1. Validation des performances d'un garde-barrière

Supports de la présentation :
http://www.ossir.org/resist/supports/cr/20050131/TestOptimisationFirewall.pdf
http://www.ossir.org/resist/supports/cr/20050131/PerformancesIptablesV2.pdf

1.1. Principes

S'il est souvent dit qu'un système de contrôle ne laissant rien passer est bien plus sécurisé qu'un système laissant passer des paquets, même après filtrage, c'est un point de vue difficile à appliquer. Les systèmes de sécurité ont donc des contraintes qui vont au-delà de la sécurité qu'ils permettent de mettre en place : ils doivent le faire de manière compatible avec les performances de ce qu'ils protègent.

L'objectif principal d'un système de sécurité est évidemment la sécurité, mais la qualité de service ne doit pas être oubliée. Le choix d'un système de sécurité dépend du nombre d'utilisateurs concernés, du type des connexions à contrôler (et des opérations de contrôle), du type des applications assujetties à ces contrôles, etc. Les applications exploitent de manière différente le réseau, donc présentent un profil d'utilisation (vu du système de sécurité) unique à chaque fois. Contrôler des flux de messagerie n'a pas grand rapport avec le contrôle des flux associés à des connexions interactives. Dans le premier cas, le système verra beaucoup de connexions TCP, chacune relativement brève, avec des volumes de données variables. Dans le second cas, les connexions seront moins nombreuses, nettement plus longues, avec de faibles volumes de données échangées.

Devant la quantité d'outils disponibles, il est intéressant d'évaluer les seuils de rupture des performances avant la mise en production (voire avant le choix), dans des situations proches de la réalité du déploiement. Les fiches de performances des produits correspondent à des utilisations qui peuvent se révéler assez éloignées de l'utilisation prévue (tests avec des pages HTML de 1 Ko par exemple, ce qui est plutôt rare de nos jours).

Mais la mesure des performances d'un système de sécurité ne doit pas non plus être biaisée par les temps de réponse des applications ainsi sollicitées. Autant que possible, il faut s'affranchir de cette facette, afin d'obtenir les performances du système de sécurité « seul ».

Nous sommes habitués aux tests fonctionnels d'un système de sécurité. Il s'agit « simplement » de valider les règles de l'outil par rapport à la politique de sécurité. Les tests de performances sont un peu différents, par exemple :

- Mesurer le nombre de création de connexions TCP par seconde,
- Mesurer le nombre de connexions ouvertes que le système peut maintenir,
- Mesurer le nombre de requêtes de type GET que le système de sécurité peut supporter,
- Mesurer le « remplissage » de la bande passante,
- Mesurer le comportement lors d'attaques.

1.2. Démonstration

La démonstration reposait sur un garde-barrière iptables sous Fedora, avec une machine dont le processeur était cadencé à 2,6 GHz, disposant d'1 Go de mémoire vive. L'architecture, illustrée à la Figure 1, est simple : le garde-barrière est placé en sandwich entre un « émetteur » (Avalanche), qui simule les utilisateurs, et un réflecteur qui simule les serveurs.

Figure 1: Architecture de la démonstration.

- *Nombre de connexions TCP par seconde* : le « client » utilise le protocole HTTP 1.0 (pas de KeepAlive ; une requête client correspond alors à une connexion TCP). Les pages demandées font de 512 à 102400 octets (0,5 à 100 Ko). Résultat : saturation du garde-barrière aux environs de 1000 ouvertures de connexion par seconde.
- *Nombre de connexions TCP ouvertes* : le « client » utilise HTTP 1.1, avec la connexion TCP qui reste ouverte entre les requêtes. Il est configuré pour attendre trente secondes entre deux requêtes, fait 10 transactions HTTP par connexion TCP, et demande des pages de 0,5 à 100 Ko. Résultat : le garde-barrière maintient 60000 connexions ouvertes en parallèle.

Le résultat de ces deux mesures permet de conclure que si le protocole majoritaire est HTTP 1.0, il est préférable de chercher un système de sécurité capable d'absorber de très nombreuses ouvertures de connexions TCP par seconde. Si le protocole majoritaire est HTTP 1.1, il vaut mieux s'orienter vers un système capable de gérer de très nombreuses connexions ouvertes en parallèle.

Les paramètres d'optimisation (pour cette démonstration) sur lesquels il est possible d'agir sont :

- La durée du KeepAlive des connexions TCP,
- Le nombre de sessions ouvertes mémorisées par iptables,
- Le délai d'attente au bout duquel une connexion est considérée comme close.

En agissant sur ces paramètres, les mesures de performances donnent les valeurs suivantes :

- Ouvertures de connexions : entre 4 et 5000 à la seconde.
- Nombre de connexions ouvertes : plus de 100000.

Ces éléments permettent de mesurer tout l'intérêt de faire des tests significatifs, et d'optimiser les outils.

Afin d'obtenir des mesures réalistes (« hors laboratoire »), il est important de tester en parallèle toutes les règles du garde-barrière. La construction d'un scénario de tests approprié est donc un travail non négligeable de la campagne de mesures. Si l'on ne teste que les règles jugées principales, les mesures obtenues restent du domaine du laboratoire, et non pas du domaine de la production.

Un document plus complet sur l'optimisation de la configuration d'Iptables est disponible sur le site de ReSIST, à l' URL :
http://www.ossir.org/resist/supports/cr/20050131/PerformancesIptablesV2.pdf