100% SÉCURITÉ INFORMATIQUE

France Metro : 7,45 Eur - BEL, LUX, PORT : 8,5 Eur - CAN : 13 $C - MAR : 75 DH

Mars - Avril 2003

MISC

6

MULTI-SYSTEM & INTERNET SECURITY COOKBOOK

INSÉCURITÉ DU WIRELESS ?

Fonctionnement - Attaques - Sécurité
Les différentes normes

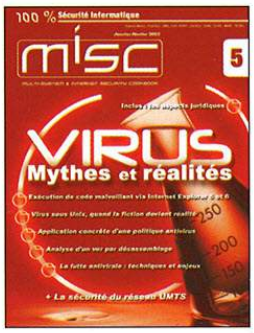
Cyber-terrorisme : fiction ou menace réelle ?

Protégez votre infrastructure réseau IP : IPv6, IP anycast

Récupérez votre code PIN ou une clé RSA avec un chronomètre

L 19018 - 6 - F: 7,45 € - RD

Diamond Editions vous présente ses 4 magazines phares sur le monde Unix



MISC - Le magazine de la sécurité informatique multi-plateforme

MISC est un magazine consacré à la sécurité informatique sous tous ses aspects : de la programmation des logiciels au durcissement des systèmes, car la sécurité informatique repose sur l'interconnexion de nombreux domaines. Et le développement quasi-exponentiel des systèmes d'information met la question de la sécurité des données, sous quelle forme que ce soit, au centre des préoccupations des administrateurs systèmes et/ou réseau, et des webmestres. C'est pourquoi MISC vise un large public professionnel, mais aussi toute personne souhaitant élargir ses connaissances en se tenant informé des dernières techniques et outils utilisés afin de mettre en place une défense adéquate. MISC propose des articles complets et pédagogiques afin d'appréhender au mieux les risques liés au piratage et les solutions pour y remédier, présentant pour cela les techniques offensives autant que défensives, leurs avantages et leurs limites, des aspects indissociables pour considérer tous les enjeux de la sécurité informatique.

7,45 euros Prix au no

WEB → www.miscmag.com

GNU/Linux Magazine France - Le magazine leader de la presse Linux en France

Visant un large lectorat de programmeurs, d'administrateurs systèmes/réseau, d'administrateurs de bases de données (DBA), de graphistes, de webmestres, mais aussi d'utilisateurs particuliers, Linux Magazine diffuse depuis plus de quatre ans un contenu rédactionnel technique permettant au lecteur d'acquérir une expertise dans de nombreux domaines. Ce contenu, "non-périssable", est en grande majorité rédigé par des auteurs en prise directe sur les problématiques liées au monde professionnel, leur expertise permettant de couvrir des cas concrets. Grâce à l'acquisition de connaissances détaillées, les lecteurs sont à même d'expérimenter des solutions qui pourront, par la suite, être mises en production au sein de leur entreprise.

5,95* euros Prix au no

* (avec CD offert comprenant utilitaires/codes sources en relation avec le numéro : exemples de programmes, bibliothèques, programmes, distributions, etc.)

WEB → www.linuxmag-france.org

Précision Mac - Découvrez les compétences Unix de votre Mac

La firme à la pomme est en train, une fois de plus, de révolutionner le monde de l'informatique personnelle et professionnelle en mariant l'ergonomie universellement reconnue de son interface avec la puissance et la stabilité d'un système Unix. En effet, Mac OS X reprend maintenant - en plus de ses avantages propres - toutes les puissantes fonctionnalités, la stabilité, ainsi que les possibilités de personnalisation inégalées d'un système Unix. Précision Mac s'adresse à tous les utilisateurs de Mac OS X, quelles que soient leurs compétences, avec des articles pratiques et pédagogiques afin d'appréhender au mieux l'utilisation de cet operating system dans le cadre professionnel autant que personnel. Ceci couvre non seulement des aspects applicatifs comme la manipulation de la ligne de commande, les recherches, ou encore l'édition de fichiers, mais également le réseau, la sécurité, l'automatisation des tâches ou la découverte de principes fondamentaux en informatique Unix. Précision Mac vise donc un large lectorat souhaitant élargir ses compétences en s'ouvrant à l'univers Unix.

4,95 euros Prix au no

Forum Précision Mac pour interagir avec la rédaction :
http://news.precision-mac.com/

WEB → www.precision-mac.com

Linux Pratique - Apprivoisez votre pingouin !

Linux Pratique se veut un magazine de soutien dans votre découverte du monde GNU/Linux. Vous avez récemment adopté un pingouin et vous souhaitez apprendre à l'utiliser ? Vous rencontrez des difficultés dans le dressage de votre pingouin ? Ce magazine, au moyen de cas concrets, vous propose de maîtriser par la pratique tous les aspects de l'utilisation personnelle de GNU/Linux. L'accent est mis sur un exposé clair et complet de chaque thème abordé, pour un usage au quotidien de son système et ce par tout type d'utilisateur, même débutant.

5,95* euros Prix au no

*(avec CD offert comprenant utilitaires/codes sources en relation avec le numéro : exemples de programmes, bibliothèques, programmes, distributions, etc.)

WEB → www.linux-pratique.com

Pour découvrir nos offres d'abonnement, voir pages 75 et 76

Consultez le site **www.ed-diamond.com** :

- Commande d'anciens numéros
- Formulaires d'abonnements (bientôt disponibles)
- Nouveautés en kiosque

Diamond Editions
6, rue de la Scheer
67600 Sélestat

Renseignements : cial@ed-diamond.com (ou 03 88 58 02 08)

DIAMOND éditions

EDITO

Danc cet édito, je pourrais aborder un sujet sensible, la volonté sécuritaire (NDLA : remarquez l'emploi neutre du terme "volonté", plutôt que "délire" ou "nécessité", qui sont partisans) transparaît de plus en plus, aussi bien au niveau des états que d'instances supérieures (le Parlement Européen qui approuve la conservation des données de connexion par exemple). Je n'aborderai pas ce qui se passe outre-Atlantique, avec la DMCA et sa petite sœur, la SSSCA : nous avons ce qu'il nous faut entre la LSQ (Loi sur la Sécurité Quotidienne) et la LSI (Loi pour la Sécurité Intérieure). Et ça se passe de la même façon dans d'autres états. Et comme si toute cette armada juridique ne suffisait pas, arrive maintenant le projet de loi pour la confiance dans l'économie numérique [1]. N'étant pas juriste, je ne rentrerai pas dans les détails de la loi et vous recommande plutôt de vous faire votre propre idée.

Je pourrais aborder le thème du dossier de ce numéro, les réseaux sans fil (nous laissons volontairement en suspens l'accord de "fil" qui, selon certaines sources autorisées prend un "s" et pour d'autres tout autant autorisées, non). Les réseaux sans fil se déploient de plus en plus. Les intérêts de ces technologies sont nombreux, comme un prix raisonnable ou une simplicité de déploiement. Mais les inconvénients ne sont pas non plus à négliger. Les concepteurs de cette norme, ceux du WEP (*Wired Equivalent Privacy*) en particulier, nous fournissent un excellent exemple de ce qu'il ne faut pas faire. Ils ont mis au point le WEP dans leur coin, sans en référer à des spécialistes. Depuis, des attaques toujours plus efficaces voient le jour. Certaines d'entre elles, liées à la cryptographie, sont connues depuis des années, bien avant que le WEP ne soit élaboré. Du coup, pour les contrer, les fabricants incluent des protections directement sur les cartes. Mais est-ce que cela n'aurait pas été plus simple, et moins coûteux, d'en tenir compte dès le début ?

Mais non, je sais finalement ce que je vais aborder. En fait, je vais faire de la publicité pour le Symposium sur la Sécurité des Technologies de l'Information et des Communications (SSTIC) [2]. Le SSTIC est une conférence francophone, organisée à Rennes en juin 2003, sur le thème de la sécurité de l'information, ce qui comprend à la fois les vecteurs d'information (comme les systèmes informatiques ou les réseaux) et l'information elle-même (cryptographie ou guerre de l'information). Outre les aspects techniques et scientifiques, l'organisation, les politiques et normes de sécurité, les télécoms (PABX, UMTS, etc.), les aspects juridiques (LSI, la DMCA européenne, etc.) ou encore les coûts sont autant de thèmes nécessaires à une complète compréhension des problèmes liés à la sécurité de l'information que nous espérons traiter lors du SSTIC. Les soumissions pour les conférences sont ouvertes jusqu'à la fin mars. Bref, je ne peux que vous inviter à visiter le site [2] pour obtenir tous les renseignements.

Sur ce, bonne lecture,

Frédéric Raynal

---

[1] http://www.telecom.gouv.fr/internet/projet_len.html

[2] http://www.sstic.org

SOMMAIRE

Ont rédigé ce numéro et y ont collaboré :

CHAMP LIBRE

CYBER-TERRORISME :
FICTION OU MENACE RÉELLE ; p 6 à 11

CRYPTOGRAPHIE

GÉNÉRATION D'ALÉA EN CRYPTOGRAPHIE; p 12 à 15

VIRUS

UN VIRUS DE BOOT FURTIF :
STEALTH ; p 16 à 19

PROGRAMMATION

COMMUNICATION
USERLAND/KERNELLAND
OU LA NAISSANCE DE KERNSH ;
p 50 à 56

SYSTEME

SÉCURISER FreeBSD 4.7 ;
p 58 à 65

FICHES TECHNIQUES

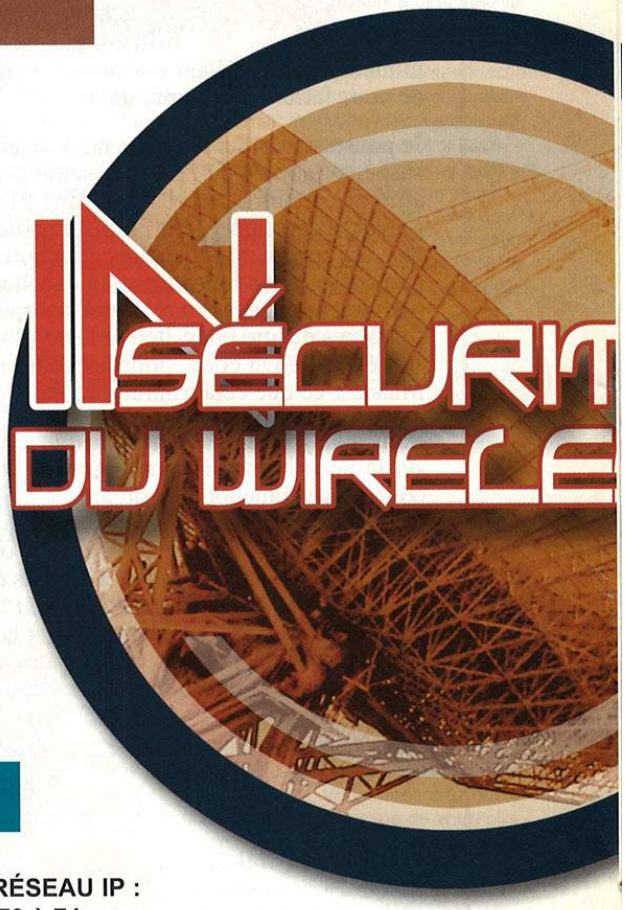
RÉSEAUX SANS FIL : MENACES,
ENJEUX ET PARADES ; p 66 à 69

RESEAU

PROTÉGEZ
VOTRE INFRASTRUCTURE RÉSEAU IP :
IPv6 ET IP anycast ; p 70 à 74

SCIENCE

RÉCUPEREZ VOTRE CODE PIN OU UNE CLÉ RSA
AVEC UN CHRONOMETRE ; p 77 à 81



Denis Bodor
Frédéric Raynal
Patrick Chambet
Pierre Loidreau
Eric Filiol
Nicolas Ruff
Laurent Dupuy
Daniel Polombo
Cédric Blancher
Eric Hazane
Nicolas Brito
Samuel Dralet
Georges Tarbouriech
Pascal Lointier
Nicolas Fischbach
Georges Bart
Nicolas Ritter
Laurent Schaffner

misc
MULTI-SYSTEM & INTERNET SECURITY COOKBOOK

DOSSIER

(IN)SÉCURITÉ DU WIRELESS

■ **PRÉSENTATION DU WiFi (Wireless Fidelity) ; p 20 à 24**

■ **PRINCIPES DE LA NORME 802.11; p 25 à 28**

■ **ATTAQUES SUR LES RÉSEAUX 802.11b ; p 28 à 35**

■ **LA SÉCURITE DU WEP ; p 36 à 42**

■ **BLUETOOTH, HIPERLAN : LES AUTRES NORMES WIRELESS ; p 43 à 49**

Bulletin d'abonnement ; p 75
Commandez nos anciens numéros ! p 76

est édité par Diamond Editions
B.P. 121 - 67603 Sélestat Cedex
Tél. : 03 88 58 02 08
Fax : 03 88 58 02 09
E-mail : lecteurs@miscmag.com
Service commercial : abo@miscmag.com
Site : www.miscmag.com

Directeur de publication : Arnaud Metzler
Rédaction
Rédacteur en chef : Frédéric Raynal
Rédacteur en chef adjoint :
Denis Bodor
Conception graphique : Katia Paquet
Impression : Leonce Deprez - Ruitz
Secrétaire de rédaction : Carole Durocher
Responsable publicité :
Véronique Wilhelm
Tél. : 03 88 58 02 08
Distribution :
(uniquement pour les dépositaires de presse)
MLP Réassort :
Plate-forme de Saint-Barthélemy-d'Anjou.
Tél. : 02 41 27 53 12
Plate-forme de Saint-Quentin-Fallavier.
Tél. : 04 74 82 63 04
Service des ventes : Distri-médias :
Tél. : 05 61 72 76 24

Service abonnement :
Tél. : 03 88 58 02 08

Dépôt légal : 2 e Trimestre 2001
N° ISSN : en cours
Commission Paritaire : 02 04 K 81 190
Périodicité : Bimestrielle
Prix de vente : 7,45 euros

Printed in France
Imprimé en France

Cyber-terrorisme :

Depuis le 11 septembre 2001, les pays largement informatisés ont commencé à prendre sérieusement en compte les risques de cyber-terrorisme contre leurs entreprises et leur société en général. Mais il ne faut pas oublier que le cyber-terrorisme, même s'il semble actuellement entrer dans une nouvelle phase d'expansion, n'est pas un phénomène nouveau. Avec une culture de la connectivité ancrée de plus en plus profondément dans les sociétés dites "modernes", il est promis à un bel avenir. Aujourd'hui, on ne saurait plus vivre sans certains services dont l'épine dorsale est constituée par des réseaux informatiques qui pourraient être réduits à néant par quelques attaques bien réelles, judicieusement menées dans le monde virtuel.

Nous allons définir dans cet article les notions de cyber-terrorisme et de cyber-terroristes, puis envisager différents scénarios possibles, examiner les armes dont disposent les cyber-terroristes, et enfin aborder les mesures à prendre pour construire une défense efficace, car la menace de cyber-terrorisme doit maintenant être intégrée dans toute étude de sécurité.

QU'EST-CE QUE LE CYBER-TERRORISME ?

Le cyber-terrorisme est la convergence entre le terrorisme traditionnel et les réseaux, à commencer par Internet. On peut donc définir le cyber-terrorisme comme l'action délibérée de destruction, dégradation ou modification de données, de flux d'informations ou de systèmes informatiques vitaux d'Etats ou d'entreprises cruciales au bon fonctionnement d'un pays, dans un but de dommages et/ou de retentissement maximum, pour des raisons politiques, religieuses ou idéologiques. Ces dommages peuvent être économiques, sociaux, environnementaux, et même vitaux pour les individus dans certains cas.

Il faut absolument distinguer le cyber-terrorisme du simple cyber-crime, qui consiste à détourner l'usage d'un système dans un but simplement crapuleux. De même, le cyber-terrorisme ne doit pas être amalgamé avec le "hacktivism", qui est certes motivé lui aussi par des éléments idéologiques, mais qui cherche surtout à réveiller la société et à l'éduquer sur certains sujets, pas forcément à la détruire. Enfin, le cyber-terrorisme se distingue du cyber-combat par le caractère généralement civil de ses cibles.

Pourquoi le cyber-terrorisme est-il destiné à avoir autant de succès ? Pour plusieurs raisons. Tout d'abord, le coût d'accès est très faible : un ordinateur portable est beaucoup moins cher qu'un explosif brisant ou qu'une arme de guerre. Ensuite, nos sociétés devenant de plus en plus dépendantes des réseaux d'information, la disparition de ceux-ci peut provoquer des effets économiques, logistiques et émotionnels considérables (voir plus loin). De plus, le public et les journalistes sont fascinés par tous les types d'attaques informatiques, ce qui conduit à une large couverture dans les médias. Enfin, la paralysie des pays dits "développés" lorsqu'ils sont privés de réseaux peut faire la part belle aux pays moins équipés et moins vulnérables de ce côté.

QUI SONT LES CYBER-TERRORISTES ?

On distingue en général 3 types de cyber-terroristes.

Les cyber-terroristes sont en général des sous-groupes de groupes terroristes traditionnels. Ces sous-groupes peuvent être non structurés et constitués d'individus peu nombreux, travaillant sans organisation particulière, avec peu de moyens, de préparation, de compétences et de stratégie, ou bien au contraire être parfaitement organisés, avec des moyens conséquents et une définition précise de leurs cibles et de leur tactique.

