

## Tout commence par une bonne administration...



### éditorial

Les problèmes de sécurité des systèmes d'information commencent toujours par un défaut d'administration des machines. LINUX ne fait pas exception en la matière, les administrateurs le savent bien. Mais les particuliers, eux, – LINUX se généralise de plus en plus comme système personnel – en ont-ils vraiment conscience? Combien choisissent les paramètres par défaut de l'installation sans se donner la peine de connaître les services dont ils ont réellement besoin? Est-ce simplement un des effets de la fameuse loi du « moindre effort » ou plus prosaïquement de l'ignorance? Qu'importe, puisqu'on aboutit au même résultat: des machines qui deviendront à terme de véritables passoires pour tous les pirates de la Terre. Malgré les méritoires progrès des distributions pour faciliter l'installation du système, il reste – et restera toujours – une question que personne ne peut résoudre à la place de l'intéressé: de quoi a-t-il besoin? L'article de Maurice Libes rappelle quelques bases indispensables à l'administration d'un système LINUX.

Mais une machine bien installée n'est qu'une étape dans une bonne administration. Même si le monde UNIX y est beaucoup moins sensible, les attaques virales restent un des problèmes récurrents de l'administration des réseaux ouverts. Jusqu'à une date récente un bon antivirus installé sur un poste de travail et mis à jour fréquemment offrait une protection qui suffisait dans la plupart des cas. Aujourd'hui, ce n'est plus vrai: la vitesse de propagation des vers Internet, les failles de sécurité récurrentes sur les logiciels de courrier les plus utilisés ou l'insouciance de certains utilisateurs, obligent à compléter l'arsenal défensif. À la sécurisation des postes clients, il faut ajouter la sécurisation des serveurs de fichiers et le filtrage des courriers – plus précisément des pièces jointes et des pourriels (spam en québécois) – sur les serveurs de messagerie. Certes, ces mesures ne peuvent être prises sans précautions préalables – en particulier sur le principe du respect des correspondances privées –, mais elles paraissent inévitables sauf à accepter que la messagerie électronique ne devienne finalement inutilisable.

A. Schwenck  
Fonctionnaire Défense

## La protection contre les vers de messagerie

Généralement, les virus et le spam sont les deux types de flux que l'on cherche à bloquer sur les serveurs de messagerie. Nous vous proposons de faire un tour de la question et de vous présenter la solution développée par l'École des Mines de Paris.

UN virus ou un cheval de Troie peuvent être envoyés par message électronique (comme n'importe quel autre fichier), mais cela ne fera pas d'eux pour autant des vers de messagerie. Ce qui fait la particularité d'un ver de messagerie, c'est justement la présence simultanée d'un code malveillant (le ver), mais également l'utilisation spécifique de la messagerie électronique comme moyen de propagation. Ce dernier aspect est généralement ignoré, et s'il permet à un nouveau ver de faire le tour de la planète en quelques heures, il permet également – s'il est correctement pris en compte – de se protéger contre le même ver (1).

### La protection par filtrage des fichiers attachés douteux

L'idée est simple: tous les vers de messagerie à diffusion rapide circulerait dans des fichiers contenant du code présenté sous forme de pièce attachée immédiatement exécutable, donc facilement reconnaissables par leur extension: exe, com, bat..., mais également vbs, js, shs..., et/ou du code malveillant directement intégré dans le message, balise « iframe » permettant le lancement automatique d'une pièce attachée, script caché, etc.

Les premiers sont les types de fichiers classés « douteux » ou « unsafe files » par Microsoft (2). Il suffirait donc de bloquer tous les messages électroniques contenant des fichiers attachés de ce type pour être protégé; quant aux seconds, il suffit que le client de messagerie leur interdise toute action.

Jusqu'à présent cette idée s'est avérée totalement efficace contre tous les vers de messagerie. Elle est mise en œuvre avec succès par de nombreux internautes grâce à l'utilisation de clients de messagerie très simples – et surtout sans faille reconnue, comme Netscape Messenger 4.7 par exemple –, puis un rejet manuel, volontaire et systématique des « fichiers douteux ». Dans ce cas, et en présence d'un ver de messagerie connu ou inconnu, l'usage d'un anti-virus de messagerie sur le poste de travail est parfaitement inutile pour empêcher une contamination; le message ne pouvant avoir aucune action, il suffit de l'effacer. L'usage du moniteur anti-virus résident est largement suffisant: si l'on tient à identifier un virus ou un ver reçu, il est parfaitement ..... suite page 2 >>>

possible de le copier puis de l'analyser avec le scanner, ou d'analyser directement la boîte de réception si le scanner le permet. Le filtrage des virus de messagerie est à réserver aux anti-virus sur serveur qui apportent un plus grand confort à partir d'un certain nombre de postes.

La seule difficulté est dans la généralisation d'une telle pratique, car les freins sont nombreux - le plus important étant l'ignorance technique de l'utilisateur moyen. Malgré la diffusion des mesures de prévention relatives à la messagerie, les résultats ne sont pas là, les causes étant le manque de méfiance, la mauvaise compréhension de la signification des extensions des fichiers, l'utilisation d'un client de messagerie vulnérable par simple ignorance de l'existence des correctifs ou le mauvais paramétrage du même client de messagerie. L'unique solution viable semble être dans l'automatisation du rejet systématique des «fichiers douteux» ; elle est d'ailleurs de plus en plus pratiquée, mais, faute de publications sur le sujet, il était jusqu'à présent difficile d'en apprécier les réels avantages et inconvénients.

