifs dans ses règles :

- UID et/ou GID de l'utilisateur à qui appartient le processus générant le flux ;
- PID et/ou SID du processus générant le flux ;
- nom de la commande dont est issu le processus qui génére le flux.

Si la fonctionnalité permettant de spécifier le nom d'une commande parait intéressant, elle est malheureusement assez limitée. Le nom qui est utilisé par cette concordance est celui qui apparait dans la liste des processus. Or ce dernier peut être aisément modifié, par exemple en établissant tout simplement un lien symbolique portant le nom d'une commande autorisée vers un exécutable interdit. Il conviendrait donc de durcir cette fonctionnalité pour véritablement reconnaître un exécutable et non une commande.

Cependant, cette concordance offre des possibilités intéressantes en ce qu'elle permet de mettre en place des politiques de filtrage différentes selon les utilisateurs de la machine. En particulier, nous sommes capables de filtrer de manière très précise des applications s'exécutant sous un UID spécifique, ce qui est le cas de nombreux serveurs, et ainsi compliquer la tâche d'un intrus qui serait parvenu à en exploiter une vulnérabilité.

4 Limites du concept et contournement

Le concept du firewall personnel est intéressant par le type de filtrage qu'il propose. Il permet ainsi de simplifier fortement la sécurisation d'un poste isolé connecté à Internet. Cependant, il est possible de contourner les restrictions mises en place.

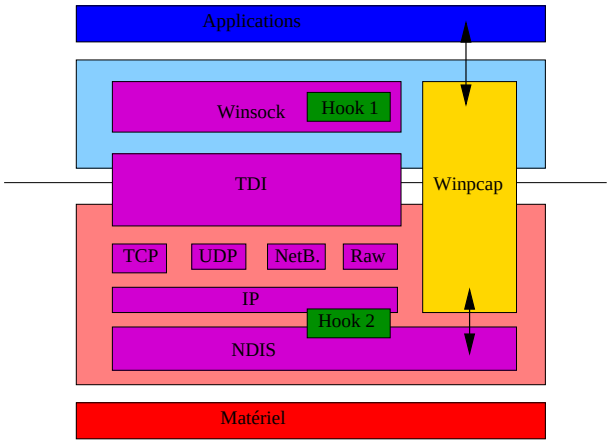
Ces méthodes d'utilisation d'applications tierses autorisées reposent peu sur de véritables failles des firewalls, mais plus sur des faiblesses du système sous-jacent. Cependant, celles-ci suffisent à remettre en cause le fondement même de leur fonctionnement basé sur une confiance dans le comportement des applications autorisées.

4.1 Accès aux couches basses

La méthode la plus simple pour contourner un filtre de paquet local est d'injecter son flux à un niveau inférieur à celui auquel il opère. Des bibliothèques permettent en effet d'injecter directement au niveau de la couche liaison des trames complètes. Parmis ces bibliothèques, on pourra citer les libnet et libdnet en environnement Unix et la winpcap [2] en environnement Windows. Des bibliothèques de capture de trames permettrons de lire le trafic reçu pour en extraire des informations. On pourra citer la libpcap sous Unix et la winpcap sous Windows.

Cette dernière vient en effet s'intégrer directement au dessus de la couche NIDS, en parallèle des couches TCP/IP et TDI, et dispose de ses propres API (cf. Tab. 3) pour proposer aussi bien la capture que l'injection de paquets. De part sa position, elle évite tous les points de contrôle mis en place par un éventuel firewall personnel. Il faut bien différencier cette bibliothèque des sockets RawIP de Windows XP, qui restent contrôlables par un firewall personnel, puisque placées entre l'API Winsock et la couche IP.