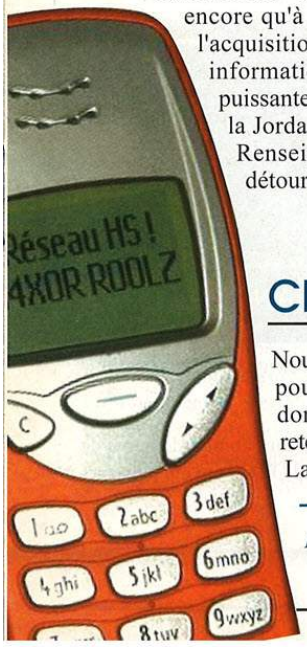
Mais on trouve aussi parmi les cyber-terroristes des sympathisans de groupes terroristes, ainsi que des hackers "patriotes", qui vont procéder à des actions de rétorsion juste après des attaques "physiques" (réelles) ou logiques (sur les réseaux) de ceux qu'ils considèrent comme leurs ennemis **[1]**. En effet, le terrorisme et l'anti-terrorisme s'emparent d'Internet. Ainsi, tout un chacun peut maintenant faire de l'anti-terrorisme de sa propre initiative, sur une base individuelle, pour le plaisir

fiction ou menace réelle ?

de se faire peur : par exemple, le groupe de cyber-antiterroristes créé et mené par le hacker allemand à la réputation controversée Kim Schmitz, alias Kimble [2]. Celui-ci a créé le ronflant "Yihat" (en référence au Jihad), acronyme de *Young Intelligent Hackers Against Terror*, qui se propose de pourchasser les terroristes sur Internet. Ses membres prétendent s'être déjà introduits sur les sites de l'Al Shamal Islamic Bank au Soudan et de l'Arab National Bank à Ryad et y avoir trouvé des données financières concernant Al Qaida et Ben Laden.

On peut aussi citer les attaques de hackers chinois contre des sites américains après le bombardement de l'ambassade chinoise au Kosovo en 1999, les attaques d'Américains contre des sites chinois lors de l'épisode de l'avion espion américain bloqué sur le sol chinois, et les attaques d'autres groupes de hackers américains (les "Dispatchers" notamment) contre les sites talibans [3] en 2001 (voir plus loin). On se rapproche dans ce dernier cas du cyber-combat [4], qui est l'équivalent sur les réseaux de l'affrontement de soldats sur un champ de bataille : les parties sont clairement identifiées et les règles d'engagement obéissent même à un code implicite se rapprochant de la convention de Genève. En effet, les hackers américains ont visiblement fait attention d'attaquer uniquement les sites gouvernementaux taliban, en laissant de côté les sites "civils". Mieux encore, lorsque ces hackers ont "abattu" une cible civile par erreur, ils se sont excusés publiquement, constituant ainsi ce qu'on peut qualifier de premier cyber-dommage collatéral déclaré.

Enfin, un dernier type de cyber-terroristes est constitué par des états. Comme il existe des "états terroristes", on commence à observer des "états cyber-terroristes". Certains n'en sont encore qu'à la phase de préparation, notamment à l'acquisition par différents moyens d'équipements informatiques performants. Ainsi, un lot de puissantes machines vendues par les Etats-Unis à la Jordanie et destinées à l'origine à équiper les Renseignements Généraux de ce pays, a été détourné au profit de la Libye [5].



CIBLES ET IMPACTS

Nous avons vu que les cyber-attentats avaient pour but de causer un maximum de dommages et/ou un maximum de retentissement médiatique, culturel ou social. La simple défiguration ("defacement") de sites Web peu importants constitue donc à peine le premier niveau des cyber-attentats. Ceux-ci consisteront plutôt à faire tomber des sites critiques ou de grande visibilité, ou à rendre inopérantes les infrastructures critiques d'un pays ou d'une organisation. On peut aussi considérer la corruption de données vitales comme un cyber-attentat, puisque la confusion et la chute de confiance créées seront de nature à porter préjudice à la société.

Les cibles des cyber-attentats seront donc constituées prioritairement par :

- Les installations de gestion des télécommunications (centraux téléphoniques, points d'accès GSM, réseaux filaires et non filaires, relais hertziens et satellites)
- Les sites de génération et de distribution d'énergie (centrales nucléaires, thermiques, sites de régulation EDF)
- Les installations de régulation des transports (aéroports, ports, contrôle aérien et maritime, gares ferroviaires et routières, autoroutes, systèmes de régulation des feux rouges des grandes agglomérations)
- Les installations de distribution de produits pétroliers (raffineries, dépôts, réseaux de stations services)
- Les centres de gestion du courrier postal
- Les sites de distribution d'eau (centres d'analyse, usines de traitement, stations d'épuration)
- Les institutions financières et bancaires (bourses nationales, réseau SWIFT, *home banking*, réseaux de distributeurs de billets)
- Les services d'urgence, de santé et de sécurité publique (police, pompiers, SAMU, hôpitaux)
- Les services gouvernementaux (sécurité sociale, assurance maladie, sites institutionnels)
- Les médias (chaînes de télévision, groupes de presse, fournisseurs de contenus divers)
- Les éléments symboliques d'une société et d'un mode de vie (grande distribution, industries représentatives, ...)

Une attaque sur plusieurs de ces cibles simultanément pourrait avoir un effet dévastateur pour un pays non préparé.

Paralysie des systèmes de communication

Le moment choisi pour les attaques est également important. Les cyber-terroristes choisiront par exemple de frapper en même temps que des événements politiques ou militaires, ou bien quand l'attention est dirigée dans une autre direction. Ils profiteront également du moment où les procédures se relâchent et où le personnel de surveillance tombe dans la routine. Mais, comme nous l'avons vu, c'est surtout en réaction à des attaques terroristes ou contre-terroristes que les pirates choisissent de frapper.

Les impacts liés à l'attaque des cibles précédentes peuvent être très variés : économiques (des actions ou une bourse peuvent s'effondrer, des entreprises faire faillite), sociaux (chômage, perte de certaines prestations, perte de son "identité sociale"), environnementaux, vitaux. Dans tous les cas, la confusion et la chute de confiance suivant les attaques seront de nature à porter préjudice à la société en général. Les gênes importantes dans les opérations de la vie courante, qui peuvent aller jusqu'au blocage total de certaines fonctions du pays (distribution de billets, d'essence, de produits frais), constituent des dommages majeurs. Certains dommages peuvent même constituer une menace sur la vie de certains individus : ainsi, la mise hors service des systèmes de contrôle de refroidissement des réacteurs d'une centrale nucléaire peut conduire rapidement à un accident radiologique majeur (surtout si la chute automatique des barres de secours a été désactivée), nécessitant l'évacuation d'une zone considérable, avec risque vital à plus ou moins long terme sur la population la plus touchée. De même, un aéroport privé de ses systèmes de contrôle aérien aura beaucoup de mal à éviter des collisions, voire des crashes d'appareils. Enfin, un système de traitement de l'eau victime d'une attaque pourra rendre dangereuse une eau qui n'aura pas été suffisamment chlorée, provoquant potentiellement des épidémies.

Le cyber-terrorisme a parfois été qualifié de terrorisme sans mort. Cela pourrait changer à l'avenir.

HISTORIQUE

L'historique du cyber-terrorisme montre que les attaques évoquées précédemment ne sont pas seulement théoriques. Sans être exhaustif, voici quelques événements marquants dans l'histoire du cyber-terrorisme :

■ **En 1996**, un sympathisant du mouvement américain White Supremacist a attaqué et temporairement mis hors service un ISP qui tentait de l'empêcher d'envoyer en masse des messages racistes. L'attaquant avait alors envoyé le message prémonitoire suivant : "Vous n'avez pas encore vu de vrai terrorisme électronique. C'est une promesse" [6].

Les scénarios de paralysie des transports...

■ **En 1997** et les années suivantes, des sympathisants du mouvement zapatiste mexicain ont effectué des intrusions à plusieurs reprises dans les systèmes logistiques mexicains et ont contribué à influencer l'opinion publique en faveur des Zapatistes, dont la situation, face à l'armée mexicaine, était critique. Des agents d'influence ont propagé des rumeurs sur l'instabilité du peso mexicain, ce qui a conduit à un effondrement de celui-ci, obligeant le gouvernement à négocier avec les rebelles.

■ **En 1998**, des militants espagnols ont attaqué l'IGC américain (*Institute for Global Communications*) en effectuant un mail bombing en direction des responsables de l'ISP et des commandes effectuées avec des faux numéros de carte bancaire. Ils menaçaient en outre d'attaquer les autres clients de l'ISP. Ces militants reprochaient à l'IGC d'héberger le site Web du journal Euskal Herria, une publication basée à New York et soutenant le mouvement indépendantiste Basque, et en particulier de soutenir le terrorisme car une partie du site contenait des informations concernant l'ETA. L'IGC a fini par retirer le site incriminé.

■ **En 1998**, la guerrila tamoule dans le nord du Sri Lanka a engorgé les serveurs des ambassades sri-lankaises avec environ 800 e-mails par jour pendant deux semaines. Les e-mails contenaient le message suivant: "Nous sommes les Tigres Noirs d'Internet et nous allons interrompre vos communications". Les services de renseignement ont qualifié ces attaques comme étant les premières attaques connues de terroristes contre les systèmes informatiques d'un état.

■ D'autres exemples non datés sont plus difficilement vérifiables: en Floride, des attaquants auraient détourné les appels au 911 (la police, aux Etats-Unis) vers un magasin de pizzas à emporter. Plus grave, au Massachusetts, un pirate a provoqué la coupure des communications d'une tour de contrôle de la FAA pendant 6 heures. En Russie, des pirates auraient utilisé un collaborateur interne de Gazprom (organisme qui détient le monopole du pétrole en Russie) pour implanter un cheval de Troie leur permettant d'obtenir le contrôle du système de distribution qui gère les flux de pétrole dans les pipe-lines.

... sont à envisager sérieusement.

■ **En 1999**, pendant le conflit du Kosovo, les ordinateurs de l'OTAN ont été les cibles de mail bombing et de tentatives de dénis de service de la part d'opposants aux bombardements de l'OTAN, dont certains situés dans les états en conflit. Les serveurs de l'OTAN ont été plusieurs fois mis hors service pendant plusieurs jours.

■ **En 1999**, après le bombardement "non tactique" de l'ambassade chinoise à Belgrade, des attaquants chinois ont déposé des messages du type "nous ne cesserons d'attaquer jusqu'à ce que la guerre s'arrête" sur des sites gouvernementaux américains.

■ **En février 2001**, un serveur (heureusement de développement) du fournisseur d'électricité California ISO [7] a été laissé connecté à Internet pendant 11 jours et, bien sûr, hacké.

■ **En avril 2001**, après la collision au-dessus de la Chine entre un avion espion américain et un chasseur chinois, et l'incarcération de l'équipage américain en Chine, des groupes de hackers des deux camps (comme les groupes "Honker Union of China" et "Chinese Red Guest Network Security Technology Alliance", en Chine) se sont menés une guerre violente. Plus de 1200 sites américains ont été défigurés ou cibles d'attaques de type déni de service distribué (DDoS), dont les sites de la Maison Blanche, de l'US Air Force, du Département de l'Energie, mais aussi d'entreprises diverses.

... voire de crash d'avions...

■ **En août 2001**, le ver Code Red, qui s'est propagé à très grande vitesse sur Internet, faisait apparaître le message "Hacked by Chinese". Même si aucune preuve n'atteste avec certitude que ce ver provient de Chine, on ne peut s'empêcher de mettre ce message en relation avec les attaques de 1999 (cf ci-dessus). La charge utile de ce ver consistait à établir un grand nombre de connexions vers le site Web de la Maison Blanche afin de provoquer un déni de service distribué. Depuis, d'autres vers (Nimda, Slammer par exemple) ont défrayé la chronique (voir paragraphe suivant).

■ Si l'on considère que la défiguration de sites Web constitue le tout premier degré de cyber-attentat, on peut étudier les conflits Inde/Pakistan et Israël/Palestine depuis 1999 jusqu'à aujourd'hui à la lumière des défigurations de sites appartenant aux différentes parties. On observe un strict parallèle entre le nombre de défigurations de sites et les événements politiques et militaires dans les régions citées. Cette comparaison révèle une connexion intime entre les conflits qui ont lieu dans le monde physique et dans le monde virtuel.

■ **En 2001 et 2002**, plusieurs documents retrouvés en Afghanistan et lors des enquêtes sur les réseaux d'Al Qaida ont montré que le cyber-terrorisme était activement étudié par Oussama Ben Laden, passionné par ce type de guerre moderne. Il a consacré des sommes importantes au recrutement dans le monde arabe des meilleurs informaticiens et spécialistes d'Internet. Un plan en ce sens lui avait été remis en juin 2001 par un intégriste séoudien de Londres. D'ailleurs, depuis la capitale britannique, les réseaux Internet intégristes sont de plus en plus actifs [8].

LES ARMES DES CYBER-TERRORISTES

Les cyber-terroristes ont à leur disposition, comme nous l'avons vu dans les paragraphes précédents, plusieurs types d'armes logiques pour accomplir leurs attaques. Ces armes sont de complexité et de portée différentes, et leurs impacts peuvent être plus ou moins forts. Parmi les armes les plus courantes, on trouve:

■ Les défigurations de sites Web et les "attaques sémantiques", qui sont utilisées aussi bien par les "hacktivists" que par les cyber-terroristes. Les attaques sémantiques consistent à modifier légèrement le contenu des pages Web afin d'en changer le sens, pour faire passer une idée différente de celle d'origine. Cette modification est difficile à détecter pour le webmaster, contrairement à la défiguration simple qui change complètement l'apparence du site Web.

■ Les dénis de service simples (DoS), utilisant soit des vulnérabilités précises du système d'exploitation, soit utilisant des techniques plus génériques comme le SYN flood.

■ Les dénis de service distribués (DDoS), utilisant un grand nombre de serveurs compromis sur lesquels tournent des programmes "zombies" attendant les instructions de ceux qui les ont implantés et qui les contrôlent à distance. De véritables "DDoSNets", constitués par des zombies capables de dialoguer entre eux (en *peer to peer*) et avec leur point de contrôle, se constituent actuellement, qui permettront de lancer des attaques coordonnées aussi rapides que meurtrières sur des cibles bien précises. A cause de ces DDoSNets, les serveurs non sécurisés de n'importe quelle entreprise peuvent se transformer en armes aux mains de cyber-terroristes. De même, les ordinateurs personnels connectés en permanence à Internet par ADSL ou par le câble constituent des cibles privilégiées pour ces DDoSNets, et peuvent, là encore, se transformer en armes.

■ Les attaques sur les serveurs DNS et les équipements de routage. Les vulnérabilités des protocoles de routage comme BGP, sensible au *poisoning*, associées à des attaques sur les *root servers DNS*, peuvent conduire à une paralysie de certaines parties d'Internet : c'est le phénomène de trou noir, où les informations à destination de certains sites disparaissent complètement. De plus, la majorité des routeurs utilisant l'OS de Cisco (IOS), de nouvelles vulnérabilités mises à jour dans IOS conduiraient à des attaques massives.

■ Les vers : Code Red, Nimda, Lion, Adore, Slammer ont montré leurs capacités. Certains prétendent même que le ver Slammer a infecté Internet en 10 minutes seulement [9]. Heureusement, celui-ci ne comportait pas de charge utile hostile et ne résidait qu'en mémoire. On peut imaginer le résultat d'un ver de ce type qui serait capable de mettre hors service instantanément les serveurs infectés, ou de modifier des informations sur ceux-ci...

■ Les intrusions classiques, permettant d'implanter des chevaux de Troie, de récupérer des données sensibles ou de mettre hors service des serveurs internes non accessibles autrement.

■ La modification furtive de données, suite à une intrusion ou à l'action d'un ver, qui serait capable de décrédibiliser une entreprise ou une organisation, ou même de faire perdre toute confiance en une institution fondée sur cette confiance, comme par exemple la bourse.

■ L'implantation de bombes logiques, qui sont capables de se déclencher selon certains paramètres ou certains événements, et peuvent, là encore, faire tomber toute une série de serveurs en même temps, ou modifier des données critiques au moment opportun et de manière automatique.

LES COMMUNICATIONS, LE RECRUTEMENT ET LA FORMATION

Les groupes islamistes se servent d'Internet de manière intensive pour leur recrutement, puis pour la formation et l'endoctrinement de leurs recrues. Les perquisitions menées depuis le 11 septembre 2001 dans les milieux européens supposés être liés à Al Qaida ont montré qu'ils ont utilisé des sites Web de recrutement de mercenaires **[10]**, comme le site Qoqaz, par exemple **[11]**. Les mêmes cassettes vidéo ont été trouvées dans une quinzaine d'appartements perquisitionnés. Ces cassettes de formation et d'endoctrinement sont vendues sur Internet, sur des sites comme Maktabah **[12]**. De même, des newsgroups et des listes de diffusion spécialisés dans la diffusion de sélections documentaires centrées sur l'Islam intégriste présentent les activités des groupes activistes et terroristes au nom de la liberté d'expression **[13]**.

Quant aux communications opérationnelles, les cyber-terroristes savent aller chercher leurs instructions sur l'un des nombreux sites Web entretenus par leur organisation (Al Qaida a installé des centres de communication Internet à Lahore, Karashi, dans les villages pakistanais du Baloutchistan, d'autres sites sont installés au Cashemire), ou encore sur des channels IRC secrets ou totalement banalisés et créés temporairement, pour l'occasion. Certains recommandent même d'utiliser des messageries instantanées de type ICQ.

La multiplication des cybercafés dans le monde entier, y compris et surtout dans les pays en voie de développement, rend la filature des internautes très aléatoire. Pour un dollar chacun, depuis les cybercafés ou les centres d'affaires des hôtels du monde entier, les cyber-terroristes d'une même "cellule-action" peuvent communiquer pendant un quart d'heure en direct, sous un pseudonyme, et s'échanger des informations sensibles relatives à un cyber-attentat avant de se volatiliser dans la nature. S'ils le jugent utile, ils peuvent confirmer leurs instructions par des messages chiffrés envoyés depuis des messageries anonymes. Mais à quoi bon alerter, par un contenu chiffré, la vigilance des services censés surveiller le contenu du trafic Internet, alors qu'il est tout aussi simple de cacher des messages en clair dans une image anodine ? Vu le nombre d'images qui transitent quotidiennement sur Internet, il n'y a aucune chance que le moindre message caché soit détecté. Point n'est besoin d'utiliser de complexes programmes de stéganographie, utilisant des techniques de masquage élaborées, comme certaines rumeurs en ont fait état. Le texte en clair ou tout juste brouillé est tout simplement ajouté au contenu du fichier image.