Nous avons donc réalisé une analyse portant sur tous les messages ayant transité par le serveur de messagerie de l'École des Mines de Paris durant le mois de mai 2002. Il en ressort :

Nombre total de messages	420 000
Messages avec des fichiers attachés	105 000
Messages avec des «fichiers douteux» refusés	4 233

Le filtrage des fichiers non sûrs n'affecte donc qu'une infime partie des messages, soit environ 1 %. Les 4 233 messages refusés comportaient 7 761 fichiers attachés dont 1431 exe, 995 pif, 918 scr, 821 bat, 54 com, 45 lnk et 9 js.

Seuls 8 messages sur 4 233 étaient sains, dont 3 seulement comprenant des exe, soit un taux de fausse alerte de 0,2 % pour cette expérience, taux comparable à celui des anti-virus soumis à des nouveaux fichiers, à la différence près que par le filtrage des «fichiers douteux», même les vers de messagerie inconnus auraient été rejetés.

La règle à retenir est donc :

Tous les messages contenant des «fichiers douteux» en pièce attachée, quelle que soit leur apparence, quel que soit l'expéditeur et quel que soit le diagnostic des anti-virus, sont à considérer comme étant «infectés» jusqu'à avis contraire d'un laboratoire anti-viral.

## Mise en pratique

L'usage montre que l'utilisateur accepte très rapidement cette pratique ; seul le filtrage de l'extension exe le surprend une ou deux fois, puis il s'adapte en archivant ses fichiers (zip, rar...); en aucun cas il n'est limité dans sa capacité d'échange des informations. Bien au contraire, il devient de plus en plus demandeur du filtrage des messages non sollicités (spams), messages dont la quantité est bien souvent supérieure à sa propre correspondance.

Pour les petites unités de réseau, la mise en pratique est aisée. Il est impératif d'utiliser un logiciel de messagerie satisfaisant au moins les critères suivants :

- Il ne doit envoyer que des messages sous forme texte, les éventuels codes HTML et images à joindre ne peuvent donc plus l'être que sous forme de pièce attachée.
- Il peut assurer la visualisation des messages en format HTML reçus, mais en aucun cas il ne doit exécuter ses scripts.
- Toutes les balises doivent être interprétées comme des pièces jointes passives (par exemple la balise iframe de Klez est vue par «The Bat» comme une pièce attachée isolée HTML sans intérêt, avec Netscape elle est simplement ignorée).

Le client de messagerie doit bien souvent être corrigé et reconfiguré pour satisfaire les critères ci-dessus, mais il faut également qu'il bloque automatiquement l'exécution des pièces attachées exécutables. Pour le moment le choix est très restreint (The Bat), mais les autres éditeurs vont certainement réagir.

La mise en pratique sur un gros réseau est plus difficile. Bien souvent on souhaite garder Windows/Office/IE/Outlook par soucis d'homogénéité. Il faut également déployer les logiciels sur des milliers de postes, maintenir et reformer les utilisateurs. Le volume des messages à traiter devient également énorme. Comme pour les virus, le filtrage des «fichiers douteux» sur le serveur de messagerie s'impose alors.

## Principe du filtrage anti-viral sur les serveurs de messagerie

Dans un service de messagerie, il y a deux types de composants : le MUA (Mail User Agent) et le MTA (Mail Transport Agent). Le

MUA est la partie destinée aux utilisateurs (Netscape, Eudora, Outlook, Pine...), alors que le MTA est la partie que l'on trouve sur les serveurs et passerelles (sendmail, postfix...).

La fonction principale d'un MTA est le routage. Il reçoit des messages en provenance des MUA locaux ou d'autres MTAs et décode la destination. S'il s'agit d'une destination locale, il fait le nécessaire pour que le message soit transmis localement, sinon il renvoie le message vers un autre MTA plus proche de la destination finale.

Des fonctions annexes s'ajoutent à la fonction routage : ce sont, pour la plupart, des fonctions de sécurité, telles le filtrage anti-spam ou l'authentification par certificats.

Ainsi, si l'on veut ajouter un filtre sur le serveur de messagerie, il y a trois endroits possibles : en entrée, en sortie, ou encore intégré au MTA par une API (interface de programmation).

Les filtres en sortie, utilisés généralement directement sur la boîte à lettres du destinataire, sont les plus simples à mettre en œuvre. Un exemple de solution de ce type est l'utilisation de procmail couplé à un scanner externe. Cette solution permet de personnaliser le filtrage au niveau de l'utilisateur. Son inconvénient est le traitement multiple lorsque le même message est envoyé à plusieurs destinataires locaux.

Le placement en entrée du MTA résout le problème du traitement multiple mais en ajoute d'autres. En effet, les fonctions annexes (anti-spam, authentification...) nécessitent le contact direct entre le MTA et son client, qui n'est plus assuré si l'on insère un filtre avant le MTA. Or, ces fonctions n'existent pas d'habitude dans les filtres anti-virus (et c'est normal!). Pour contourner cette difficulté, on insère le filtre anti-virus entre deux MTAs (installés parfois sur deux machines différentes) : le premier s'occupant des fonctions annexes (anti-spam) et le deuxième de la fonction routage.

C'est cette solution (complète) que l'on trouve le plus souvent chez les éditeurs d'anti-virus pour serveurs de messagerie (e.g. Mail-Monitor de Sophos). La raison est simple : les deux logiciels étant distincts, il n'y a pas de relation de dépendance entre l'éditeur d'anti-virus et le fournisseur du MTA.