Si l'appel à ces bibliothèques sous Unix suppose les droits superutilisateur, la situation est extrêmement différente sous Windows. D'une part parce que les Windows 95, 98 et Me ne possèdent pas de notion d'utilisateur et de droits. De fait, n'importe quelle application est susceptible d'accéder à de telles fonctionnalités. D'autre part parce que sous les version de Windows qui gèrent des droits (NT, 2000, XP), les droits d'administration ne sont nécessaires que pour l'installation de la bibliothèque. Une fois celle-ci installée, n'importe quel utilisateur à la possibilité d'y faire appel. Or, certaines machines possède cette bibliothèque, de part de vieilles installations ou pour des outils de monitoring réseau.



Tab. 3. Positionnement de la couche winpcap

Fort de la possibilité d'accéder à ces deux types de bibliothèques, nous sommes en mesure de développer des applications capables de communiquer via le réseau sans jamais passer par les mécanismes de vérification du firewall, voire de rediriger les flux d'applications classiques dans un canal caché de ce type. L'outil WhiteCane [3,4], par exemple, permet de rediriger un flux UDP banal vers n'importe quelle application écoutant sur un port local.

Sous certains Unix, il est également possible d'insérer des modules permettant à des flux spécifiques d'échapper au couches de filtrage de paquets. Le module nfbypass [5] permet ainsi de faire échapper au contrôle de Netfilter tout paquet ayant pour source ou destination une adresse IP arbitraire.

4.2 Accès aux applications autorisées

Une technique simple consiste à exploiter des services mis à disposition par certaines applications via des liens OLE ou des méthodes ActiveX. On peut ainsi se servir de Microsoft Internet Explorer pour mandater des flux. Cette approche peut se révéler particulièrement intéressante. D'abord parce que Internet Explorer est autorisé dans l'immense majorité, sinon la totalité, des configurations de firewall personnel. Ensuite, parce que cette application ne peut pas être restreinte comme nous avons pu le constater plus tôt. Enfin, lorsque le passage par un mandataire (proxy) est nécessaire, Internet Explorer, configuré en conséquence, l'utilisera automatiquement, authentification comprise si celle-ci a déjà été effectuée une fois durant la session. On se reportera à la présentation de JAB [6].

La technique la plus efficace consiste certainement à exploiter des "fonctionnalités" des environnements Windows NT/2000/XP qui permettent à un utilisateur d'introduire du code, via une DLL, dans l'espace mémoire d'un processus existant, comme pour un plugin, via la directive CreateRemoteThread()

[7,8,9]. Cette insertion nécessite un privilège particulier, SeDebug, que tous les utilisateurs du système possèdent par défaut. Cette technique permet donc à n'importe quelle application d'exécuter des commandes dans le contexte d'une autre, et ainsi d'en usurper les privilèges au niveau du firewall personnel pour initier ou recevoir des flux réseau. Un exploit générique a été publié sur Bugtraq pour illustrer cette méthode [10].

4.3 Attaque du firewall

Si nous disposons de droits suffisants, ce qui est le cas des environnements Windows 95/98/Me, nous pouvons tenter d'attaquer le firewall personnel directement. Parmi les technique possibles, nous pourrons citer la désactivation pure et simple du processus de protection (cf. virus BugBear [11]) ou l'altération des fichiers de configuration [12]. Il faut bien se garder à l'esprit qu'une application autorisée n'est finallement caractérisée que par trois éléments dans un fichier de configuration non signé :

- le nom de l'exécutable ;
- son chemin absolu sur le système de fichier ;
- une somme de contrôle MD5.

Du fait qu'une somme MD5 ne constitue pas une signature, elle peut-être recalculée par n'importe qui. Dès lors, nous sommes capable de modifier une entrée de plusieurs manières :

- modification du binaire et calcul d'une nouvelle somme MD5 ;
- modification du chemin et calcul d'une nouvelle somme MD5 ;
- modification du nom et calcul d'une nouvelle somme MD5 ;
- etc.

Nous pourrons aussi utiliser la technique d'injection de DLL pour en modifier le comportement si nous possédons les privilèges adéquats. Ceci nous garantit en plus la furtivité, puisqu'aucun signe ne trahira le canal que nous venons d'ouvrir. Enfin, il ne faut pas oublier que le firewall est un service comme les autres qui peut lui aussi faire l'objet de vulnérabilités pouvant conduire à la compromission du poste [13].