PRINCIPES DE DÉFENSE

Les conflits actuels étant de plus en plus accompagnés par des attaques informatiques, la menace de cyber-terrorisme doit maintenant être intégrée à toute politique de sécurité. Les principes de défense à appliquer sont ceux de la sécurité des systèmes d'information en général : vous avez entre les mains une excellente source de référence pour cela. Il faut donc travailler sur les concepts de défense en profondeur, coupler la sécurité organisationnelle et la sécurité logique, sans oublier la sécurité physique.

En ce qui concerne la sécurité technique, la lecture de MISC vous apportera les bases nécessaire. Pour mémoire, on peut citer :

■ Maintenir un état de vigilance élevé

■ Effectuer ou faire effectuer une veille technique concernant les dernières vulnérabilités publiées

■ Mettre à jour très régulièrement les systèmes d'exploitation et les applications

■ Limiter le nombre de services disponibles sur les serveurs et désactiver tous les autres

■ Sécuriser les configurations, et en particulier changer tous les mots de passe par défaut, appliquer des mots de passe solides, et utiliser le principe du moindre privilège pour faire tourner les services

■ Mettre en place un filtrage d'accès en entrée mais aussi en sortie afin que votre plate-forme ne puisse pas être utilisée comme source d'attaques vers d'autres cibles

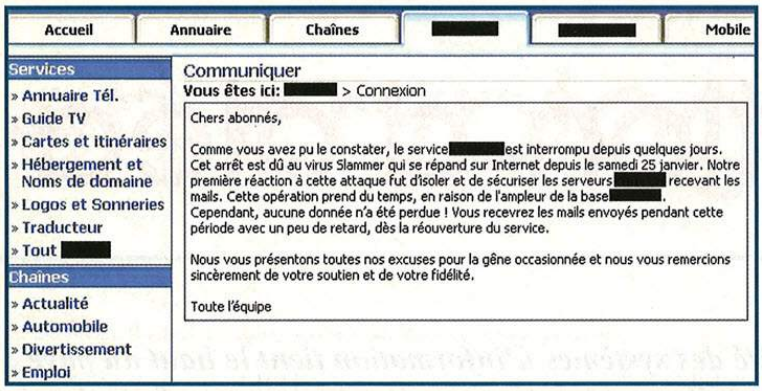
■ Installer des anti-virus côté serveurs et côté clients et les mettre à jour très souvent

■ Activer les systèmes de journalisation disponibles sur les systèmes et les applications; centraliser, analyser et sauvegarder les logs régulièrement

■ Utiliser des IDS correctement configurés et analysés régulièrement

■ Effectuer des sauvegardes régulières, les stocker dans un endroit sûr et tester les procédures de restauration dé manière régulière

■ Effectuer des sauvegardes



Le ver Slammer a conduit à l'interruption de nombreux services

Il ne faut pas non plus oublier les risques internes (voir le cas de Gazprom cité précédemment), d'autant plus que les cibles sont de plus en plus constituées par les postes de travail internes, moins sécurisés et opérés par du personnel non formé aux questions de sécurité. La sensibilisation du personnel est donc primordiale, tout comme la sécurisation des postes de travail, et en particulier des navigateurs Web et des clients de messagerie (voir MISC N° 1).

L'AVENIR

Les risques à venir ont de grandes chances de provenir de DDoSNets de plus en plus élaborés et de vers de plus en plus intelligents, qui vont certainement voir le jour. Des chercheurs ont prédit l'émergence de nouvelles sortes de vers (*Warhol worms*, *flash worms*), qui pourraient se diffuser sur Internet en quelques minutes ou même quelques secondes, laissant peu de temps aux administrateurs pour réagir. Des vers hybrides combinant un ensemble de vulnérabilités anciennes, afin de maximiser leurs chances d'infection, ou bien des vers exploitant des vunérabilités non encore publiées, et donc non patchables immédiatement, vont certainement voir le jour. De tels vers ne laisseront pas d'autre alternative que de stopper les services en attendant la diffusion du correctif de sécurité par les éditeurs.

Pour l'instant, les vers que nous avons vus passer étaient d'une technologie assez fruste. Des vers plus sophistiqués et intelligents, pouvant se mettre à jour de manière autonome en allant télécharger des plug-ins plus récents sur des sites précis, pouvant aller chercher des nouvelles cibles et des instructions sur des channels IRC, pourront effectuer des besognes beaucoup plus dangereuses tout en étant beaucoup plus discrets.

CONCLUSION

Les frappes physiques sont de plus en plus menées en parallèle avec des frappes logiques. Le cyber-terrorisme fait maintenant intégralement partie des tactiques des groupes terroristes, et les attaques informatiques augmentent en volume, en sophistication et en coordination.
Face à cette menace de plus en plus pressante, l'Europe a commencé à réagir. En lançant le projet de Cyber Security Task Force, la Commission Européenne entre dans la première phase d'élaboration d'une doctrine par l'expression des besoins, c'est-à-dire l'énumération des vulnérabilités générées par la société de l'information. La Commission Européenne a sollicité la Rand Corporation Europe pour réfléchir sur cette typologie des vulnérabilités. Si nul ne songe à contester l'expertise de ce *think tank*, un certain nombre de réserves peuvent être formulées sur l'objectivité de l'analyse de ce cabinet américain... **[14]**

Patrick Chambet - http://www.chambet.com
Consultant Senior - Edelweb - Groupe ON-X
http://www.edelweb.fr - http://www.on-x.com

Crédits photo: P.C. - N.R.

RÉFÉRENCES

[1] *"Cyber Attacks During the War on Terrorism: A Predictive Analysis."* Dartmouth Institute for Security Technology Studies. **http://www.ists.dartmouth.edu/ISTS/counterterrorism/cyber_attacks.htm**

[2] http://www.kimble.org/mostwanted.htm http://www.kill.net

[3] Le mot Taliban est un pluriel.

[4] Le cyber-combat sera traité par ailleurs.

[5] Le Monde du Renseignement No 425 du 14 mars 2002.

[6] Dorothy Denning, *"Hacktivism, and Cyberterrorism: The Internet as a Tool for Influencing Foreign Policy."* **http://www.nautilus.org/info-policy/workshop/papers/denning.html** (12 décembre 2001)

[7] http://www.caiso.com

[8] Roland Jacquard, *"Les archives secrètes d'Al Qaida"*, Ed. Jean Picollec.

[9] http://www.cnn.com/2003/TECH/internet/02/05/virus.spread.reut/index.html

[10] Roland Jacquard, *"Les archives secrètes d'Al Qaida"*, op.cit.

[11] http://www.qoqaz.com http://www.cybcity.com/azzamjihad/ Fatwas: **http://www.faharis.net/fatwa.shtml**

[12] http://www.maktabah.net/home.asp

[13] Jean Guisnel, *"Guerres dans le cyberespace"*, Ed. La Découverte

[14] Christian Harbulot, *"La guerre cognitive: A la recherche de la suprématie stratégique"*, 25 septembre 2002.

Génération d'aléa

À l'heure actuelle, la sécurité des systèmes d'information tient le haut du pavé : authentification, signature, chiffrement, intégrité... Il est à la mode de parler de PKI, d'infrastructures sécurisées. Cependant, tout ce beau monde n'est sûr que tant que le maillon le plus faible est sûr, et il est un lieu commun à ces divers domaines, un prérequis tellement banal, si simple que, bien souvent, on ne l'évoque même plus en public : il existe des moyens simples d'engendrer de l'aléa cryptographique.

PRÉAMBULE

Combien de fois trouve-t-on : je tire au hasard un module RSA, une clé secrète, un bidule privé... et on peut continuer tant qu'on veut... Tout cela se fait en pratique par ce que l'on appelle vulgairement des générateurs d'aléa, en fait de pseudo-aléa, mais on verra pourquoi. Ce sont des procédés qui permettent d'engendrer des 0 et des 1, de la même manière qu'on tire à pile ou face. Mais si votre générateur d'aléa est biaisé, quand bien même vous auriez les meilleurs algorithmes et infrastructures du monde, tout cela ne vaudra rien. L'utilisation de l'AES, du DES, et des différents systèmes de chiffrement par blocs impose que l'on tire aléatoirement des blocs de bits. Si quelqu'un de malveillant met la main sur votre générateur d'aléa, il est possible qu'il puisse reconstituer tout ou partie de cette clé privée que vous gardez jalousement. Et en général, tous ces générateurs sont publics. Quelles garanties a-t-on sur la qualité de l'aléa qu'ils engendrent ?

Cet article n'est pas dédié à une dissertation sur ce qu'est le hasard, puisque – hormis en physique quantique – celui-ci n'a pas de réalité tangible, mais sur les moyens d'engendrer de l'aléa qui soit cryptographiquement sûr. Le but est également de sensibiliser le lecteur à ce problème crucial. Mais ne désespérons pas, il existe de bons générateurs de pseudo-aléa cryptographique, mais ils ne sont pas forcément implantés par défaut dans votre système d'exploitation.

Avez-vous confiance dans votre `random()` ?

ALÉA ET CHIFFREMENT INCONDITIONNELLEMENT SÛR : LE SYSTÈME DE VERNAM

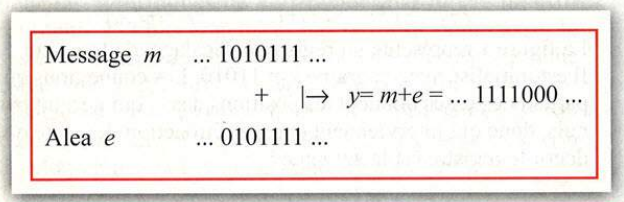
Depuis fort longtemps – disons au moins depuis la Renaissance – certaines personnes se sont préoccupées de trouver le système cryptographique "incassable" qui permettrait à deux personnes de communiquer de telle sorte que toute autre personne que le destinataire du message chiffré serait incapable de décrypter ce message. Eh bien, ce chiffrement que l'on appelle "inconditionnellement sûr" existe. Il a été découvert au début du siècle par Vernam. Shannon a prouvé quelques décennies plus tard qu'il était impossible de le casser. Par impossible, j'entends quelle que soit la puissance de calcul dont on dispose, et quel que soit le temps dont on dispose. C'est impossible, point final.

Le principe du chiffrement de Vernam est le suivant :

- *Chiffrement* : Alice veut envoyer un message m de longueur ℓ bits à Bob. Alice dispose d'un générateur d'aléa A, qui engendre bit sur bit. C'est un procédé qui permet d'engendrer des bits aléatoirement. A produit une suite de e bits de longueur. Alice envoie $y = m+e$ à Bob.
- *Déchiffrement* : Bob calcule $y - e = m$, et retrouve ainsi le message m.

Ce système est donc un système de chiffrement à clé secrète dont la clé est la suite de bits e.

en cryptographie



Principe du chiffrement de Vernam

Ce système de chiffrement est trivial et incassable, alors pourquoi ne l'utilise-t-on pas comme système de chiffrement à clé secrète, à la place de tous les systèmes que nous connaissons comme le DES, l'AES... Il existe plusieurs raisons à cela, mais la principale est que nous ne savons pas construire du vrai aléa qui satisfasse à la fois les exigences de vitesse et de sécurité.

En effet, le système de Vernam a été prouvé incassable pourvu que le processus de génération du bloc de bits e soit parfaitement aléatoire. Cette exigence se traduit dans les faits par :

1) ***Uniformité de la génération de bits*** **:** pour chaque bit aléatoire engendré, il y a une chance sur deux de tirer un 1, et donc une chance sur 2 de tirer un 0.

2) ***Indépendance de la génération de bits*** **:** quel que soit le nombre de bits de la suite aléatoire dont on dispose, il est impossible de deviner quel prochain bit a le plus de chances de sortir.

Un système vérifiant la première assertion est assez facile à construire. N'importe quelle fonction `random()` implantée dans vos systèmes doit la vérifier. Par raccourci, on dit qu'un tel système produit du bon *aléa statistique*. En général, cet aléa suffit pour balayer l'ensemble des possibilités de façon uniforme, dans les expressions du style "je tire au hasard un nombre entre 0 et 100".

En revanche, construire un système vérifiant la seconde assertion est très problématique. En effet, imaginons que je suis un attaquant du système de Vernam et que je parvienne à me procurer le bloc e de bits engendré par le générateur aléatoire A d'Alice utilisé dans un système de Vernam. Supposons que

$$e = 1010101010101010101$$

Bien que la fréquence d'apparition des 0 et des 1 soit la même, il saute aux yeux que le bloc e est extrêmement structuré. En tant qu'attaquant, je vais me dire que, bien que la génération d'aléa par A soit uniforme, la prochaine séquence de bits sera encore 0101010... J'en déduirai donc que si A produit le bit 1, il y a de bonnes chances que le bit suivant soit un 0 et celui d'après un 1... J'aurai donc découvert ce que l'on nomme un biais statistique dans le générateur A. Ici, il est facile de l'utiliser pour attaquer le système, puisque l'on connaît la structure des suites de bits engendrées par A.

Un autre problème survient. Comme engendrer du bon aléa est difficile, on pourrait être tenté de réutiliser un bloc de bit e que l'on a déjà utilisé. Malheureusement, cette option est très dangereuse. En effet, supposons que deux messages m_1 et m_2 aient été chiffré avec le même bloc e, $y_1 = m_1 + e$ et $y_2 = m_2 + e$. L'attaquant qui intercepterait ces blocs n'aurait qu'à calculer la différence $y_1 - y_2 = m_1 - m_2$. Cette quantité est indépendante de l'aléa et fournit ainsi de précieuses informations sur les messages m_1 et m_2 initiaux.

PSEUDO-ALÉA

L'exemple précédent nous montre que pour avoir un système de chiffrement inconditionnellement sûr, il faut au minimum produire beaucoup de vrai aléa, et le produire à la volée, dès qu'arrive un message à chiffrer. A l'heure actuelle, nous ne savons pas le faire. Pourtant, l'idée de Vernam a poursuivi son chemin et constitue aujourd'hui le cœur des systèmes de chiffrement à flot, du type "téléphone rouge", dans lesquels on fabrique du bon "pseudo-aléa" cryptographique à partir d'une petite quantité de "bon aléa", la graine. Pour ces raisons, on appelle ces systèmes *générateurs pseudo-aléatoires*.

Leur principe de fonctionnement est le suivant :

■ On commence par engendrer quelques centaines de bits de *bon aléa*, en prenant le temps qu'il faut, et en vérifiant la qualité de l'aléa engendré par une batterie de tests existants. Ces quelques centaines de bits de bon aléa vont constituer la graine – *seed* – de notre générateur.

Pour fabriquer le nombre de bits aléatoires constituant la graine, il y a plusieurs méthodes. Par exemple, pour les fonctions `rand()` ou `random()` de Linux, on évalue une fonction de beaucoup de paramètres, en espérant que cela sera suffisamment aléatoire. On peut compter le nombre de cycles du processeur entre deux événements, prendre les temps d'accès mémoire... La génération de graine ressemble en général assez à de la cuisine. Cependant, il faut rester prudent, et faire passer à ce générateur de graine

certains tests pour vérifier qu'il est *suffisamment* aléatoire. Typiquement, on utilise ce genre de générateur pour engendrer quelques centaines de bits d'aléa, qui jouent le rôle de clés secrètes de systèmes de chiffrement par blocs du type AES, ou bien DES. Malheureusement, génération de bon aléa et vitesse sont contradictoires. Ils ne sont donc pas utilisables dans des chiffrements à flot.

■ A partir de cette graine, on construit du pseudo-aléa, en utilisant différents outils mathématiques. L'avantage d'un tel système est que l'on peut contrôler la vitesse de génération du pseudo-aléa ainsi que la qualité du pseudo-aléa.

Si, pour un chiffrement de type Vernam, on utilise un générateur pseudo-aléatoire, le système n'est plus "inconditionnellement sûr". En effet, on peut énumérer l'espace des graines qui est fini en temps fini. Une des conséquences est que la taille de la graine doit être suffisamment grande pour qu'il soit impossible d'énumérer l'espace.

Voici deux exemples de générateurs pseudo-aléatoires. Je tiens à préciser qu'ils sont tous les deux mauvais du point de vue cryptographique, mais ils ont l'avantage d'être simples à décrire.

Le (m, ℓ)-générateur linéaire par congruences :

■ On prend un entier $n = 2^m - 1$. On se donne deux entiers $1 \leq a,b \leq n-1$. Soit ℓ, un entier, tel que $m+1 \leq \ell \leq n-1$. L'entier ℓ est la longueur des blocs engendrés par le générateur. Soit $s_0 \leq n$. s_0 est un entier de m bits et joue le rôle de la graine.

■ On définit la suite

$$s_i = a\, s_{i-1} + b \bmod n, \quad i \leq 1.$$

■ La suite de bits engendrée par le (m, ℓ)-générateur linéaire par congruence est définie par $(e_1, e_2..., e_\ell)$, où $e_i = s_i \bmod 2$.

Prenons un petit exemple avec les paramètres suivants : m =4, n=15, a=2, b=1 et ℓ =7. Alors, la table 2 nous donne les suites de bits engendrées en fonction de la graine.

Graine	Suite pseudo-aléatoire
0	1110111...
1	1101110...
2	1100110...
4	1010101...
5	1000100...
14	0000000...

Table 2: (4,7)-générateur par congruences linéaires

En fait, ce générateur est assez rapide mais n'est pas sûr du tout d'un point de vue cryptographique, car trop linéaire.