La troisième solution consiste à utiliser une API de façon que le filtre devienne partie intégrante du MTA. Ce type de filtre peut alors accéder à - et utiliser - toutes les variables internes du MTA pour effectuer un traitement plus fin des messages. Cette solution est beaucoup plus efficace, mais elle exige une interface différente pour chaque MTA (sendmail, postfix...). Des exemples de ce type de solution sont j-chkmail, amavis-milter - qui n'est pas une solution antivirale, mais une interface permettant de connecter les deux MTAs les plus courants (sendmail et postfix) à pres que n'importe quel scanner antivirus - et la solution proposée par Trendmicro.

..... suite page 3 >

## Références

- [1] <http://www.certa.ssi.gouv.fr/site/CERTA-2000-INF-002/index.html.2.html>
- [2] Unsafe Files : <http://support.microsoft.com/default.aspx?scid=kb;EN-US;q262631>
- [3] j-chkmail home page : <http://j-chkmail.ensmp.fr>
- [4] RFC 2045-9 - MIME (Multipurpose Internet Mail Extensions)
- [5] RFC 2505 - Anti-Spam Recommendations for SMTP MTAs

## Le filtrage des fichiers attachés douteux sur les serveurs de messagerie

Une application pratique du filtrage des fichiers dits « douteux » sur serveurs a été développée par le Centre de Calcul de l'École Nationale Supérieure de Mines de Paris et est opérationnelle depuis janvier 2002. Il s'agit de j-chkmail (3), logiciel libre développé au départ pour les besoins internes. Le filtrage des messages ayant des fichiers attachés a été le but premier de j-chkmail, mais pas le seul. L'idée était aussi de le rendre aussi extensible que possible, de façon à satisfaire les besoins et de pouvoir ajouter facilement des nouveaux critères de filtrage. C'est ainsi que des fonctionnalités de filtrage de spam sont apparues rapidement, le but n'étant pas de remplacer les outils déjà existants et efficaces, mais d'en créer des nouveaux.

### Application anti-virale

j-chkmail a son propre moteur d'analyse dont le principe est complètement différent de celui des anti-virus.

L'insertion de documents dans un message se fait grâce à des balises. Ces balises contiennent des informations permettant la restauration des documents par le destinataire : notamment le nom du fichier et le type de codage. Le format de ces balises est défini par les RFCs (Request For Comments) 2045-9 (4,5). j-chkmail analyse ces balises et extrait les informations permettant d'identifier le nom et le type du fichier (par son extension). Si le type correspond à un des types définis, le message entier est remplacé par un message d'alerte envoyé aussi bien à l'émetteur de message qu'à son destinataire (le message d'origine peut être mis en quarantaine).

Les avantages sont importants : le traitement des messages est beaucoup plus rapide, et tout nouveau virus est détectable sans délai et sans besoin de mise à jour du moteur d'analyse.

Un autre point d'intérêt de j-chkmail est le traitement qu'il fait sur les balises « MIME ». En fait, avec le déploiement des anti-virus sur les serveurs de messagerie, les derniers virus cherchent le moyen de pouvoir passer au travers des filtres. Un moyen est l'insertion de malformations sur le corps du message de façon que le traitement du filtre soit faux et que le scanner de l'anti-virus ne soit pas appelé, ou alors appelé avec des mauvaises informations.

*Prenons deux exemples :*

Certaines versions de GIBE insèrent des caractères nuls juste avant les balises MIME (interdit selon le protocole SMTP) ; un filtre écrit en langage C et utilisant des bibliothèques standard de traitement de chaînes de caractère pour

croire que c'est la fin du message, et le traitement est arrêté sans que le fichier attaché soit transmis au scanner.

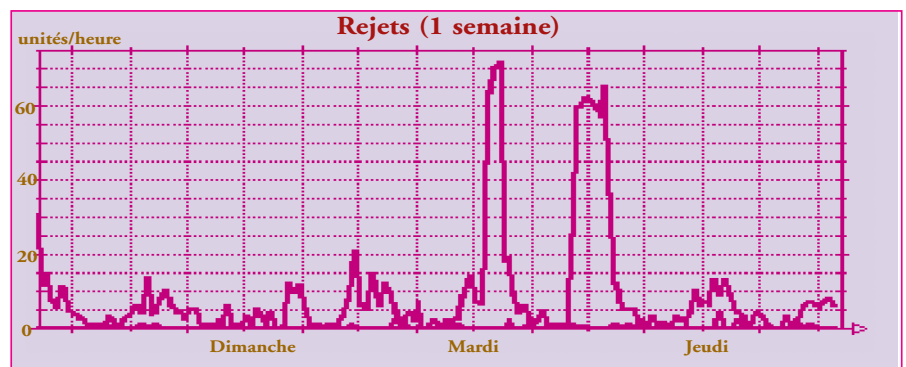
Certaines variantes de Klez insèrent des caractères non valables (selon la norme MIME) dans le nom du fichier, de façon que la fonction de décodage des balises retourne une erreur. Par exemple un nom de fichier avec des espaces, sans qu'il soit mis entre guillemets. C'est ainsi qu'au mois de mai le nombre de virus retenus par jour sur le serveur de messagerie de l'École des Mines est passé de 70 à 250. Cela prouve que le but de Klez a bien été atteint.