4.4 Problèmes de configuration

Les firewalls personnels souffrent en outre de problèmes de configuration qui altèrent leur efficacité. D'une part, tous les produits n'offrent pas, comme nous avons pu le voir précédemment, les mêmes fonctionnalités en terme de configuration. D'autre part, leurs soucis de simplicité et d'ergonomie entrainent parfois des problèmes de filtrage.

- L'impossibilité de limiter une application autorisée permet à un programme intrus d'obtenir un accès complet au réseau en exploitant une technique d'injection de flux par l'intermédiaire d'une application autorisée. C'est une faiblesse majeure des outils basiques qui ne peuvent donc pas répondre à une problématique de sécurisation avancée.

- L'impossibilité de gérer d'autre protocoles que TCP, UDP et ICMP permet à un programme intrus de créer des canaux en utilisant des protocoles que le firewall pourrait ne pas prendre en charge.
- La gestion des états souvent incomplète implique la mise en œuvre de jeux de règles compliqués et faibles, en particulier dans le cas de protocoles applicatifs impliquant des négociations de flux et celui de la gestion des erreurs ICMP.
- Le jeu de règles d'origine (configuration par défaut) laisse des ouvertures permettant le passage de flux vers l'extérieur, en particulier pour les règles de gestion de DHCP et DNS, comme l'ont montré des alertes, dont une récente, sur des produits commerciaux [14,**?**]. En outre, les règles de base ouvrent toujours l'interface de loopback. Cette dernière peut être exploitée par des outils pour servir de relai de communication entre un flux autorisé sur le réseau et une application à priori interdite écoutant sur l'interface de loopback [3,4], implémentant ainsi un canal caché.
- La méthode de configuration peut elle-même entrainer la mise en place de règles trop laxistes par manque d'information des messages ou par simple lassitude de l'utilisateur devant l'avalanche de popups à laquelle il doit faire face. En effet, il doit valider toute nouvelle application, revalider toute application patchée (cf. somme de contrôle) et prendre une décision pour tout paquet reçu non attendu.

4.5 Le discours marketting

Comme tout produit commercial, les firewalls personnels sont proposés à grands renforts de slogans percutants qui promettent la "protection optimale contre les pirates, les vers et le trojans". Comme nous l'avons vu précédemment, il faut grandement relativiser tout cela. En effet, ce discours ne tient pas compte des faiblesses des systèmes que ces produits sont sensés protéger. Il est par exemple difficile de promettre la protection contre les trojans pour un système sur lequel le premier processus venu peut en tuer un autre, le firewall en tête, comme l'ont appris à leurs dépends les utilisateurs infectés par BugBear [**?**],...

De plus, le discours semble volontairement alarmiste. Les messages d'alertes et les journaux d'activité utilisent souvent des termes du domaines de l'intrusion. Le but est vraisemblablement double, permettant d'une part à l'outil d'auto-justifier une certaine utilité et invitant d'autre part l'utilisateur à investir dans la version payante bien mieux founie en fonctionnalités.

5 Les bases d'une mise en œuvre efficace

5.1 Une utilisation éclairée

Lorsqu'on met en place une solution de sécurité, il est vital d'en maîtriser tous les aspects. L'utilisateur doit donc, pour en tirer le meilleur parti, savoir ce qu'est un firewall personnel, ce qu'il fait et ne fait pas, de manière à évaluer au mieux le niveau de protection fourni.

En outre, l'utilisateur doit être conscient que les concepts manipulés par ce type d'outil ne sont pas simples au point de ne pas avoir à les connaître. Il doit donc posséder un minimum de connaissances pour effectuer une configuration efficace.