Les Registres à décalage à rétroaction linéaire :

La structure la plus simple, et de très loin la plus rapide, qui a été envisagée pour construire du pseudo-aléa est le registre à décalage à rétroaction linéaire dit encore (LFSR). Cet objet est défini par sa longueur et par ses coefficients de rétroaction. Un registre de longueur ℓ est initialisé avec une graine, ensemble de bits aléatoires de longueur ℓ, $(e_1,e_2..., e_\ell)$. Une fois l'initialisation du registre fixée, les bits suivants s'obtiennent par la formule de récurrence :

$$e_{\ell+i} = c_1\, e_i + c_2\, e_{i+1} + \cdots + c_\ell\, e_{i+\ell} - 1$$

La figure 1 représente un registre à décalage, de longueur 5. Il est initialisé avec la graine s_0= 11010. Les connexions qui partent des cases donnent les positions des c_i qui ne sont pas nuls, donc qui interviennent dans la rétroaction. La suite qui décrit le registre est la suivante :

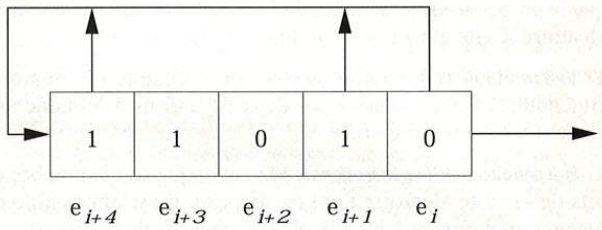
$$e_{i+5} = e_i + e_{i+1} + e_{i+4},$$

et le registre donne comme suite pseudo-aléatoire :

01011010010111010...

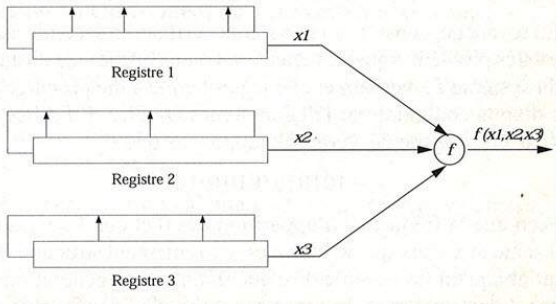
On connaît beaucoup de choses d'un point de vue mathématique sur les LFSR. Ils ont de bonnes propriétés

1



LFSR de longueur 5

2



Combinaison de LFSR's avec une fonction booléenne.

d'aléa statistique, mais leurs propriétés sont catastrophiques d'un point de vue cryptographique. En effet, il suffit d'au plus $2\ell+1$ bits de la suite engendrée par le LFSR pour retrouver la graine, qui sous-tend tout le système. La méthode qui consiste à retrouver la graine est connue sous le nom d'algorithme de Berlekamp-Massey.

Donc, si on prend une graine de 128 bits, qui est habituellement considérée comme une très bonne sécurité pour les algorithmes de chiffrement par blocs, il suffit d'obtenir 257 bits engendrés par le LFSR pour casser le système. Partant, il faut n'envoyer que des messages de taille inférieure à 257 bits – c'est-à-dire de moins de 80 – caractères. Pas terrible non ? Ce problème est dû essentiellement à la linéarité de ces registres. En cryptographie, tout ce qui est linéaire n'est pas bon...

Afin de pallier cet inconvénient, les concepteurs de chiffrement à flot utilisent plusieurs LFSR combinés par une fonction booléenne qui prend en entrée des sorties des LFSR et renvoie des bits - 0 ou bien 1 en sortie. Cette fonction sert justement à casser la linéarité des registres. C'est une des raisons pour lesquelles les fonctions booléennes sont des objets très étudiés d'un point de vue théorique.

CONCLUSION

La conclusion de cet article est que la prudence s'impose lorsque quelqu'un vous fournit système cryptographique clé en main. Avant toute chose, il est utile de vérifier que le générateur pseudo-aléatoire est cryptographiquement sûr, sinon on s'expose à quelques surprises. Pour ce faire, il existe des batteries de tests que l'on peut sans problème trouver sur la toile.

Pierre Loidreau

RÉFÉRENCES

D. Stinson, *Cryptographie : Théorie et pratique.*

http://www.tcs.hut.fi/ helger/crypto/link/ pseudorandom/

http://www.cs.berkeley.edu/ daw/rnd/

Un virus de boot

Si la menace des virus de boot a considérablement diminué depuis quelques années, ce notamment grâce aux antivirus qui offrent dans ce domaine une protection généralement efficace contre les types connus, cette menace n'a pas totalement disparue. L'inconscience de certains utilisateurs, les ouvertures toujours plus grandes offertes par la technologie, la multiplicité des périphériques "bootables", et surtout l'imagination des programmeurs de virus peuvent faire des virus de boot une menace encore d'actualité. Le problème est qu'en cas d'infection réussie, les risques sont alors importants de par la nature même de ce type de virus. Cet article présente les virus de boot, leurs principes et plus spécifiquement le virus STEALTH, qui offre des fonctionnalités de furtivité remarquables.

INTRODUCTION

L'intérêt principal des virus de *boot* réside dans le fait qu'ils interviennent avant le lancement du système d'exploitation (OS) (et donc de tout logiciel, en premier lieu l'antivirus !). Il est donc impossible de stopper son lancement au niveau de l'OS par le biais d'un quelconque antivirus. Ce dernier doit intervenir avant, c'est-à-dire directement au niveau du BIOS. Si plusieurs constructeurs de BIOS intègrent effectivement un tel antivirus, il faut reconnaître que leur efficacité est limitée : ils ne savent gérer que le démarrage de Windows. Autrement dit, un secteur de démarrage maître modifié aux fins de lancer un autre OS (Linux par exemple, mais aussi certaines autres versions de Windows) sera détecté comme infecté ! Les utilisateurs finissent alors par le désactiver. De plus, le contournement de ces antivirus de BIOS n'est pas impossible.

Agissant très tôt, un virus de boot peut éventuellement mettre en place un certain nombre de mécanismes lui permettant d'augmenter son efficacité, et en particulier de limiter ou d'interdire sa détection. C'est la raison principale pour laquelle les virus de boot représentent une menace toujours actuelle. Il faut malheureusement compter sur l'ingéniosité des programmeurs pour développer une toujours plus grande capacité de furtivité. Il faut surtout insister sur le fait, jamais envisagé mais pourtant logique, qu'un virus de boot peut concerner n'importe quel OS et pas seulement les systèmes Windows. Cela offre un champ de possibilités intéressant, notamment sous Linux ou autre Unix libres, qui deviennent de plus en plus répandus.

Pour bien comprendre le mode de fonctionnement d'un virus de boot et en comprendre les potentialités, nous allons étudier le virus STEALTH ("furtif" en anglais) créé par Mark Ludwig. Ce virus, qui reprend certains mécanismes du virus pakistanais BRAIN, est un condensé de tout ce qu'un virus de boot peut déployer de ruse pour parvenir à ses fins. Si ce virus sous sa forme publiée est désormais détecté par les principaux antivirus, il n'est pas absurde de penser que sous une forme modifiée, adaptée et améliorée, il parvienne à infecter de nombreux ordinateurs. STEALTH est en quelque sorte encore un virus actuel.

Le lecteur intéressé trouvera le code source détaillé de ce virus dans le livre de Mark Ludwig [2], sur lequel cet article est fondé.

LE PROCESSUS DE DÉMARRAGE

Afin de bien comprendre comment agit un virus de boot comme STEALTH, il faut d'abord savoir ce qui se passe lors de la séquence de démarrage.

LA SÉQUENCE DE DÉMARRAGE

A la mise sous tension, le processeur entre en mode réinitialisation, remet à zéro tous les emplacements mémoire, effectue un contrôle de parité de cette mémoire et initialise le registre CS (registre 16 bits agissant comme sélecteur d'accès aux adresses 32 bits des segments mémoires, le registre CS (*Code Segment*) gère les segments de code d'un programme) avec l'adresse FFFFH et le registre IP à zéro.

Ce registre contient les 16 bits de poids forts du registre 32 bits EIP (*Extended Instruction Pointer*), et indique l'adresse relative au début du segment de code considéré (*offset*) de la prochaine instruction à exécuter. L'adresse complète de l'instruction est alors donnée par CS:IP. Le code du BIOS (*Basic Input/Output System*), localisé à l'adresse FFFFH:0000H est alors lancé.

furtif : STEALTH

Le BIOS effectue un certain nombre d'opérations parmi lesquelles :

■ Parcours des principaux ports pour détecter et initialiser les périphériques présents (appel notamment aux interruptions 11H, reconnaissance matérielle, et 12H, reconnaissance de la mémoire). Il vérifie également que les périphériques indispensables pour la suite fonctionnent correctement.

■ Chargement de la table des vecteurs d'interruptions en mémoire basse (256 adresses des routines d'interruptions). Cette table est utilisée par le BIOS et l'OS pour la gestion et l'appel des interruptions (quand cela est le cas). La *BIOS Data Area* est également chargée. Localisée à l'adresse 0040H:0000H, elle permet la gestion des périphériques présents (par exemple, à l'adresse 0040H:0013H se trouve mémorisée la quantité de mémoire physique effectivement disponible).

Ensuite, le BIOS détermine s'il existe un périphérique amorçable (disquette, disque dur, CD-ROM, ZIP...), c'est-à-dire contenant un programme de démarrage de 512 octets en tête 0, piste 0 et secteur 1, caractérisé par la valeur des deux derniers octets (offset 1FEH; la valeur AA55H indique la présence d'un secteur de démarrage valide). Dans le cas de machines *multiboot*, ce secteur est dit principal. Il est alors chargé en mémoire par le BIOS à l'adresse 0000H:7C00H qui lui transfère le contrôle.

Le secteur de démarrage principal lance ensuite les programmes spécifiques à l'OS. Dans le cas du multiboot, il y a d'abord lecture de la table de partition présente dans le secteur maître, puis lancement du programme de démarrage localisé dans un secteur de démarrage secondaire de la partition correspondant à l'OS sélectionné.

Au final, il est essentiel de constater que jusqu'à cette dernière étape, tout le processus est indépendant du système d'exploitation. C'est ce fait qui rend l'action des virus de boot universelle et potentiellement très dangereuse. Leur action consiste donc à infecter le programme de démarrage contenu dans le secteur principal. Le plus souvent, cela se fait par l'intermédiaire d'une disquette oubliée dans le lecteur et sur laquelle on démarre la machine, ou bien le secteur de démarrage infecté du disque dur infecte celui des disquettes introduites dans le lecteur.
Il est également possible d'infecter (mais cela est plus beaucoup plus dur et risqué) le secteur de démarrage du disque dur directement à partie de l'OS.

Dans tous les cas, la tâche est difficile car la contrainte expresse est que ce programme doit conserver après infection une taille maximale de 512 octets. Nous allons voir comment un virus de boot comme STEALTH s'affranchit aisément de cela.

LE SECTEUR DE DÉMARRAGE

Afin de bien comprendre les mécanismes du virus, il convient de bien comprendre quelles données sont à sa disposition, en premier lieu celles contenues dans le secteur de démarrage. Ce sont les suivantes (pour plus de détails voir [1]) :

■ Des données concernant l'unité physique sur laquelle se trouve ce secteur (disquette, disque...). Elles concernent essentiellement l'organisation et la structure de l'unité (nombre de secteurs, de têtes de lecture, de FAT (*File Allocation Table*, structure de données permettant de déterminer si les secteurs sont affectés à un fichier ou non ou s'ils sont défectueux), nature du support, son type...). Le BIOS et le DOS s'en servent pour piloter l'unité.

■ La table des paramètres du disque. Elle permet au BIOS de pouvoir physiquement piloter l'unité et en contient les caractéristiques techniques. Une table par défaut est d'abord chargée par le BIOS, puis le secteur de démarrage la remplace par la table spécifique à l'unité.

Lors de son exécution, le secteur de démarrage recherche les fichiers systèmes spécifiques à lancer pour charger l'OS. Il lui faut pour cela déterminer où commence le répertoire racine de l'OS. Les données dont il dispose en son sein lui permettent un tel calcul (valeur FDRS, ou *First root directory secteur*) :

```
FDRS  = FAT_COUNT x SEC_PER_FAT + HIDDEN_SECS + FAT_START
```

Puis, à partir de ce répertoire, il doit charger le premier fichier système à lancer. Son adresse est donnée par la valeur FDS (*First Data Sector*) indiquée par :

```
FDS = FDRS + [(32 x ROOT_ENTRIES) + SEC_SIZE  - 1 ]/SEC_SIZE
```

Le nombre d'octets à charger est finalement trouvé dans la table de répertoire contenant toutes les données de fichiers présents.

LE VIRUS STEALTH

Le virus comprend trois parties :

■ La première est un secteur de démarrage viral (SDV) remplaçant le secteur original lors de l'infection.

■ La deuxième est le corps principal du virus (CPV). Le virus présentant de nombreuses fonctionnalités, la limite des 512 octets est trop contraignante, et le virus utilise pour cette partie six secteurs qu'il doit dissimuler. La tâche du SDV sera de charger ces six secteurs afin de restaurer le virus dans son intégralité.

■ La troisième consiste en une copie du secteur de démarrage avant infection (SDO).

L'action du virus se décrit alors simplement selon trois phases :

- Le secteur viral SDV est chargé lors du boot de la machine.
- Il charge en mémoire le corps principal du virus (CPV) de manière résidente. Le virus est alors pleinement actif et est susceptible d'infecter d'autres secteurs de démarrage.
- Le contrôle est alors redonné au secteur de démarrage originel (SDO) pour que la machine boote normalement. Cela se fait par simple lancement du secteur SDO. Cette dernière étape est particulièrement astucieuse car elle permet au virus d'être totalement indépendant du système d'exploitation.

LA ROUTINE DE RECHERCHE

Cette partie a pour fonction de rechercher les unités amorçables non encore infectées. Le problème principal est de ne pas réinfecter une unité déjà vérolée. Le virus STEALTH utilise pour cela les 30 premiers octets de son propre code : leur présence trahira un infection antérieure et le virus ne fera rien (routine CHECK_DISK).

Le second aspect que le virus doit prendre en charge est le moment favorable pour infecter une unité. Le problème se pose différemment selon la nature de la cible :

- S'il s'agit d'un disque dur, la seule possibilité est que l'utilisateur a booté sur disquette infectée. C'est une éventualité peu fréquente, le plus souvent résultant d'un oubli d'une disquette dans le lecteur, mais cela suffit pour l'infection. Le secteur infecté est chargé et installe le virus dans le secteur de démarrage du disque dur.
- S'il s'agit d'une disquette ou d'un autre support amovible amorçable du même type, l'infection se propage alors à partir du disque dur infecté. L'avantage est que dans cette configuration, de nombreuses unités de ce type vont pouvoir être vérolées. STEALTH agit dans ce cas chaque fois que l'utilisateur effectue une commande se traduisant par une lecture de la disquette. A cette fin, il détourne l'interruption 13H (services disques du BIOS) au démarrage la machine :

```
INT_13H :
                STI                     ; autorisation des interruptions
                                          hardware non masquables.
                CMP AH,2                ; service de lecture demandé ?
                JNZ I13R                ; sinon retourne le contrôle à l'appel
BIOS
                CMP DH,0                ; demande sur tête 0 ?
                JNZ I13R                ; sinon retourne le contrôle à l'appel
BIOS
                CMP CH,0                ; demande sur piste 0 ?
                JNZ I13R                ; sinon retourne le contrôle à l'appel
BIOS
        RF0:    CMP DL,80H              ; demande de lecture sur le disque dur
                JNC I13R                ; sinon retourne le contrôle à l'appel
BIOS
                CMP CL,1                ; lecture du premier secteur ?
                JNZ I13R                ; sinon retourne le contrôle à l'appel
BIOS
                CALL CHECK_DISK         ; appel de la procédure de vérification
                                          d'infection
                JZ  I13R                ; si déjà infectée, retour à l'appel
BIOS
                CALL INFECT_FLOPPY      ; appel procédure infection disquette
        I13R:   JMP DWORD PTR CS:[OLD_13H] ; retour final à l'appel BIOS original
```

où OLD_13H contient l'adresse où le vecteur d'interruption 13H original a été stocké avant d'être remplacé par cette nouvelle procédure.

LA ROUTINE DE COPIE

Une fois la cible identifiée, l'infection consiste à y copier le virus. Cette copie doit se faire de sorte :

- qu'il n'y ait aucune interférence avec le système susceptible de trahir la présence du virus ;
- que le virus soit le plus indétectable possible (furtivité) d'un point de vue statique ;
- que le virus soit le plus portable possible (l'infection doit réussir quel que soit le support cible, en particulier toutes sortes de disquettes).

En fait, tout est dicté par la contrainte de taille sur le secteur de démarrage : 512 octets. Or, STEALTH occupe sept secteurs, donc six doivent être cachés quelle que soit la cible. Le virus peut traiter les disquettes 5"1/4 (360 Ko et 1.2 Mo) et 3"1/2 (720 Ko et 1.4 Mo) (procédures INFECT_360K, INFECT_12M, INFECT_720K et INFECT_14M) et tous les disques durs (procédure INFECT_HARD).

La procédure de copie se décompose alors de la manière suivante :

- Le virus mémorise le secteur de démarrage original (SDO).
- Si l'unité à infecter est une disquette, le virus déclare certains clusters inoccupés (cluster = deux secteurs, en général pour une disquette), sains comme défectueux, et y installe la plus grosse partie de son code (CPV et SDO). Pour cela, il suffit de mettre la valeur FF7H dans la *File Allocation Table* (la FAT attribue une valeur à chaque cluster décrivant selon son état d'occupation : 0, le cluster est libre, un pointeur sur le cluster suivant dans le cas d'un fichier (structure de liste chaînée, FF8H - FFFH si le cluster correspond à la fin du fichier et </ttFF7H si le secteur est défectueux). Les secteurs choisis dépendent du type de disquette. Par exemple, pour une disquette de 1.4 Mo, STEALTH utilise les secteurs 13-17 (pour le corps principal du virus), et 18 pour le SDO, de la piste 79, tête 0. Le choix des pistes est judicieux car il correspond à des clusters utilisés en dernier. Cette approche est subtile car les secteurs défectueux sont ignorés par l'OS, et donc par les logiciels lancés par cet OS (en premier lieu les antivirus). Le virus modifie également la FAT2 (copie de la FAT pour restauration en cas de problème). A noter que cette technique de leurre est applicable à tous les OS.
- Si l'unité est un disque dur, le virus installe CPV et SDO dans la piste 0, tête 0, à partir du secteur 2. Cette piste est indépendante de l'OS et permet d'y cacher avantageusement le virus. Le plus petit des disques durs contient au minimum une bonne dizaine de secteurs.
- Installe le secteur de démarrage infecté (SDV) en tête 0, piste 0 et secteur 1, après l'avoir modifié avec les paramètres de l'unité en cours d'infection (informations sur la table de partition, les paramètres techniques...).