Mais j-chkmail, comme toute autre solution de filtrage sur serveur de messagerie, n'est pas la solution ultime. Il faut la compléter avec un anti-virus sur les postes des utilisateurs. L'analyse de l'activité de filtrage sur le serveur (voir figure) montre un phénomène intéressant. Dans ce graphique, à l'exception des deux pics dominants apparaissant le mardi et le mercredi, tous les messages refusés sont des messages entrants du site. Ces deux pics correspondent, en fait, à l'activité d'un ordinateur portable qui avait été infecté ailleurs. Si le filtrage sur le serveur de messagerie protège les postes locaux des agressions venant de l'extérieur par la messagerie, ce n'est pas le cas pour les agressions locales pouvant se propager, par exemple, par le biais du partage de fichiers.

regroupement (groupe de travail) n'ayant aucune raison de recevoir du mail en provenance de l'extérieur.

Le filtrage selon la cadence de connexion est redoutable : la cadence des messages publicitaires, envoyés le plus souvent par des robots, ressemble à des rafales, tandis que celle des messages envoyés par des humains ressemble le plus souvent à un processus poissonien. La détection des rafales se fait par le seuillage du nombre de connexions effectuées par chaque passerelle. Le comptage est fait sur une fenêtre glissante d'une dizaine de minutes. Ce filtrage donne des bons résultats, à condition d'ajouter les serveurs de liste extérieurs auxquels les utilisateurs sont abonnés. Ce critère équivaut à une liste noire dynamique.

On peut aussi obtenir des très bons résultats avec un filtrage basé sur la bonne déclaration DNS de la passerelle de messagerie. Pour relayer du spam, les spammers utilisent, le plus souvent, des machines secondaires d'un domaine et jamais des machines principales, soit parce que ces dernières sont plus protégées, mais aussi pour ne pas se faire remarquer. Assez souvent, ces machines secondaires n'ont pas de déclaration DNS ou alors des déclarations incohérentes. La déclaration DNS est comme une pièce d'identité. Cette vérification équivaut à demander la pièce d'identité



### Application anti-spam

En ce qui concerne le filtrage anti-spam, la plupart des fonctionnalités actuelles de j-chkmail cherchent la vérification de conformité et la construction d'une liste d'accès dynamique adaptative. Les intérêts du contrôle d'accès dynamique et adaptatif sont la construction et l'entretien automatiques de la liste d'accès pendant le fonctionnement du filtre, ainsi que le temps de réaction plus court que celui des listes statiques telles la base accès de sendmail ou les listes noires sur DNS (maps, osiru, ordb...).

La vérification de conformité est faite sur les champs d'en-tête (from, to, sujet...) et l'enveloppe.

Avec j-chkmail, on peut limiter en interne l'envoi de messages à certaines adresses. Ce sont, par exemple, les adresses de service ou de

du facteur qui vient vous remettre un paquet : s'il refuse de la présenter ou s'il présente une pièce qui n'est pas la sienne, le paquet est refusé. Le taux de faux positif est faible mais on ne peut pas le négliger : sur environ 10 000 messages refusés quotidiennement par ce critère, on compte en moyenne un faux positif pour une centaine de passerelles. Ces faux positifs sont facilement identifiables puisqu'ils sont parmi les adresses avec un nombre faible de connexions. Pour résoudre cela, j-chkmail utilise un mécanisme de quota de connexions par jour et une « liste blanche ». Lorsqu'il s'agit d'une passerelle connue, le responsable est contacté et l'accueil se passe généralement très bien.

Pour satisfaire des besoins plus particuliers, j-chkmail possède une interface permettant de connecter des filtres réalisés par l'utilisateur dans son langage préféré ..... suite page 6 ..... ➤

# Utiliser Linux... oui, mais pas les yeux fermés !

Cet article a dû être abrégé pour pouvoir être publié dans ce numéro. Dans sa version complète, il comporte de nombreux exemples indispensables à ceux qui veulent « passer aux travaux pratiques ». Vous pouvez retrouver cette version à l'URL [http://www.cnrs.fr/Infosecu/linux\\_sans\\_peine.rtf](http://www.cnrs.fr/Infosecu/linux_sans_peine.rtf)

## Objectifs

Votre PC Linux est connecté à Internet ? Il est donc soumis à des attaques par des outils automatiques qui vont exploiter les failles qu'ils trouvent. Cet article est destiné à faire prendre conscience de ces risques et à proposer un ensemble de mesures destinées à mettre en place, ou améliorer le niveau de sécurité de base sur un système Linux connecté en réseau.

Nombre d'utilisateurs, issus du monde Windows ou Mac, ont une culture informatique qui ne les prédispose pas toujours à connaître les fonctionnalités et arcanes d'un système d'exploitation multi-utilisateur et multitâches (i.e. plusieurs programmes s'exécutent de façon quasiment simultanée)... En outre la connexion de nos postes à un réseau planétaire comme Internet change la nature du problème par rapport aux années 80. L'ouverture sur un réseau de cette échelle centuple les risques d'actes de malveillance informatique : intrusions, ou refus de services. En effet sur un système multitâche comme Linux, un grand nombre de programmes sont présents et actifs en mémoire. Appelés « serveurs » ou « démons », ils s'exécutent sans trace apparente à l'écran (un serveur de messagerie, un serveur www par exemple). Cependant sur votre machine, ces programmes sont à l'écoute, et en attente de connexions provenant de l'extérieur. Si le poste Linux est peu ou mal administré, trop ouvert sur l'extérieur, ou encore s'il propose des versions de logiciels obsolètes, les facteurs de risques permettant des actes malveillants seront d'autant plus grands.