5.2 Gestion efficace des droits

Pour assurer à l'outil un socle système adéquat, une gestion efficace des droits utilisateur est indispensable. La première mesure à prendre est de proscrire tout système n'implémentant pas les notions d'utilisateur et de droits d'accès. Comme nous l'avons vu précédemment, un tel système ne peut pas fournir les mécanismes pouvant empêcher le contournement trivial d'un firewall.

La gestion des droits doit être configurée conformément à la règle du moindre privilège. Il y a fort à parier que les utilisateurs d'un système bureautique n'aient pas besoin des privilèges d'administration ou SeDebug par exemple... Sur un système comme Windows XP en version Family, tous les utilisateurs ont les droits d'administration par défaut, c'est dommage.

Les applications autorisées par le firewall doivent être restreintes au maximum en terme de protocoles et de jeu de ports. Ces restrictions devront être reprises au niveau d'éventuels équipements de sécurité situés en aval dans l'infrastructure réseau.

5.3 Une bonne hygiène de vie

Enfin, de bonnes habitudes s'imposent et constituent la base d'une utilisation sûre :

- travailler avec compte non privilégié ;
- ne pas ouvrir n'importe quel document ;
- mettre en place un bon antivirus et le tenir à jour ;
- tenir son système et ses applications à jour ;
- être attentif aux messages d'alertes des applications.

Les voies les plus efficaces pour introduire un trojan sont d'une part le génie social, basé sur l'exploitation de la crédulité des utilisateurs, et d'autre part les failles des applications courantes comme Internet Explorer, Outlook et Outlook Express, la combinaison des deux se révélant souvent catastrophique.

6 Conclusion

Comme nous avons pu le voir, le concept de firewall personnel est intéressant, mais souffre de limitations parfois fortes. Certaines sont dûes à l'implémentation des outils, d'autres au système sur lesquels ils s'exécutent. Quoi qu'il en soit, ce type d'outil n'en perd pas pour autant tout intérêt et se révèle un composant indispensable de protection des postes directement connectés à Internet.

Cependant, il ne peut assurer à lui seul la protection intégrale d'un poste informatique quel qu'il soit, loin de là. Il doit impérativement être complété

d'autres mécanismes de sécurité. Mais plus que tout dispositif technique, ce sont surtout une prise de conscience et une éducation de l'utilisateur qui constitueront les mesures les plus efficaces

7 Références

Références

1. Linux Netfilter, `http://www.netfilter.org/`
2. Winpcap, `http://winpcap.polito.it/`
3. M. Blanc, É. Detoisien, A. Guignard & L. Oudot, Pénétration de réseaux et backdoors furtives, Linux Magazine Hors Série 12, novembre 2002.
4. Éric Detoisien, WhiteCane,`http://valgasu.rstack.org/tools/whitecane.zip`
5. Truff, nfbypass, `http://projet7.tuxfamily.org/factory/releases/nfbypass.c`
6. Nicolas Grégoire, Prise de contrôle via Internet Explorer d'une machine compromise située en réseau inconnu, SSTIC 2003
7. Ivo Ivanov, API hoking revealed, `http://codeguru.earthweb.com/system/apihook.html`
8. Zoltan Csizmadia, Injecting a DLL into another process's address space, `http://codeguru.earthweb.com/dll/LoadDll.shtml`
9. John Peloquin, Remote library loading, `http://codeguru.earthweb.com/dll/Loader.html`
10. Xenophi1e, Thermite, Bugtraq `http://cert.uni-stuttgart.de/archive/bugtraq/2003/02/msg00268.html`
11. Virus BugBear, `http://www.virusbtn.com/resources/viruses/bugbear.xml`
12. Kerio Personal Firewall Remote Authentication Packet Buffer Overflow Vulnerability, `http://www.securityfocus.com/bid/7180`
13. ZoneAlarm Personal Firewall Port 67 Vulnerability, `http://www.securityfocus.com/bid/1137`
14. Kerio Personal Firewall Firewall Filter Bypass Vulnerability, `http://www.securityfocus.com/bid/7436`