STEALTH : LES MÉCANISMES D'ANTIDÉTECTION

Ce qui fait de STEALTH un virus particulièrement élégant sont ses mécanismes de furtivité. Certes, il existe désormais des techniques beaucoup plus raffinées pour contrer les évolutions de la lutte antivirale, mais celles de STEALTH restent remarquables et valent d'être présentées.

La première des astuces est la lutte contre un éventuel examen du secteur de démarrage par un antivirus. Ce secteur contient la seule partie du virus qui n'est pas cachée dans des clusters défectueux (ce afin de pouvoir accéder et charger CPV et SDO).

- Dans le cas d'une disquette, tout ordre de lecture du secteur (interruption 13H) est intercepté par le virus (qui est alors résident ; voir plus loin). Si la demande de lecture concerne un secteur autre que le secteur de démarrage, le virus redonne le contrôle à l'interruption 13H originale. Dans le cas contraire, le virus redirige la lecture vers le secteur contenant la copie saine du secteur de démarrage (SDO). L'antivirus conclut à la non-infection.
- Dans le cas d'un disque dur, le virus interdit toute lecture ou toute écriture dans les secteurs 2 à 7 de la piste 0, tout en faisant croire que l'ordre demandé s'est effectué sans problème.

Seul un examen extérieur à l'OS (en bootant sur une disquette système saine) permettra de détecter le virus (SDV).

Le virus, étant résident, et après avoir passé le contrôle au secteur de démarrage original, doit veiller à ne pas être détecté. Pour cela, le virus soustrait de la mémoire à l'OS. Lors du démarrage, le BIOS stocke la quantité de mémoire physique disponible à l'adresse 0040H:0013H, en kilo-octets. Cela permet au DOS de savoir quelle quantité de mémoire il peut utiliser (dans la limite des 640 Ko). Le virus STEALTH prenant le pas avant le DOS, il dérobe de la mémoire en soustrayant la quantité qui lui est nécessaire (4 kilo-octets) et s'installe en partie haute de mémoire. Le DOS n'y accédera pas, car pour lui, elle n'existe pas ; le virus agira via le déroutement de l'interruption 13H. En résumé :

- SBV se charge en mémoire haute à l'adresse 9820H:7C00H, puis il lit les 6 autres secteurs (CPV et SDO) et les place en mémoire juste après lui (de 9820H:7000H à 9820H:7BFFH).
- Le SBV soustrait alors 4 kilo-octets à la valeur stockée en 0040H:0013H.
- Le SBV détourne l'interruption 13H vers le virus.
- Enfin, il déplace SBO de 9820H:7A00H vers 0000H:7C00H et l'exécute. Une séquence de démarrage normale s'effectue ensuite.

Cette dernière fonctionnalité de furtivité est particulièrement dangereuse, la partie de la mémoire utilisée n'existant pas pour l'OS. La détection du virus résidant ne peut plus se faire directement.

CONCLUSION

Le virus STEALTH est un virus élaboré aux mécanismes d'antidétection élégants. Ce virus et ses techniques, désormais bien connus des principaux logiciels antivirus, illustrent bien l'ingéniosité d'un programmeur déterminé à les contourner. En conséquence, il ne faut pas sous-estimer le risque des virus de boot, notamment pour des plates-formes non Windows où ces techniques sont aisément transposables. Ces virus sont toujours d'actualité.

Un autre aspect des choses qui doit les faire redouter, est la multiplicité des médias d'amorçage. Si l'utilisation des disquettes, et leur protection physique en écriture, est relativement bien gérée (mais combien de gens, qui n'ont pas d'antivirus, oublient une disquette dans le lecteur lors de l'extinction de la machine), il n'en est pas de même pour les CD-ROM (il serait facile d'infecter les exécutables d'installation et de formatage d'une copie illégale de Windows par exemple) et les ZIP (dont la protection en écriture est logicielle). On peut craindre, si ce n'est déjà le cas, que de nouveaux virus de boot aux capacités de furtivité novatrices n'exploitent ces possibilités quel que soit l'OS considéré, dans le futur.

Alors, comment lutter contre les virus de boot ? D'abord avoir un bon antivirus et le mettre à jour régulièrement. Mais cela ne suffit pas. Il faut également modifier la séquence de démarrage au niveau du BIOS (action réversible en cas de *crash* logique du disque dur) et ne permettre l'amorçage que sur le disque dur. Ceci n'est efficace que si le BIOS est lui-même protégé ensuite par un mot de passe. Or, un audit régulier montre que plus de 85 % des machines des administrations et des entreprises ont une séquence de démarrage inappropriée et dangereuse (ce qui permet notamment de facilement s'introduire sur la machine en cas d'accès physique et d'y installer par exemple le module serveur d'un cheval de Troie, une bombe logique ou autre produit : moins de cinq minutes sont nécessaires et ce sans trace).

Eric Filiol

Ecole Supérieure et d'Application des Transmissions

Laboratoire de cryptologie et de virologie

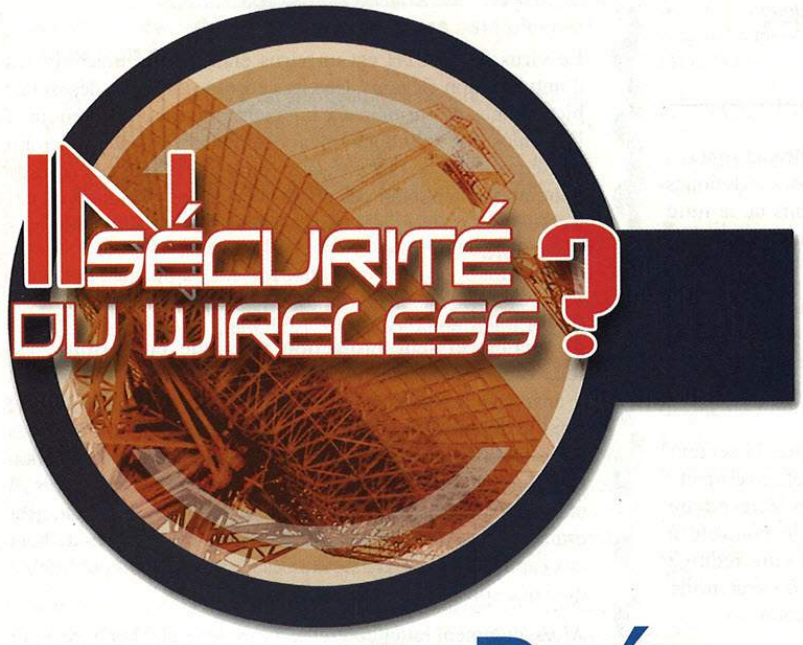
efiliol@esat.terre.defense.gouv.fr

http://www-rocq.inria.fr/codes/Eric.Filiol/index.html

RÉFÉRENCES

[1] Michael Tischer. *La Bible du PC : Programmation Système*, 6e édition, Micro Applications, 1996.

[2] Mark Allen Ludwig. *The little black book of computer viruses*, American Eagle Publications, 1996.



1 Présentation du WiFi (Wireless Fidelity)
2 Principes de la norme 802.11
3 Attaque sur les réseaux 802.11b
4 La sécurité du WEP
5 BlueTooth, Hiperlan : les autres normes Wireless

1 Présentation du WiFi (Wireless Fidelity)

L'informatique est incontestablement un secteur engagé dans une course perpétuelle et effrénée à l'innovation. Cette compétition aux enjeux commerciaux immenses n'est d'ailleurs pas sans conséquences sur la qualité des logiciels et la pérennité des technologies offertes aux utilisateurs finaux (mais ceci est un autre débat). Les réseaux informatiques n'échappent pas à cette règle avec le développement du sans fil.

EMERGENCE DE LA NORME IEEE 802.11

Après avoir connu une course au débit dans les années 90, les services réseau se devaient de suivre la mode de la mobilité initiée par les téléphones portables. On est ainsi passé de l'Ethernet 10 Mb au Gigabit Ethernet qui se popularise de plus en plus, tandis que les connexions modem sont passées du coupleur acoustique 300 bauds (mais qui s'en souvient encore ?) à l'ADSL Megabit et au VDSL.

Au milieu de nombreuses tentatives propriétaires plus ou moins heureuses, une norme s'est aujourd'hui imposée au point de contrôler près de 95% du marché : la norme IEEE 802.11 (ISO/IEC 8802-11). La première norme (publiée en 1997) autorisait un débit jusqu'à 2 Mb/s. L'année suivante, la norme 802.11b (commercialement appelée "WiFi" pour Wireless Fidelity) autorise des débits jusqu'à 11 Mb/s : c'est la norme la plus usitée aujourd'hui.

Aujourd'hui cette norme est disponible gratuitement au téléchargement, profitez-en ! [1]

AVANTAGES

La norme WiFi présente des avantages indéniables sur ses concurrentes, qui ont fait sa popularité auprès des constructeurs et des utilisateurs :

- Norme internationale maintenue par l'IEEE et indépendante d'un constructeur en particulier ;
- Fonctionnement similaire à Ethernet, évitant le redéveloppement de couches réseau spécifiques ;
- Rayon d'action important (jusqu'à 300m en champ libre), permettant de se déplacer à l'intérieur d'un bâtiment sans perte de connexion ;
- Débit acceptable (11 Mb/s, soit un débit effectif de données autour de 6 Mb/s en pointe) ;
- Mise en œuvre extrêmement facile :
  - pas de travaux,
  - pas de déclaration préalable dans la plupart des pays,
  - pas de licence radio à acheter,
  - coût d'une installation inférieure à 150 euros par poste,
  - épargne les travaux de câblage dans les locaux anciens ou vétustes.

LE RESTE DE LA "FAMILLE"

La norme 802.11b n'est pas (comme son "b" l'indique) la seule norme de la famille 802.11 puisque, à l'heure actuelle, les travaux suivants sont plus ou moins aboutis : voir tableau ci-dessous.

DES DÉTAILS GÊNANTS

Les effets pervers de la course à l'innovation des technologies sans fil sont tangibles pour les utilisateurs finaux.

Malgré les efforts de normalisation de l'IEEE, les plus gros constructeurs "améliorent" souvent la norme à leur sauce, espérant ainsi capter des parts de marché : l'extension la plus connue est le "double rate" qui autorise un débit de 22 Mb/s dans la bande des 2,4 GHz, au prix d'une incompatibilité des matériels avec ceux d'autres constructeurs.

Mais le point le plus critique concerne les sécurités intégrées au protocole standard, qui sont irrémédiablement et profondément défaillantes, comme il sera largement expliqué dans les pages à venir.

Norme	Titre	Contenu
802.11a (WiFi 5)		La norme du futur proche : 5 Ghz, 54 Mb/s
802.11b (WiFi)		La norme actuelle : 2,4 Ghz, 11 Mb/s
802.11c	Bridge Operations Procedures	
802.11d	Global Harmonization	Adresse les problèmes légaux
802.11e	MAC Enhancements for QoS	
802.11f	Inter Access Point Protocol	Améliore la qualité de service pour les utilisateurs itinérants
802.11g (DRAFT)	Physical Layer Update	Changement de modulation "backward-compatible" qui autorise un débit de 54 Mb/s sur du 802.11b Du matériel compatible est déjà disponible sur le marché.
802.11h (DRAFT)	Spectrum Managed 802.11a	Document dédié aux problèmes légaux européens liés à l'utilisation de la bande des 5 GHz
802.11i (DRAFT)	MAC Enhancements for Enhanced Security	Un remplacement progressif de la "sécurité" WEP. La V1 est compatible avec l'existant et repose sur le protocole TKIP. La V2 intègre 802.1x et AES : elle est incompatible avec le parc existant.

LE FONCTIONNEMENT EN DÉTAIL

PRINCIPES RADIOÉLECTRIQUES

Le principe de fonctionnement radioélectrique du WiFi est basé sur une technologie militaire, dont le principal objectif est de limiter la détectabilité des stations en émission. Ainsi, l'émission du signal s'effectue sous le niveau de bruit et nécessite la connaissance de l'algorithme de modulation pour être détecté. Cet argument était présenté (y compris sur le site de Cisco) comme une sécurité pour les communications WiFi. Malheureusement, n'importe quelle carte du marché étant compatible avec la norme, elle peut détecter les communications avoisinantes et se comporter en "sniffeur" ! Présenter les communications point à point comme indétectables est donc une grave erreur de sécurité (mais nous verrons que ça n'est pas la seule).

Canal	Fréquence (GHz)
1	2,412
2	2,417
3	2,422
4	2,427
5	2,432
6	2,437
7	2,442
8	2,447
9	2,452
10	2,457
11	2,462
12	2,467
13	2,472
14	2,484

La première norme (publiée en 1997) autorise les modulations DSSS[1] et FHSS[2] avec des débits allant jusqu'à 2 Mb/s. La version 1998 de la norme autorise des débits jusqu'à 11 Mb/s et se limite à la modulation DSSS.

Les bandes de fréquences allouées sont divisées en 14 canaux séparés de 5 MHz, ce qui donne la table ci-contre.

Bien entendu, les fréquences réellement libérées par les militaires et utilisables varient d'un pays à l'autre, comme nous le verrons par la suite. Compte tenu du débit de 11 Mb/s (et en vertu du théorème de Shannon diront les plus cultivés), la largeur du spectre d'émission peut atteindre 22 MHz. Pour limiter les interférences ("overlapping") dommageables à la bande passante, une installation multi-bornes judicieuse repose donc sur les canaux 1, 6 et 11 distants de 25 MHz.

Pour conclure sur le sujet de la radioélectricité (très intéressant par ailleurs mais malheureusement trop ardu pour une brève introduction telle que celle-ci), une émission de fréquence 2,4 GHz a une longueur d'onde de l'ordre de 12,5 cm et se propage quasiment en ligne droite, rebondissant sur les obstacles électriquement adéquats (telles que des gaines de ventilation), ce qui permet parfois de les capter en des endroits inattendus…

[1] **DSSS** : "Direct-Sequence Spread Spectrum"
[2] **FHSS** : "Frequency-Hopping Spread Spectrum"
[3] **CSMA/CA** : Carrier Sense Multiple Access / Collision Avoidance
[4] **CSMA/CD** : Carrier Sense Multiple Access / Collision Detection
[5] **Autorité de Régulation des Télécommunications**

COMMUNICATIONS BAS NIVEAU

De manière synthétique, la norme 802.11 est fortement similaire à la norme 802.3 (Ethernet) pour tout ce qui concerne les couches réseau !

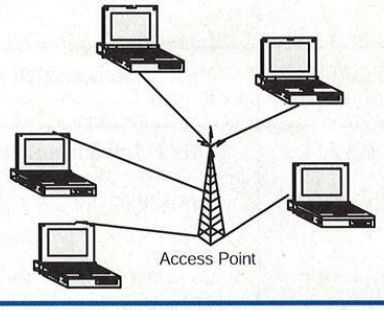
Les seules différences sont les suivantes (sans trop rentrer dans les détails techniques qui seront couverts par la suite) :

- Le 802.11 évite les conflits de transport (technique CSMA/CA[3]) là où le 802.3 détecte et corrige les conflits de transport (technique CSMA/CD[4]) ;
- Les adresses MAC des passerelles sont ajoutées en entête de chaque trame ;
- La norme intègre en standard le support de l'itinérance utilisateur ("roaming") ;
- La norme définit de nouvelles trames de gestion protocolaires dont les plus importantes sont les trames balises ("beacons") – qui annoncent tous les 1/10ème de seconde la présence du point d'accès – et les trames d'association – qui permettent la négociation des paramètres de connexion entre la carte et le point d'accès ;
- *Last but not least*, le protocole prévoit des mécanismes de sécurité (facultatifs) tels que le chiffrement RC4 des communications.

A propos de cartes et de points d'accès, il faut bien comprendre que le WiFi supporte deux modes de fonctionnement :

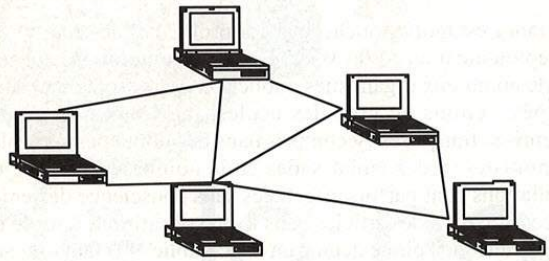
- Les connexions point à point entre N cartes (mode Ad Hoc), formant un réseau maillé ;
- Les connexions via un point d'accès qui fait office de "hub" (mode Infrastructure) – cette topologie en étoile est la plus couramment rencontrée.

1



Réseau en étoile

2



Réseau maillé

LE POINT SUR LA RÉGLEMENTATION

Les bandes de fréquence allouées au WiFi sont actuellement utilisées par certains types de radars militaires, dont la durée de vie est encore d'une dizaine d'années. Il est dès lors tout à fait compréhensible que le retour à la vie civile de ces fréquences s'effectue parfois dans la douleur…

Aux **Etats-Unis**, seuls les canaux 1 à 11 sont ouverts au public. Au **Japon**, la situation est encore plus restrictive puisque seul le canal 14 est disponible ! En **Europe**, l'ensemble des canaux sont en général utilisables librement, sauf en **France** qui fait encore une fois figure de mauvais élève. En effet, notre ART[5] [2] nationale impose des conditions draconiennes aux émissions dans la bande des 2,4 GHZ :

- Seules les émissions confinées à l'intérieur de locaux privés (on se demande comment) et de puissance inférieure à 100 mW sont autorisées librement.
- Si l'émetteur est accessible depuis un lieu à ciel ouvert, seuls les canaux 10 à 14 sont utilisables (dommage pour les propriétaires de matériel configuré par défaut sur le canal 1 ou le canal 6, et ils sont nombreux). La puissance d'émission ne doit pas dépasser 10 mW. Toutefois, des dérogations (délivrées par le Ministère de la Défense !) sont possibles jusqu'à 100 mW et sur un domaine privé uniquement.