Linux possède, de base, une grande variété de programmes pouvant assurer la sécurité.

Ces lignes visent à :

- attirer l'attention de certains utilisateurs sur la nécessité d'avoir une administration de la machine comprenant un seuil minimal de sécurité du système et du réseau
- faire connaître et passer en revue quelques éléments pour mettre en place un niveau de sécurité minimum.

## Lors de l'Installation

Certaines distributions Linux (Redhat, Mandrake, Suse, Debian...) facilitent la phase d'installation du système en proposant des classes d'installation pré-définies. Dans ce cadre-là, il s'agit de ne pas installer le système « à l'aveuglette », sans connaître la portée de ce que l'installation automatique de la distribution aura choisi pour vous. Tout dépend de l'utilisation future de la machine que l'on installe :

est-ce un poste bureautique, une station de développement ou bien une machine serveur réseau ?

Dans tous les cas, il sera nécessaire de :

- vérifier et compléter les choix prédéfinis en sélectionnant l'option « choix des paquetages individuellement ». Cela permettra de découvrir et d'apprendre à connaître les noms et fonctions des différents paquetages logiciels afin de décider en connaissance de cause si oui ou non ils sont nécessaires sur votre machine... Au bénéfice du doute, il est sûrement préférable de ne pas installer un service si on ne sait pas à quoi il sert. Il sera en effet très facile de le rajouter par la suite si on en a besoin à l'aide des commandes « rpm » ou « dpkg ».
- choisir les services que l'on souhaite lancer au démarrage... Il est nécessaire de ne pas négliger cette phase et de bien choisir les services qui seront strictement nécessaires. Enfin, à l'issue de l'installation, il sera nécessaire de vérifier ce qui a été effectivement installé.

## Post Installation : vérifier les services lancés au démarrage

L'installation est terminée, félicitations ! Un certain nombre de paquetages sont donc installés sur la partition système Linux « / ».

Il est maintenant nécessaire de passer en revue les services qui sont effectivement lancés en mémoire, afin de ne pas laisser de services inutiles qui pourraient être ultérieurement des cibles potentielles pour d'éventuels pirates sur Internet.

## Interdire le lancement de certaines applications

Que voit un pirate en herbe depuis l'extérieur de votre site ? Quels renseignements peut-il obtenir sur votre machine ? Le logiciel nmap disponible sur Linux est un outil d'exploration qui permet de déterminer quels services sont présents et actifs sur une machine Linux (resp. Unix). Pour un éventuel pirate, il n'y a qu'à découvrir les services qu'une machine héberge, relever le numéro des versions des serveurs présents et tenter d'exploiter d'éventuelles failles (failles qui sont d'ailleurs décrites sur un grand nombre de sites Internet). Ces sites fournissent les vulnérabilités de telle ou telle application, ainsi que les programmes permettant d'exploiter cette vulnérabilité. Il est donc nécessaire de montrer le moins de services ouverts possibles en désactivant ce qui est inutile, et de contrôler les connexions sur nos machines serveurs.

## Désactiver les services inutiles

Le lancement des services réseau d'un système Linux repose sur un système dit « client-serveur ». Le schéma est simple : une machine « cliente » initie une connexion vers un « serveur » en demandant le lancement d'une application particulière. Ces services (encore appelés dans le jargon « démons », ou « serveurs ») sont des programmes qui sont lancés de deux manières différentes :

- soit par le programme inetd : inetd est un programme activé dès le démarrage du système, et qui attend en mémoire des demandes de connexion pour un service donné. Quand un client extérieur (ou même local à la machine) demande une connexion par le réseau, inetd détermine quel est le service à lancer en consultant son fichier de configuration/etc/inetd.conf.
- soit au démarrage de la machine : sans passer par inetd. C'est le cas de certains services très sollicités (comme un serveur de mail, ou un serveur www) qui doivent être présents en mémoire et répondre rapidement et directement aux demandes de connexion sans passer par le « super serveur » inetd.

**Première étape de sécurité :** vérifier le contenu de /etc/inetd.conf. Il ne faut laisser dans ce fichier que les services que l'on désire explicitement activer, et aucun autre.

**Deuxième étape de sécurité :** ce schéma client-serveur basique est trop simple, et pas assez sécurisé ! En effet, par défaut, on ne contrôle pas « qui » (i.e. quelle machine cliente) est à l'origine de la demande. Or les services qu'une machine propose n'ont pas forcément à être disponibles à tout l'Internet ! Il est nécessaire de restreindre l'utilisation des services réseau vers un ensemble de machines à qui on souhaite réserver ce service. Sauf cas particuliers de serveurs publics (comme un serveur WWW, ou de messagerie), les services réseau ne devraient pas être destinés à être accessibles depuis tout l'Internet. Aussi il faut sécuriser ce schéma de base simpliste de deux manières :

- soit en utilisant tcpwrapper
- soit en utilisant xinetd

## Contrôler l'accès aux services par tcpwrapper ou xinetd

Dans le contexte de sécurité actuel, l'utilisation de tcpwrapper ou xinetd sur Linux devient indispensable ! L'idée de base de tcpwrapper (ou de xinetd comme successeur de inetd) est d'exercer un contrôle sur l'identité de la ..... suite page 5 >>>

..... suite de la page 4

machine cliente qui veut lancer un service réseau du serveur. tcpwrapper permet donc de limiter les accès aux services réseau offerts par une machine serveur en fonction de l'adresse réseau des machines clientes.

**Utilisation de tcpwrapper :** tcpwrapper est un petit programme appelé «tcpd» qui est lancé par inetd avant le service demandé par la machine cliente. Pour activer tcpwrapper (i.e tcpd) il suffit de le rajouter dans le fichier /etc/inetd.conf. Ce système est intégré basiquement dans les distributions Linux depuis longtemps (rpm -q tcp\_wrappers pour vérifier sa présence).

Dans ce schéma, inetd lance tcpwrapper avant le service convoité, lequel tcpd va alors engager une série de vérifications pour savoir si le service demandé est permis ou non pour la machine cliente extérieure. Les contrôles de tcpwrapper se font dans deux fichiers :

- /etc/hosts.deny : indique la liste des services qui sont interdits pour certaines machines et domaines internet ;

- /etc/hosts.allow : indique ceux qui sont autorisés. Dans quel ordre se font les contrôles entre hosts.allow et hosts.deny ? L'accès à un service est autorisé si tcpwrapper trouve le couple <service:réseau> correspondant dans le fichier hosts.allow ; sinon le fichier hosts.deny est consulté et l'accès est refusé si cette paire y est trouvée. Si le couple <service:réseau> n'est dans aucun de ces deux fichiers, l'accès au service est accepté par défaut.

La politique la plus efficace dans ce fichier hosts.deny est de commencer par tout interdire en écrivant <All:All>. Ce qui signifie que tous les services de ma machine sont interdits à tout le monde. Cela nous permet de mettre en place une politique de sécurité visant à «Interdire tout ce qui n'est pas explicitement autorisé». Il nous faut alors définir ce qui est explicitement autorisé dans /etc/hosts.allow

**Utilisation de xinetd :** xinetd est le programme qui est amené à remplacer inetd. xinetd intègre de façon native le supplément de sécurité de tcp\_wrappers et apporte quelques éléments de contrôle supplémentaires.

On trouvera un très bon article en français sur les apports et nouveautés de xinetd sur <http://www.fr.linuxfocus.org/Francais/November2000/article175.shtml>

## Le problème des mots de passe

**Complexité.** On ne le dira jamais assez : les mots de passe doivent être suffisamment complexes pour ne pas être découverts par des individus soit dotés de bon sens (pour trouver toto, azerty, vélo), soit dotés du logiciel «john-the-ripper» qui permet de décrypter les mots de passe dans un fichier de mots de passe. En bref, un bon mot de passe doit comporter au minimum six caractères, être une combinaison de minuscules, majuscules, chiffres, et caractères de

punctuation. Le mot de passe ne doit pas être un mot du dictionnaire ! Le plus simple est de trouver une combinaison d'une phrase qui ait un sens mnémotechnique. Exemple : «J'ai deux amours, mon pays et Marseille» pourrait donner G2amPeM.

**Chiffrage.** Il est bon de le savoir... ou de le rappeler : lorsque vous tapez quelque chose sur votre clavier, lors de la connexion à un service distant (un serveur pop3 par exemple, lorsque vous relevez vos mails...), tout ce que vous tapez, toutes les informations du dialogue client-serveur (et donc votre mot de passe) circulent par défaut de façon lisible (on dit «en clair» dans le jargon) sur le parcours entre le client et le serveur. Il suffit à un individu mal intentionné de lancer un programme d'écoute (un «sniffer» dans le jargon) sur un PC connecté au réseau pour voir circuler vos noms et mots de passe de façon lisible: imaginez la facilité de piratage par la suite !

Deux solutions principales s'offrent à nous pour contrecarrer ses possibilités d'écoute sur le réseau. Ces solutions sont progressivement mises en place par les administrateurs systèmes et réseaux de nos campus et laboratoires :

- avoir une architecture réseau la plus segmentée possible (en utilisant des commutateurs ethernet, ou des réseaux virtuels «vlan», par exemple) ;

- recourir désormais à des applications clients-serveurs qui utilisent la cryptologie. Ces applications mettent en œuvre des algorithmes de chiffrement (comme RSA par exemple), fournissant un système à deux clés (une dite «privée» et l'autre «publique») qui permettent de chiffrer les données sur le réseau. Ce chiffrement rend les données difficilement déchiffrables par un tiers qui n'aurait pas la clé correspondante.

## Principales applications utilisant la cryptologie

Par défaut, les échanges qui se font sur l'internet entre un programme client et une application serveur se font «en clair», laissant ainsi toute possibilité à un individu malveillant d'intercepter et lire ce qui est échangé lors d'une connexion avec un serveur distant (www ou mail ou ftp...). Ainsi lorsque vous tapez votre mot de passe sur votre poste, en vue de relever votre courrier électronique, ou bien d'accéder à un site bancaire via le Web, une possibilité existe que votre mot de passe soit lu par une tierce partie. On imagine alors les conséquences néfastes sur la sécurité des systèmes informatiques, et sur votre vie privée. Cette éventualité est désormais inacceptable, et il convient de la contrôler en utilisant des programmes utilisant du chiffrement.

Il est désormais nécessaire de privilégier et d'utiliser toute application permettant un chiffrement du dialogue client-serveur. Dans ce système, en bref, les parties «cliente» et «serveur» chiffrent les données (resp. déchiffrent) par un système de clé privée/clé publique. Pour l'utilisateur final, ce procédé ne nécessite aucune action supplémentaire ou complexe. Il est juste nécessaire d'utiliser des applications clientes susceptibles d'engager une transac-

tion chiffrée. La majeure partie du travail se fait du côté du serveur (et donc pour l'administrateur du serveur).