A ce sujet, il faut savoir que l'ART parle de PIRE (Puissance Isotrope Rayonnée Equivalente), qui est largement fonction de l'antenne utilisée. Ainsi, 30 mW de puissance d'émission peuvent correspondre à 100 mW PIRE avec une bonne antenne. Il faut aussi savoir qu'aux Etats-Unis la puissance PIRE autorisée est de 1000 mW !

En résumé, la situation législative a longtemps été telle que décrite dans le tableau ci-dessous.

Depuis le 7 novembre 2002, l'ART a, d'une certaine manière, assoupli sa position : les opérateurs de télécoms et les fournisseurs de service seront autorisés à mettre en service des points d'accès dans les lieux de passage (gares, aéroports, hôtels, etc.) sans déclaration ni autorisation préalable. Les associations et les collectivités locales pourront effectuer des expérimentations entre le 1er janvier 2003 et le 1er janvier 2006 uniquement, à condition d'avoir déposé un dossier entre novembre et décembre 2002 ! A la date de rédaction de cet article ces expérimentations n'étaient possibles que dans quelques dizaines de départements "libérés" par le Ministère de la Défense, toutefois sous la pression populaire la législation ne cesse d'évoluer (rendez-vous sur le site de l'ART pour plus de renseignements).

On le voit, l'ART s'oriente clairement vers une exploitation exclusivement commerciale du WiFi. Il faut dire que les intervenants invités au débat préparatoire à l'Assemblée Nationale s'appelaient Orange, Cisco et Sagem !

INITIATIVES PRIVÉES

L'un des grands espoirs du WiFi, qui mobilise les efforts de nombreuses associations et communautés locales, c'est l'accès Internet Haut Débit dans des régions inaccessibles par l'ADSL (réseaux dits "RLAN").

Malheureusement, la réglementation imposée par l'ART ne favorise pas de telles initiatives, et bien que les opérateurs associatifs ou privés tentés par l'aventure puisse déposer un dossier à l'ART depuis le 12 novembre dernier, les projets seront soumis à approbation du Ministère de la Défense et gratuits pendant 18 mois uniquement.

En avance de phase sur le contexte législatif, de nombreux projets associatifs se développent en France :

- **France-Wireless** [3] dont l'objectif est de déployer un réseau associatif national 802.11b/g.
- **WiFi-France** *alias* Paris sans-fil *alias* WiFi-Paris [4], dont le site plus technique ravira les débutants.
- **WiFi-*** où * = {Strasbourg, Metz, Montauban, Grenoble, Rennes, Nantes…} qui reflètent les initiatives locales dans de nombreuses villes de province.

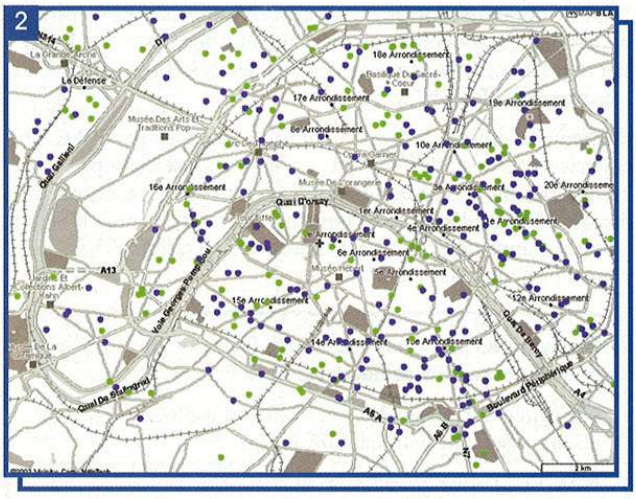
Fréquences	Puissance en intérieur	Puissance en extérieur
1-14	PIRE < 10 mW	PIRE < 2,5 mW
8-14	PIRE < 100 mW	Sur propriété privée uniquement et sous réserve d'autorisation PIRE < 100 mW

Aujourd'hui, ces associations restent le meilleur moyen de se former et de se procurer du matériel, qui demeure relativement inabordable dans les magasins de détail français (mieux vaut commander aux Etats-Unis) ! Toutefois, ces associations souffrent parfois d'un manque d'ambition : pour avoir assisté personnellement à une réunion de WiFi-Paris, il est amusant de voir que s'y côtoient enseignants, anarchistes libertaires, passionnés d'informatique, et que tous ne s'accordent pas sur l'avenir du WiFi.

Il faut dire que les initiatives associatives françaises sont limitées par le cadre législatif et les moyens financiers de leurs membres. A Londres, par exemple, le réseau associatif beaucoup plus développé a été reconnu d'utilité publique.

Côté commercial, rien de très "grand public" pour le moment, mais des projets en préparation du côté de chez Orange paraît-il. On saluera aussi Wifix [10], dont l'objectif est de mettre en relation des fournisseurs d'accès Internet sans-fil avec des clients intéressés par le nomadisme (le tout moyennant finances). La plus médiatisée des initiatives reste le Colombus Café à Neuilly qui offre gratuitement l'accès Internet WiFi à ses clients. Des études sont en cours pour le "câblage" des aéroports, tandis que les grands hôtels disposent souvent déjà d'une infrastructure opérationnelle.

2



La carte des adhérents de WiFi-Paris

C'est aux Etats-Unis que sont nées les initiatives pionnières, telles que Seattle Wireless [5]. Il est impressionnant de constater là-bas combien cette technologie s'est banalisée : hôtels (y compris les procédures d'enregistrement), aéroports [6], conférences...

Toutefois, à l'euphorie du début succèdent parfois des revirements brutaux liés aux problèmes de sécurité, qui ont été dans certains cas largement sous-estimés.

PRATIQUES CONSTATÉES

La France est moins touchée par ces problèmes de sécurité suite au déploiement tardif du WiFi. Force est toutefois de constater que de nombreux organismes publics ou privés sont aujourd'hui équipés : citons en vrac des écoles, mais aussi de grandes entreprises françaises (y compris dans des domaines sensibles), des ministères, des ambassades et de nombreuses PME. Ces installations sont parfois effectuées sans conscience des enjeux de sécurité, d'où des articles dans la presse satirique sous le titre peu glorieux de "piraté depuis un banc public" ! Il faut dire aussi que de nombreux portables sont aujourd'hui équipés de cartes WiFi intégrées et supportées nativement par Windows XP.

Chez les particuliers, le WiFi rencontre un certain succès (en particulier le modèle AirPort qui domine largement le marché français). Parmi les derniers modems NOOS, certains intègrent d'ailleurs une option WiFi. Malheureusement, en termes de sécurité, la situation est encore plus catastrophique : plus des 2/3 des utilisateurs ne mettent pas de clé WEP, laissent les mots de passe par défaut (qui apparaissent dans le nom de la borne). La palme revient au routeur NOOS, qui configure automatiquement l'ensemble du réseau local sur tout poste à proximité grâce à la fonction UPNP de Windows XP !

Mais cela fait l'objet d'un autre article…

Nicolas RUFF

nicolas.ruff@edelweb.fr
Consultant Sécurité Windows & Réseaux

Laurent DUPUY

laurent.dupuy@freesec.com
Consultant Indépendant en Sécurité

RÉFÉRENCES

[1] Les standards IEEE 802
http://standards.ieee.org/getieee802/802.11.html

[2] Le site de l'ART **http://www.art-telecom.fr/**

[3] France Wireless **http://wireless-fr.org/**

[4] Paris sans Fil **http://www.paris-sansfil.net/**

[5] Seattle Wireless **http://www.seattlewireless.net/**

[6] Wireless Airports Association
http://www.wirelessairport.org/

[7] Dossier "Vivre le Net" sur les réseaux sans fil
http://www.vivrele.net/dossiers/hautdebit/rlan_wifi

[8] WiFi Alliance **http://www.weca.net/OpenSection/index.asp**

[9] "Les nouveaux passe-murailles"
Le Point, en date du 19/07/2002

[10] Wifix **http://www.wifix.com/**

2 Principes de la norme 802.11

Les standards 802.11 de l'IEEE définissent le mode de fonctionnement de réseaux de données sur un support radio. Cet article présente brièvement les caractéristiques techniques de ces réseaux, en s'intéressant plus spécifiquement aux trois standards existants ou émergents que sont le 802.11a, le 802.11b, et le 802.11g.

CARACTÉRISTIQUES PHYSIQUES

Le 802.11b est aujourd'hui disponible mondialement, contrairement au 802.11a, principalement implanté aux États-Unis et commençant seulement à se répandre en Europe. Le premier utilise la bande de fréquences comprise entre 2.4 et 2.483 GHz, séparée en 14 canaux de 22 MHz. Le lecteur aura immédiatement remarqué qu'en pratique, seuls trois de ces canaux sont complètement disjoints, les canaux 1, 6 et 11. En France, seuls 13 canaux sur 14 sont autorisés, contre seulement 11 aux États-Unis. Le 802.11a, quant à lui, utilise plusieurs plages de fréquences séparées dans la bande des 5 GHz. Enfin, le 802.11g est une extension du 802.11b pour permettre de plus hauts débits dans la bande des 2.4 GHz.

En termes de vitesse, le 802.11b est le moins performant de tous, offrant quatre différents débits théoriques, respectivement 1, 2, 5.5 et 11 Mbps. Une version propriétaire développée par TI, le 802.11b+, est compatible avec les cartes et points d'accès (*Access Point* ou AP) 802.11b, mais propose des débits de 22 Mbps quand tous les équipements supportent cette norme. Le 802.11a, quant à lui, propose 6, 9, 12, 18, 24, 36, 48 et 54 Mbps. L'un comme l'autre permettent d'agréger plusieurs canaux sur une même zone en utilisant plusieurs points d'accès. Ainsi, il est possible de monter le 802.11b jusqu'à un débit théorique de 33 Mbps. Enfin, le 802.11g est compatible avec le 802.11b, proposant des débits de 5.5 et 11 Mbps, mais il permet aussi de travailler à 54 Mbps en utilisant les mêmes techniques que le 802.11a. Il existe également des technologies propriétaires (développées par Texas Instruments et Intersil) permettant d'utiliser le 802.11g à des fréquences intermédiaires.

	Débits (Mbps)	Bande de fréquence	Disponibilité (France)
802.11a	6, 9, 12, 18, 24, 26, 48, 54	5 GHz	A l'étude
802.11b	1, 2, 5.5, 11	2.4 GHz	Oui
802.11b+	1, 2, 5.5, 11, 22 (TI)	2.4 GHz	Oui
802.11g	5.5, 11, 33 (Intersil), 6-54 (TI), 54	2.4 GHz	Non

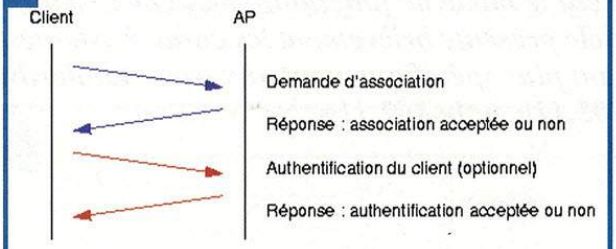
MODES DE COMMUNICATION

Dans le monde merveilleux du sans-fil, les deux types d'éléments pouvant communiquer sont les clients (ordinateurs plus ou moins portables, "désorganiseurs" personnels, etc.) et les points d'accès. Ces derniers sont en pratique des bornes qui s'annoncent à leur entourage et peuvent être utilisées comme relais par les clients. Les points d'accès peuvent également servir de pont vers un réseau filaire classique.

LE MODE INFRASTRUCTURE

Le mode le plus utilisé est souvent appelé *infrastructure*. Dans ce mode, un client va s'*associer* à un point d'accès pour pouvoir l'utiliser, le processus d'association pouvant ou non inclure une authentification. Une fois cette opération effectuée, le client peut donc communiquer avec toutes les autres stations associées au même point d'accès, et si l'AP est relié à un réseau filaire, l'utiliser comme pont vers ce réseau.

1



Exemple d'association

LE MODE AD HOC

Quand l'interface réseau sans fil d'un client est dans le mode *ad hoc*, il peut directement établir un réseau point-à-point avec un autre client équipé d'une configuration similaire. Il est tout à fait possible pour un client donné d'avoir plusieurs liens ad hoc différents simultanément.

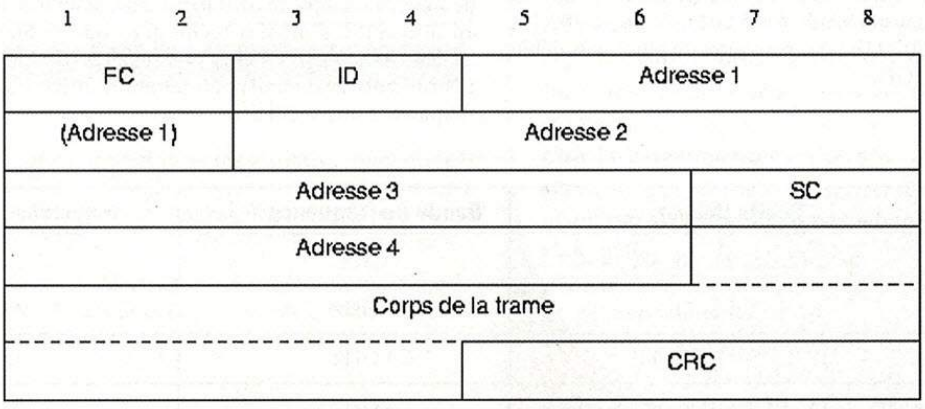
FORMAT DES TRAMES 802.11

De manière générale, une trame 802.11 se présente sous le format suivant (en partant de la couche MAC, nous ne parlerons pas ici de la couche physique sous-jacente, qui par ailleurs varie entre les différentes déclinaisons du 802.11) :voir figure 2.

Les champs correspondent respectivement à :

- **FC**, pour *Frame Control* : version du protocole et type de trame (gestion, données ou contrôle).
- **ID**, pour *Duration/ID* : valeurs utilisées pour l'envoi de certains messages et le calcul du *Network Allocation Vector*.
- **Adresses :** en fonction du champ "frame control", jusqu'à quatre adresses peuvent être utilisées dans une trame. Chacune de ces adresses peut correspondre à l'un de cinq types différents : adresse source, adresse destination, adresse de l'émetteur, adresse du récepteur, et adresse du point d'accès.
- **SC**, pour *Sequence Control* : indique un numéro de fragment et un numéro de séquence, permettant de réordonner des fragments et de repérer les paquets dupliqués.
- Le **corps de la trame** contient les données de niveau supérieur, par exemple un paquet IP.
- **CRC** (*Cyclic Redundancy Check*) : un simple *checksum* permettant de vérifier l'intégrité du paquet.

2



Trame MAC 802.11

3

1	2	3	4	5	6	7	8
Protocole		Type		Sous-type			
To DS	From DS	More frag	Retry	Pw Mgt	More data	Wep	Ordre

Frame control 802.11

Il est également intéressant de décomposer le champ FC bit à bit (et non plus octet par octet comme dans le schéma précédent) : voir figure **3**.

- **Protocole :** la version du standard 802.11.
- **Type :** gestion, contrôle ou données.

Le type *gestion* contient par exemple les demandes d'association et les messages d'annonce d'un point d'accès.

Le type *contrôle* est utilisé pour l'accès au média, avec des messages comme les demandes d'autorisation d'émission (utilisés seulement dans certains cas spécifiques).

Le type *données* concerne les communications normales.

- **Sous-type :** dépend du type.
- **To DS**, pour *To Distribution System* : positionné à 1 si le message est à destination du système de distribution, 0 sinon.
- **From DS** : positionné à 1 si la trame provient du système de distribution.
- **More frag :** positionné à 1 s'il reste des fragments appartenant à la même trame après le fragment courant.
- **Retry :** positionné à 1 s'il s'agit d'une réémission du fragment courant.
- **Pw Mgt** : indique le mode de gestion de l'énergie que la station utilisera après avoir transmis le fragment courant.
- **More data :** indique que la station a encore un certain nombre de trames dans son tampon.
- **WEP :** indique ou non l'utilisation de l'algorithme de chiffrement WEP (voir plus loin).
- **Ordre :** indique si la trame est émise en utilisant une classe spéciale.

CARACTÉRISTIQUES LOGIQUES

Un réseau sans fil 802.11 est caractérisé par un certain nombre d'éléments. Parmi ceux-ci, on retrouve en particulier le canal utilisé, qui va définir précisément la bande de fréquence utilisée. Dans la mesure où la grande majorité des cartes réseaux 802.11 sont capables de scanner l'ensemble des canaux pour découvrir un éventuel point d'accès, le choix du canal n'a que peu d'impact pratique sur le déploiement du réseau (à un détail près, seuls certains canaux étant autorisés pour une utilisation en extérieur en France).

Conjointement avec le canal, c'est le *Service Set Identifier*, ou SSID, qui va définir le réseau en lui-même. On parle parfois également de *Basic* SSID, pour marquer la différence avec l'*Extended Service Set Identifier*, ou ESSID. Il s'agit simplement d'un identifiant déclaré sur le point d'accès, et que les clients doivent fournir pour s'associer. On parle de SSID (ou BSSID) quand il s'agit d'un point d'accès isolé, et d'ESSID quand un ensemble de points d'accès sont configurés avec le même SSID pour offrir une couverture plus étendue à un même réseau, un utilisateur pouvant alors se déplacer d'une cellule à l'autre sans perte de connectivité. Par défaut, un point d'accès s'annonce régulièrement à son entourage par l'émission d'annonces, ou *beacon frames*. Cela dit, il est tout à fait possible d'interdire à un AP de s'annoncer ainsi.

Il a déjà été dit qu'une authentification du client pouvait être requise avant de permettre son association à un point d'accès.