Ce mode de transmission s'appelle SSL (Secure Socket Layer), normalisé en TLS (Transport Layer System). Que peut-on sécuriser avec les applications utilisant SSL/TLS ?

**Le relevé de son courrier électronique.** Seuls les programmes de mail suivants proposent un dialogue sécurisé par SSL/TLS. Il s'agit de Outlook Express ou TheBat sur Windows, Netscape messenger (toute plate-forme), Kmail ou Mutt (sur Linux). En ce qui concerne Eudora, seule les versions supérieures à 5.1 proposent ce chiffrement. Pour utiliser ce chiffrement sur vos programmes, il vous suffit de chercher, dans les options de configuration du programme, la case indiquant «utiliser une connexion sûre SSL». Bien entendu il faudra que du côté du serveur de messagerie, l'administrateur du système ait installé les serveurs d'accès aux boîtes de mail permettant le chiffrement (POPS et/ou IMAPS).

**L'accès à un site web.** Pour sécuriser les échanges avec un serveur web distant, il suffit juste pour le client d'utiliser la syntaxe https en lieu et place de http lors de la connexion à un site web. Ainsi <https://www.camif.fr/> permettra d'accéder au serveur WWW d'un site en mode sécurisé, les échanges étant chiffrés par le serveur. Du côté du serveur, il est bien entendu nécessaire que l'administrateur du site distant ait mis en place un serveur WWW sécurisé supportant SSL, sans quoi la connexion sera refusée.

**La connexion sur une machine distante.** Il s'agit de se connecter et de travailler sur une machine B, alors que l'on se trouve devant une autre machine A distante. On peut donc utiliser au maximum les potentialités d'un réseau de machines. Depuis longtemps Linux (et tous les Unix) permettent de travailler en réseau en utilisant les processeurs et disques de machines différentes. Il y a quelque temps, les programmes de base s'appelaient telnet et rlogin. Ces programmes souffrent d'un certain nombre de faiblesses en partie dues au fait que les échanges encore une fois entre client et serveur sont potentiellement lisibles avec un analyseur de réseau. Il est désormais indispensable de reléguer aux oubliettes l'utilisation de ces deux anciens programmes de connexion. D'autant qu'un substitut parfait est disponible ! SSH pour Secure Shell (ou SSF pour sa version française utilisant des clés inférieures à 128 bits).

**L'échange de fichiers par ftp.** Là encore, le dialogue client-serveur ftp peut être chiffré par «sftp» grâce aux serveurs SSHv2. Certaines applications clientes ([www.ssh.org](http://www.ssh.org)) offrent des accès et transferts très ergonomiques.

## Contrôler l'intégrité de son système (tripwire, rpm -V)

En cas d'intrusion sur votre système Linux, il est fréquent que les pirates installent ce que l'on appelle un «rootkit» (disponibles sur [suite page 6](#))

suite de la page 5

Internet). Un «rootkit» (boîte à outils d'un faux administrateur malveillant) est un ensemble de logiciels qui vont permettre à un intrus de s'introduire sur un système cible et de le modifier en laissant le moins de traces possibles. Le but pour l'intrus étant d'agir en passant inaperçu du vrai administrateur. Par exemple, la commande «ls» sera modifiée pour ne pas laisser apparaître certains fichiers du rootkit lui-même. La commande «ps» cachera certains processus, la commande «netstat» n'affichera pas certaines connexions ouvertes, etc. Le «cracker» peut alors facilement installer un accès caché au système (encore appelé «backdoor»), sans que l'administrateur puisse le déceler. Dans ce contexte il est très difficile de déceler une intrusion par les moyens de base. Il est donc indispensable d'utiliser des systèmes de contrôle d'intégrité qui permettent de savoir quels fichiers du système ont pu être modifiés.

Le paquetage «tripwire» (<http://www.tripwire.org>) d'utilisation assez simple, permet de vérifier l'intégrité d'un système Linux et de contrôler toute modification du système par rapport à un état de base sain. Tous les fichiers rajoutés, détruits, ou modifiés seront décelés. tripwire est donc un programme de détection d'intrusion dans la mesure où il permet de vérifier très rapidement l'état d'un système après intrusion, et de repérer si un intrus a modifié le système. «tripwire» peut être lancé régulièrement par l'ordonnanceur de tâches (cron) de Linux. Un rapport est envoyé par mail.

Une autre manière permettant de vérifier l'intégrité d'un ensemble de fichiers est de s'appuyer sur les gestionnaires de paquetages «rpm». L'option -V de la commande rpm indique si le paquetage installé a été altéré.

## Mettre à jour rapidement une application avec Linux ?

En cas de réception d'un avis de sécurité (l'administrateur système d'un laboratoire CNRS devrait logiquement en être informé par le biais des CERT-Renater ou CERT-A) informant d'une faille critique sur un logiciel, il est alors nécessaire de faire rapidement une mise à jour du paquetage incriminé. Pour cela, il faut se procurer la version corrigée sur certains sites officiels de l'Internet ([debian.org](http://debian.org), [mandrake.com](http://mandrake.com), [rpmfind.net](http://rpmfind.net), [redhat.com](http://redhat.com), [isc.org](http://isc.org)...) et

## Quelques références

<http://www.urec.fr/securite/>  
<http://www.cru.fr/securite/>  
<http://www.linuxsecurity.com>  
<http://www.xinetd.org>  
<http://www.fr.linuxfocus.org/Francais/November2000/article175.shtml>  
<http://www.tripwire.org>  
<http://perso.univ-rennes1.fr/bernard.perrot/SSF/>  
<http://www.openssh.org>  
<http://www.ssh.com>

l'installer. Cela dépend de la distribution que vous utilisez :

- par exemple, DrakConf (sur une distribution Mandrake), choix «mise à jour des logiciels», ou plus directement MandrakeUpdate automatise totalement la récupération et l'installation d'une version corrigée d'un logiciel ;
- de manière moins directe, mais tout aussi efficace, le site <http://www.rpmfind.net> permet de chercher et de récupérer la dernière version d'un programme corrigé. Une fois le paquetage récupéré, il suffit de lancer la mise à jour sur votre système, par la commande rpm.