Les différents modes d'authentification sont les suivants :

- **ouvert :** pas d'authentification requise ;
- **fermé :** le client doit connaître le SSID ;
- **WEP :** le client doit connaître le SSID et un secret partagé (voir page suivante).

3 sur les

OU TOUT CE QUE VOUS

Aah, la magie du sans fil. Le plaisir d'arriver dans une salle de conférence, de s'installer avec son portable sur les genoux et d'être déjà relié au vaste monde sans avoir le moindre petit câble à brancher. Terminés, les câbles réseaux dans tous les sens pour les LAN-parties. Finis, ces fils disgracieux au milieu de votre cabinet médical. L'heure est à la légèreté et à la simplicité, les réseaux s'affranchissent de leurs traditionnels supports cuivrés.

Malheureusement, en l'état actuel des choses, l'utilisation d'ondes radio comme support pour des réseaux informatiques est sensiblement équivalente à l'affichage d'un panneau publicitaire géant invitant tout un chacun à venir s'y connecter. Pire encore, les rares mesures de sécurité présentes dans ces technologies sont à peu près aussi efficaces qu'une clôture en bois qui tenterait d'arrêter un char.

MAMAN, MAMAN, REGARDE, Y'A NOS PAQUETS À LA RADIO !

Tout comme sur un réseau filaire, il est possible sur un réseau sans fil (ou *wireless*) d'observer passivement le trafic, ce que l'on appelle plus communément *sniffer*. Sur un réseau filaire, Ethernet par exemple, il suffit d'avoir accès à un port d'un quelconque *hub* ou *switch* connecté au réseau local pour pouvoir tranquillement jouer les voyeurs (moyennant l'investissement

WEP : WIRED EQUIVALENT PRIVACY

Un problème évident inhérent aux réseaux sans fil s'impose à nous : comment garantir la confidentialité d'informations circulant sur un support radio, que n'importe qui peut capter ? Pour ce faire, l'utilisation d'un protocole cryptographique semble être le choix le plus logique. Le WEP, pour *Wired Equivalent Privacy* (et non *Wireless Encryption Protocol*, comme de nombreuses âmes simples en sont convaincues), est un protocole dédié au chiffrement des trames 802.11. Son utilisation est relativement simple, puisqu'il suffit de déclarer des clés de session WEP sur le point d'accès et sur chaque client, et de leur confirmer que les communications doivent être chiffrées. Le lecteur avisé aura remarqué que ces clés sont donc statiques, et que toute modification de la configuration s'avèrera plutôt lourde.

En pratique, le WEP s'appuie sur l'algorithme de chiffrement (symétrique) RC4, avec des clés d'une longueur de 64 ou 128 bits. Cet algorithme génère un flux de données, qui sera superposé comme un masque aux données émises sur une interface réseau sans fil (l'opération réalisée est un XOR, c'est-à-dire un *ou exclusif* bit à bit). Pour chaque paquet, la clé employée utilise 24 bits de vecteur d'initialisation (ou plus simplement IV), laissant 40 ou 104 bits pour la partie statique de la clé. C'est cette partie statique, de 5 ou 13 octets, qui est déclarée sur les clients et le point d'accès. Comme d'autres articles de ce dossier l'explicitent, cette solution ne présente pas un niveau de sécurité satisfaisant.

Il existe également des technologies d'authentification d'un client auprès du point d'accès qui permettent l'utilisation de clés WEP dynamiques, beaucoup plus sûres. En revanche, ces technologies (notamment l'authentification 802.1x) demandent une infrastructure plus complexe, puisqu'elles imposent généralement l'utilisation d'un serveur d'authentification externe au point d'accès, ainsi que l'installation de clients spécifiques sur chaque client.

ET LA SÉCURITÉ DANS TOUT ÇA ?

L'objectif de cet article n'étant que de donner un aperçu de la technologie 802.11, l'angle de la sécurité n'y a pas été abordé. Le lecteur est cordialement invité à lire la suite du dossier pour cela ;-)

Daniel Polombo
bozo@miscmag.com

Attaques réseaux 802.11b

POUVEZ FAIRE DES ONDES QUI SE BALADENT ALENTOUR

dans une paire de jumelles adéquates dans le cas d'un switch). En règle générale, cela impose à un éventuel attaquant d'obtenir un accès physique aux locaux et un peu d'intimité pour établir une relation de confiance avec la prise réseau qu'il aura repérée.

Sur un réseau sans fil, c'est encore plus simple. Pas besoin de trouver une prise, ni même de se faire passer pour le livreur de pizzas pour pénétrer physiquement sur le site concerné. Les ondes radios, ayant un esprit libertaire assez développé, sont relativement délicates à confiner dans une enceinte fermée. Il suffit à notre méchant pirate de garer sa vieille Panda Fire noire (avec les pare-chocs violets) à proximité de l'immeuble dans lequel est déployé le réseau sans fil, et pour peu qu'il ait une carte réseau d'une puissance suffisante, il pourra accéder sans effort au trafic interne de ce réseau.

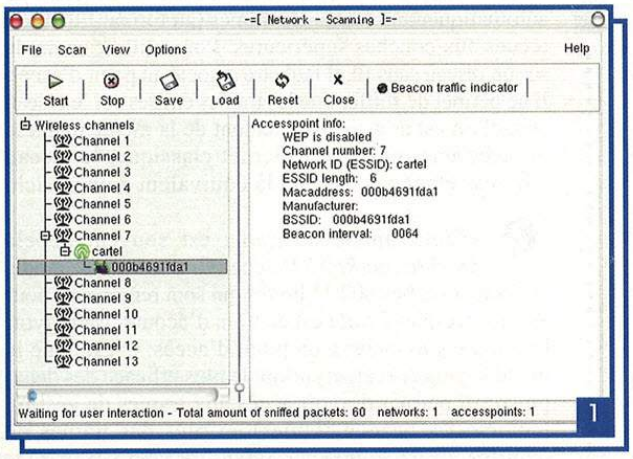
VOYEURISME ORGANISÉ

Notre objectif n'est pas ici de parler des différents outils qui permettent de sniffer le trafic réseau. Les *Tcpdump* et autres *Ethereal* **[1]** sont des outils suffisamment courants et documentés pour qu'une énième présentation ne s'impose pas dans cet article. Nous allons en revanche nous intéresser de plus près à divers autres outils d'analyse de trafic dédiés aux environnements wireless.

Cependant, avant cela, il convient d'avoir une idée un peu plus précise de la façon dont un client peut se connecter à un point d'accès wireless. Dans la mesure où la technologie la plus répandue est de très loin le 802.11b **[2]**, tous nos exemples s'appuieront sur ce support, qui, rappelons-le, travaille dans la bande des 2,4GHz. Le 802.11b supporte deux modes : le mode *infrastructure* (BSS) pour lequel chaque client se connecte à un point d'accès (AP pour *Access Point*), constituant un réseau centralisé, ou le mode *ad hoc* (ou distribué, IBSS) où chaque client établit un lien point à point avec un autre client, constituant ainsi un réseau maillé. Dans la suite de l'article, nous discuterons du mode infrastructure puisque c'est le mode de fonctionnement par défaut des installations 802.11, et de fait, la mise en œuvre la plus couramment constatée. Du point de vue de la sécurité, les mécanismes utilisés sont les mêmes, entre un client et un AP, ou entre deux clients. Notre discours est donc tout à fait applicable aux deux modes de fonctionnements.

Pour pouvoir utiliser un point d'accès wireless, un client doit s'y *associer*. Pour ce faire, il lui faut simplement connaître le numéro de canal utilisé par le point d'accès d'une part, et d'autre part le SSID (ou ESSID). Précisons également à l'intention du lecteur averti que le cas des mesures cryptographiques sera étudié plus loin dans cet article.

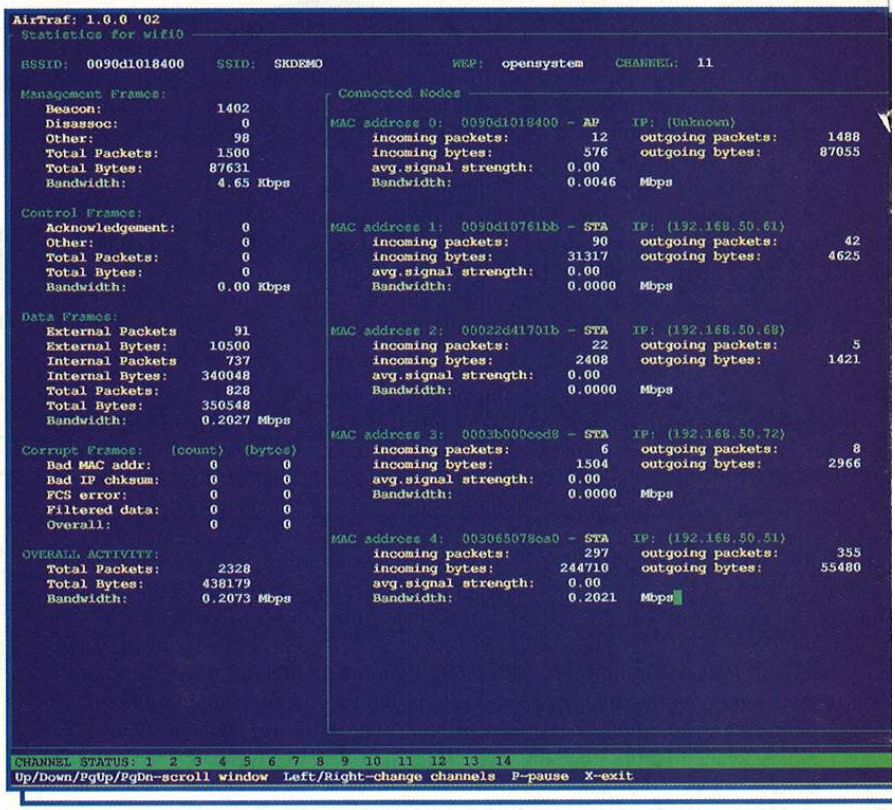
Pour ce qui est de trouver un canal sur lequel écouterait un hypothétique point d'accès, la plupart des cartes wireless du marché le font déjà lorsqu'elles cherchent à s'associer. Certaines, comme les Aironet de Cisco, permettent d'écouter simultanément sur tous les canaux, d'autres se contentant de sauter d'un canal à l'autre en rapide succession. Le point d'accès émettant régulièrement des trames spécifiques pour s'annoncer (nommées *beacon frames* par nos amis anglophones), il est aisé de déceler leur présence. Une fois l'animal repéré, il suffit de se tapir dans les fourrés quelques instants et d'observer le trafic avec un des



Wellenreiter

sniffers susmentionnés pour obtenir le SSID qu'il utilise, ce dernier étant en effet transmis en clair avec chaque paquet. Pour ce faire, prenons par exemple l'un des nombreux scanners de réseaux sans fil existant, *Wellenreiter* (sous Linux, fig. **1**) ou *Aitraf* (également sous Linux, en console). Ce dernier permet une analyse très complète, fournissant des statistiques très détaillées pour chaque point d'accès détecté comme l'analyse du trafic 802.11b, les stations associées, voire même l'analyse du trafic réseau si celui-ci circule en clair (cf fig. **2**).

Comme on peut le voir, celui-ci nous fournit immédiatement toutes les informations nécessaires, et même un peu de superflu. Connaissant le SSID, nous pouvons maintenant nous associer à ce point d'accès et l'utiliser pour communiquer. Notons au passage que l'association n'est absolument pas nécessaire pour l'écoute passive du réseau – le lecteur avisé aura d'ailleurs remarqué que s'il fallait s'associer pour pouvoir sniffer le réseau, la situation se serait quelque peu compliquée. Il est aussi important de remarquer que si certains AP vous proposent comme fonctionnalité de sécurité de cacher leur SSID dans leurs trames d'annonce, ce masquage est complètement inutile puisque ce SSID sera tout de même présent dans les trames d'association et de communication émise par les clients.



Airtraf, statistiques pour un AP

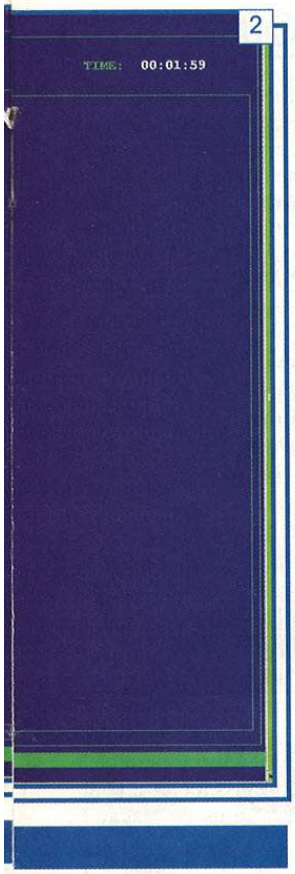
Il y a deux façons d'écouter un réseau sans fil.

1 La première est en tout point identique à l'écoute d'un réseau filaire, c'est-à-dire qu'on passe l'interface en mode *promiscuous*, de sorte qu'elle remonte automatiquement toutes les trames (au format Ethernet) reçues aux couches supérieures. Pour utiliser ce mode sur un réseau sans fil, il faut être associé au point d'accès. Il ne permet de sniffer que les trames émises sur le réseau auquel on est associé, exactement de la même manière qu'avec une interface Ethernet classique, le réseau wireless étant à ce niveau là équivalent à un switch.

2 L'autre mode utilisable est souvent appelé *monitor*, ou *RFMON*, chez Cisco. Dans ce mode, ce sont les trames 802.11 brutes qui sont remontées, pour les canaux que la carte est en train d'écouter, sans avoir besoin de s'associer à un point d'accès. C'est donc le mode le plus discret, et surtout le plus efficace des deux, puisqu'il permet d'avoir accès aux trames de gestion (comme les beacon frames) en plus des trames de données. C'est le mode qu'on utilise pour scanner les réseaux wireless.

Le premier problème et le plus évident est la non-confidentialité des informations qui transitent. En soi, c'est déjà assez grave, d'autant plus que la plupart des utilisateurs de réseaux sans fils sont inconscients des risques qui les menacent, et donc n'hésitent pas à faire transiter des informations sensibles sur ces supports. Un exemple qui a frappé les esprits en 2002 est celui de la société américaine Best Buy (cf VulnDev [3]), qui utilise entre autres des terminaux sans fil pour les transactions bancaires dans ses quelques centaines de magasins, et n'avait pas activé le moindre système cryptographique. Sam le pirate pouvait donc s'installer tranquillement sur le parking du magasin et obtenir diverses informations nominatives, et occasionnellement un numéro de carte bancaire.

Plus grave encore, une fois associé, notre malfaiteur peut accéder aux mêmes ressources que n'importe quel client légitime du réseau sans fil, et très vraisemblablement à un certain nombre de ressources sur le réseau câblé qui existe certainement derrière : un point d'accès ne se contente pas de mettre en relation des clients wireless, il peut également servir de pont entre un réseau sans fil et un réseau filaire. Un certain nombre d'attaques seront présentées dans la suite de cet article, mais avant cela, il convient de s'intéresser au principal mécanisme de sécurité utilisé dans les réseaux 802.11b.



2

SANS FIL MAIS PAS SANS FILET ?

La problématique de sécurité liée aux réseaux sans fil n'est pas nouvelle, et il y a longtemps qu'une réponse a été apportée, visant notamment les problèmes de confidentialité. Malheureusement, cette solution, qui répond au doux nom de *WEP* (signifiant *Wired Equivalent Privacy*), présente une faille majeure.

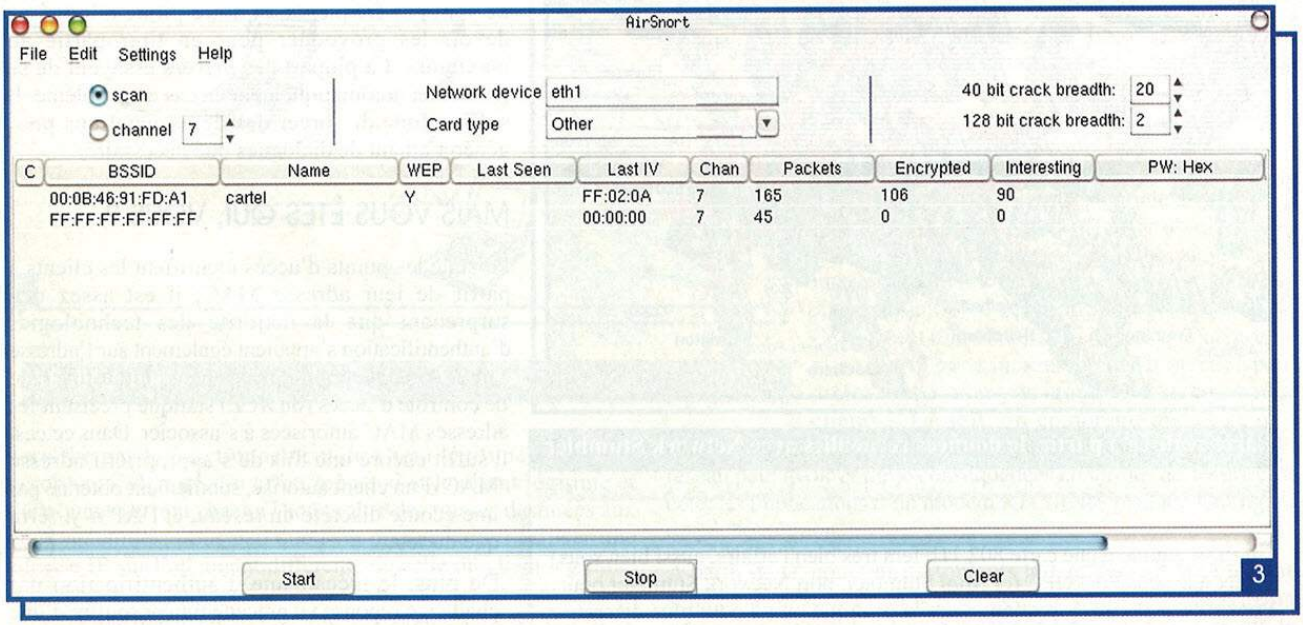
En effet, ce protocole, qui permet de chiffrer intégralement les données transmises entre des équipements 802.11b, utilise une clé de session commune au client et au point d'accès. La connaissance de cette clé suffit donc à un client pour communiquer avec le point d'accès (si l'on y ajoute le canal et le SSID, qui comme il l'a été indiqué précédemment s'obtiennent plus qu'aisément). Or, pendant une communication protégée par WEP, certaines fuites d'information peuvent permettre de déduire cette clé de session. Pour une analyse cryptographique de cette faille, le lecteur se reportera à l'excellent article d'Éric Filiol et Frédéric Raynal dans ce même numéro **[4]**. Nous nous contenterons ici de retenir qu'il existe des outils permettant d'observer passivement le trafic, et potentiellement d'en déduire la clé au bout d'un certain temps, comme *Airsnort*.