## Sécurité Linux avancée : filtrage au niveau du transport réseau (ipchains)

Enfin, dernier point qui mérite d'être signalé, le système Linux a depuis longtemps la possibilité de faire du filtrage de trames réseau TCP/IP. Cela signifie que Linux a la capacité d'accepter ou de refuser de traiter des trames réseau qui lui proviennent en fonction de plusieurs critères : protocole réseau utilisé, adresse source, adresse destination et port applicatif qui est demandé. Ce système de filtrage au niveau du noyau Linux s'appelle «ipchains» pour la série des noyaux Linux 2.2., ou «iptables» pour la série des noyaux 2.4.

On peut donc faire du filtrage d'accès au niveau des demandes de connexion réseau provenant de l'extérieur. Cela est un peu plus général et performant que le filtrage au niveau des applications que nous avons citées plus haut (par tcpwrapper et xinetd). En effet, à ce niveau on peut intervenir directement au niveau des couches de transport réseau et des protocoles comme TCP, UDP, ICMP mais également au niveau de n'importe quelle application. On peut en outre différencier des connexions entrantes (input), sortantes (output), ou transmises (forward) dans le cas où le PC Linux possède deux interfaces réseau et agit comme un routeur.

Ce système permet également de faire la différence entre des demandes de connexions issues de l'intérieur du site et celles issues de l'extérieur (connexions dites «established»). Ce système est un peu plus complexe à mettre en œuvre et demande une bonne connaissance du protocole TCP/IP et des ports applicatifs... mais il est très efficace. Par défaut, le noyau Linux accepte tout en entrée, sortie, et redirection.

## En conclusion

Ces quelques conseils vous permettront de rester connecter au réseau mondial... avec une bonne sécurité de base, et de tirer tous les bénéfices et avantages de Linux et des logiciels libres sans trop de dommages...

**Maurice Libes**

Centre océanologique de Marseille  
[Maurice.Libes@com.univ-mrs.fr](mailto:Maurice.Libes@com.univ-mrs.fr)

suite de la page 3

(Perl, Java, C...), ou alors des scanners antivirus bien diffusés dans la communauté de la recherche.

## Conclusion

Le serveur de messagerie est un point de passage central dans un système de messagerie. Il est donc naturel qu'il soit un point privilégié pour le filtrage des messages indésirables : les virus et le spam.

L'approche de filtrage anti-virale de j-chkmail est très intéressante par son efficacité, par l'absence de mises à jour fréquentes et par les faibles ressources consommées sur le serveur de messagerie, détail considérable si le serveur traite un trafic important. Mais si elle permet de dégrossir très fortement le trafic sur le serveur, la protection anti-virale n'est pas complète. La contamination peut se faire par d'autres moyens, et il est toujours indispensable de compléter la protection sur le serveur par un anti-virus à jour aussi bien sur les postes clients, mais aussi sur les serveurs de fichiers et, plus généralement, sur tout point permettant le partage de l'information.

L'élimination complète du spam n'est pas réalisable par un outil unique. La méthode la plus efficace est l'analyse de contenu, qui peut poser un problème déontologique en plus de la forte consommation de ressources sur le serveur. j-chkmail apporte quelques fonctionnalités qui, intégrées à d'autres outils - tels les listes noires sur DNS - permettent de réduire considérablement le niveau de spam arrivant sur la boîte à lettres de l'utilisateur final.

**Jose Marcio Martins Da Cruz**

[Jose-Marcio.Martins@ensmp.fr](mailto:Jose-Marcio.Martins@ensmp.fr)  
 Ecole Nationale Supérieure  
 des Mines de Paris  
 Centre de Calcul

**Roland Garcia**

[roland-garcia@wanadoo.fr](mailto:roland-garcia@wanadoo.fr)  
 Microtec Toulouse

## SÉCURITÉ INFORMATIQUE

numéro 41 octobre 2002

SÉCURITÉ DES SYSTÈMES D'INFORMATION

**Sujets traités :** tout ce qui concerne la sécurité informatique. Gratuit.  
**Périodicité :** 5 numéros par an.  
**Lectorat :** toutes les formations CNRS.

**Responsable de la publication :**

ROBERT LONGEON  
 Centre national de la recherche scientifique  
 Service du Fonctionnaire de Défense  
 c/o IDRIS - BP 167, 91403 Orsay Cedex  
 Tél. 01 69 35 84 87  
 Courriel : [robert.longeon@cnrs-dir.fr](mailto:robert.longeon@cnrs-dir.fr)  
<http://www.cnrs.fr/infosec>

ISSN 1257-8819

Commission paritaire n° 3105 ADEP  
 La reproduction totale ou partielle  
 des articles est autorisée sous réserve  
 de mention d'origine