Comme l'indique la figure **3**, Airsnort écoute le trafic réseau à la recherche de paquets dits "intéressants", qui peuvent permettre la déduction d'une partie de la clé.

Une autre piste pour casser le WEP repose sur la phase d'authentification, qui repose sur un challenge permettant à l'AP de vérifier que le client est bien en possession de la clé WEP en vigueur. Pour cela, il génère une chaîne aléatoire qu'il envoie au client. Ce dernier la chiffre avec la clé et retourne le résultat, qui sera utilisé pour valider ou non l'authentification. La faille est claire : si nous suivons toutes les phases d'authentification, nous obtenons d'un côté des challenges et de l'autre ces mêmes challenges chiffrés. Nous pouvons donc mettre en œuvre une attaque en *texte clair* (ou en clair connu). Une autre attaque en texte clair **[5]** consiste à récupérer le contenu des trames, soit en forçant la génération de messages par un client vers un site contrôlé ou vers le réseau filaire Ethernet se trouvant derrière l'AP si on y a accès, puisqu'en ce beau pays, les trames gambadent déchiffrées **[6]**...

Enfin, énormément de logiciels de configuration de points d'accès ou de clients 802.11 dérivent leur clé WEP d'un mot de passe selon un algorithme connu, comme ici avec l'utilitaire *Nwepgen* :

```
cbr@elendil:~$ /sbin/nwepgen mon_pass 13
e6:74:fa:02:8d:b5:d4:e6:d3:50:fd:b6:bd
c2:55:e8:bb:4f:5a:0e:e5:3c:92:ab:84:3f
2a:45:d4:34:dc:bb:08:11:16:d9:8f:1c:58
2a:96:5c:dd:0d:ce:4d:7f:0b:6f:d8:d5:68
```



Airsnort

La première clé obtenue sera utilisée comme clé 0, la seconde comme clé 1, etc. Ainsi, lorsque nous voyons un AP protégé par du WEP, que nous l'avons identifié et que nous savons qu'il permet ce genre de pratique, il sera nettement plus simple et souvent plus rapide d'attaquer l'éventuel mot de passe générateur par dictionnaire ou force brute.

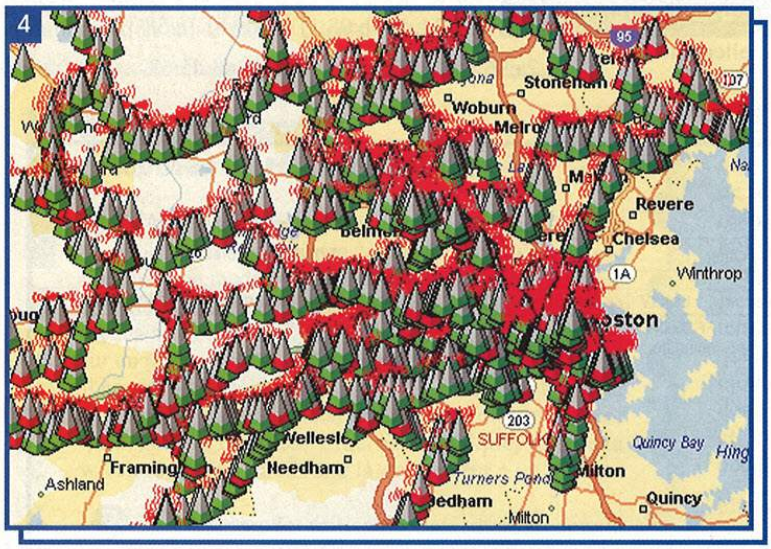
La page "Wireless Research" [7], en plus de l'article d'Éric Filiol et Frédéric Raynal [4], sera un excellent point de départ pour découvrir plus en détail les faiblesses du WEP.

WIFI SPOTTING

C'est grâce à de tels outils de détection de réseaux sans fil qu'a vu le jour le *wardriving* [8, 9]. Le wardriving consiste à localiser des points d'accès et enregistrer leur configuration. Le *wardriver* peut ainsi se constituer une carte des environs répertoriant tous les AP accessibles. Un outil comme *Network Stumbler*, associé à un périphérique GPS permet de générer automatiquement des cartes extrêmement lisibles [10], comme le montre la figure 4 (*Kismet* et *Airsnort* peuvent également tirer partie de ces modules). De telles cartes, selon les données collectées et celles retenues pour la génération, peuvent montrer :

- le fabricant et le modèle du point d'accès ;
- le canal utilisé ;
- le SSID utilisé ;
- la présence ou non de WEP et, le cas échéant, sa force, etc.

4



Carte réalisée à partir de données recueillies avec Network Stumbler

Un PDA équipé d'une carte 802.11b fera très bien l'affaire, aussi bien sous Linux que sous PocketPC (cf Mini Stumbler, port Network Stumbler pour PocketPC 3.0/2002), rendant alors la pratique on ne peut plus discrète.

ÉCOUTER N'EST PAS JOUER

L'écoute passive du réseau, aussi gênante soit-elle, est loin d'être la menace la plus sérieuse qui pèse sur votre environnement WiFi. Maintenant que nous savons qu'un méchant pirate peut s'installer au cœur de notre réseau aussi confortablement qu'un consultant dans son hamac, intéressons-nous un peu aux loisirs de l'animal.

VOUS CHANTIEZ ? EH BIEN, DANSEZ MAINTENANT !

Il est remarquablement aisé de créer un déni de service sur un réseau sans fil. Ainsi que l'indique l'article de ce dossier présentant le fonctionnement des standards 802.11, un type particulier de trames wireless, appelées *trames de gestion*, sert entre autres aux demandes d'association et de désassociation envoyées par un client à un point d'accès.

Or, les points d'accès sont généralement de grands naïfs, utilisant l'adresse MAC du client pour l'identifier de manière – théoriquement – unique. Ainsi, s'il reçoit une demande de désassociation en provenance d'une adresse MAC *X*, il obéit sagement et désassocie l'adresse MAC *X*. Vous l'aurez deviné, il suffit donc à notre pirate de s'attribuer l'adresse MAC d'une quelconque machine cible pour pouvoir l'empêcher d'utiliser le réseau sans fil, et ce sans bouger du fond de son hamac. Il ne reste qu'un petit pas à faire pour écrire un outil qui écouterait le réseau et émettrait une trame de désassociation aussitôt qu'il voit un client communiquer, et crac, plus de réseau.

Au-delà du cyber-vandalisme, ces dénis de services sont fort utiles pour pénétrer un réseau sans fil. Nous avons vu que nous pouvions attaquer le WEP en observant les authentifications. Or, ces évènements ne sont pas fréquents, et notre pirate allongé va devoir les provoquer pour en tirer partie au maximum. La plupart des *drivers* essayent de se réassocier automatiquement en cas de problème. Il suffira donc de forcer des désassociations pour générer autant de dialogues que nécessaire.

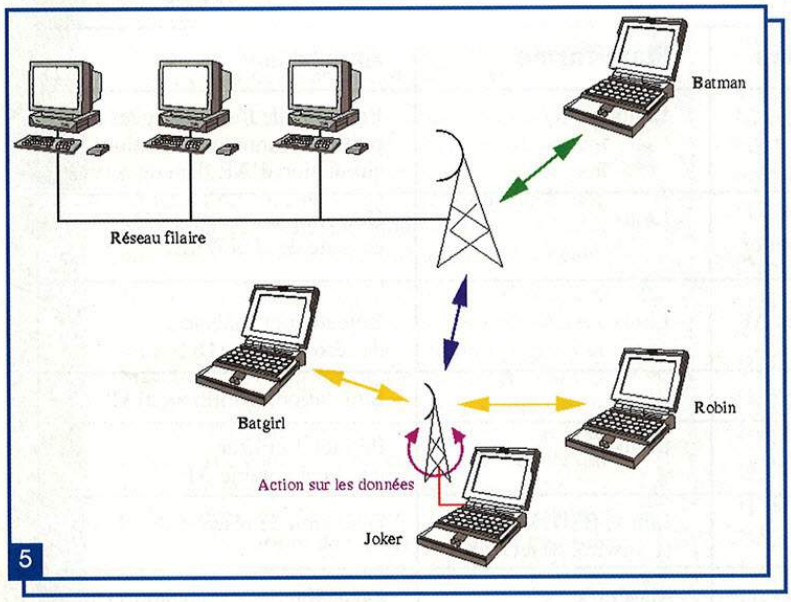
MAIS VOUS ÊTES QUI, VOUS ?

Puisque les points d'accès identifient les clients à partir de leur adresse MAC, il est assez peu surprenant que la majorité des technologies d'authentification s'appuient également sur l'adresse MAC. Ainsi, le point d'accès peut définir une liste de contrôle d'accès (ou *ACL*) statique précisant les adresses MAC autorisées à s'associer. Dans ce cas, il suffit encore une fois de s'approprier l'adresse MAC d'un client autorisé, subtilement obtenue par une écoute discrète du réseau, et l'AP n'y verra que du feu.

De plus, le mécanisme d'authentification par challenge/réponse vu précédemment souffre d'une faille qui permet à un attaquant de s'associer sans connaître la clé WEP. Si celle-ci lui sera nécessaire

pour communiquer de manière normale, cette association lui permettra d'injecter des trames valides sur le réseau en jouant avec une faille du système de contrôle d'intégrité [4, 11], qui lui permettra de modifier des trames arbitraires. De quoi mettre un peu le bazar dans ce monde hertzien.

Fort heureusement, des technologies d'authentification plus récentes et un tantinet plus évoluées existent. On peut citer parmi elles le 802.1x, qui permet de contraindre un client à s'authentifier avant de l'autoriser à utiliser un port d'un équipement actif, c'est-à-dire par exemple un port sur un switch, mais aussi une association sur un point d'accès. Cette authentification peut utiliser différentes méthodes, allant du simple couple utilisateur/mot de passe aux certificats, en passant par l'utilisation de *tokens* physiques. Un peu moins heureusement, une fois un client authentifié et donc associé au point d'accès, ce dernier ne retient que son adresse MAC comme critère pour autoriser ou non les communications. Ainsi, une fois un client associé, notre méchant pirate (qui n'a toujours pas décollé de son hamac) peut s'approprier son adresse et profiter de son authentification. Il y a deux points intéressants à noter quant à cette attaque. D'une part, contrairement au cas précédent, ici, il faut que le client dont on s'approprie l'adresse MAC soit présent et associé à l'AP. D'autre part, l'attaque est non disruptive, c'est-à-dire qu'elle ne perturbe pas le fonctionnement du client légitime dont nous dérobons l'identité. En effet, même si et le client légitime et l'attaquant verront chacun l'intégralité des trames destinées aux deux machines, pour peu que l'on prenne soin d'utiliser une adresse IP sur l'attaquant différente de celle du client légitime, la couche IP se chargera d'effectuer le tri entre les paquets, évitant par exemple que le client légitime ne réponde avec un TCP RST dès que l'attaquant essaye d'ouvrir une session TCP.



AP illégitime

En outre, l'association 802.1X/802.11 présente des failles découlant d'une part des spécifications 802.1X et d'autre part du manque caractérisé de confidentialité de 802.11 qui permettent de s'approprier des associations (*hijacking*) ou de mettre en place des attaques de type *Man In The Middle* pour s'authentifier [12]. Nous ne développerons pas ici ce type d'attaques, mais il est bon de savoir que 802.1X ne peut pas être efficace en wireless parce que la phase d authentification nécessite une confidentialité que les réseaux 802.11 ne peuvent pas apporter.

LE MONDE SANS FIL EST À VOUS

Une fois que nous sommes capables d'accéder à un AP, nous pouvons commencer à nous promener sur le réseau qui nous est ainsi ouvert et regarder ce qu'on peut faire avec. Il n'y a pas à chercher bien loin : on peut faire tout ce qu'il est possible de faire sur un réseau Ethernet classique, de l'usurpation d'adresse MAC comme vu précédemment, à la corruption de cache ARP qui entraîne, comme vous avez pu le lire dans un article [13] d'un MISC précédent, la compromission complète de votre domaine de *broadcast* Ethernet.

LICENSE TO FAKE

Si les redirections de flux de niveau 2 et 3 peuvent être intéressantes, être capable de mettre en place une redirection de plus bas niveau ouvre même de nouvelles possibilités. Une attaque assez simple à réaliser est la mise en place d'un point d'accès illégitime, plus couramment appelé *rogue AP*. Une fois activé, il sera capable de se substituer aux yeux d'un client 802.11 à tout AP de configuration équivalente dont la puissance reçue au point considéré serait plus faible.En effet, le mode de fonctionnement d'une carte 802.11 est de s'associer au signal le plus fort présentant les paramètres d'accès désirés (canal, SSID et clé WEP essentiellement). Ceci permet le passage d'un AP à un autre (*roaming*) lorsque la surface couverte par le réseau Wireless est trop grande pour un seul point d'accès. Ceci peut se révéler d'autant plus intéressant que certains drivers s'associent automatiquement par défaut au point d'accès le plus puissant si sa configuration est acceptable. Ce qui veut dire en termes simples que mettre en place un AP puissant sans aucune protection peut drainer vers vous un nombre conséquent de clients... De fait, on comprend mieux ce que l'introduction d'un AP non autorisé, malicieux ou non, sur un réseau peut avoir comme conséquences en termes de sécurité. À côté, les implications d'un modem RTC/RNIS planqué font figure de péché mignon.

À partir de là, notre intrus, toujours tranquillement installé dans son hamac et sirotant un bon jus de fruit, est à même de faire un peu ce qu'il veut du trafic généré par les clients qu'il vient de rediriger chez lui. L'application immédiate est une attaque de type

Man In The Middle. Il associe une seconde interface wireless à un AP légitime à qui il renvoie le trafic, après avoir fait ce qu'il avait à faire avec : écoute, rejet, modification, redirection, etc. Ce type d'attaque est possible pour deux raisons. La première tient dans le fait souvent ignoré que si vous avez deux AP de configuration identique, vous pouvez choisir celui auquel vous allez vous associer. La seconde repose sur le niveau de puissance des signaux reçus que vous pouvez moduler en utilisant des antennes plus ou moins directives, vers les clients d'une part et le véritable point d'accès d'autre part. Notez bien en outre que le trafic généré comprend évidemment le trafic émis par les clients, mais aussi le trafic qui leur est destiné, qu'il soit émis du réseau wireless ou non.

Il est évidemment possible de générer des dénis de services par ce biais. À l'aide d'un outil comme *FakeAP*, initialement conçu à des fins de protection contre le wardriving, nous pouvons simuler des points d'accès fictifs présentant une configuration adéquate pour que des clients tentent de s'y associer... En vain.

MA SÉCURITÉ S'EST ENVOLÉE !

Nous pouvons nous appuyer sur la figure **5** pour dresser une liste (non-exhaustive) de tout ce que pourrait faire notre intrus (Joker) sur le réseau qu'il attaque :

- écouter passivement tout le trafic généré par les trois clients wireless (mode *monitor*) ;
- générer des DoS sur l'ensemble des clients wireless ;
- intercepter localement le trafic généré par Robin et Batgirl ;
- détourner, détruire et modifier localement le trafic généré par Robin et Batgirl ;
- usurper localement l'identité de Robin et Batgirl ;
- écouter, intercepter, détourner, détruire et modifier activement le trafic généré par d'une part les clients wireless et d'autre part les stations connectées au réseau filaire (*ARP cache poisoning*) ;
- usurper l'identité des clients wireless et des stations filaires (*ARP cache poisoning*) ;
- accéder à toutes les ressources accessibles depuis ce réseau.

De quoi faire devenir fou et/ou paranoïaque n'importe quel RSSI un peu tatillon...

Nom	Chipsets supportés	Plate-forme	Application
Airjack [14]	Hermes, Prism2	Linux	Emission de trames forgées (en particulier trames de gestion), simulation d'AP, déni de service
Airsnort [15]	Hermes, Prism2, autres avec mode monitor	Linux	Détection de réseaux 802.11b, cassage de clés WEP
Airtraf [16]	Cisco, Prism2 via HostAP	Linux	Détection et analyse de réseaux 802.11b
FakeAP [17]	Prism2 via HostAP	Linux	Simulation de milliers d'AP
HostAP [18]	Prism2	Linux	Permet d'utiliser un client comme AP
Kismet [19]	Tous ceux ayant un mode monitor	Linux, BSD, Win32 (Cygwin), MacOS X	Détection de réseaux 802.11a/b
Macstumbler [20]	Airport	MasOS X	Détection de réseaux 802.11b
Mini Stumbler [21]	Hermes	PocketPC 3.0, 2002	Détection de réseaux 802.11b
Network Stumbler [21]	Hermes	Windows 9x, 2000, XP	Détection de réseaux 802.11b
Wavestumbler [22]	Hermes	Linux	Détection de réseaux 802.11b
Wellenreiter [23]	Cisco, Hermes, Prism2	Linux, BSD	Détection de réseaux 802.11b, brute-force des SSID
WifiScanner [24]	Prism2	Linux	Détection de réseaux 802.11b

LE TOOLKIT

La liste des outils permettant différents types d'analyses et d'attaques sur des réseaux sans fil est longue, mais il est intéressant de présenter brièvement quelques-uns des plus répandus. Les *chipsets* supportés indiqués dans la table ci-contre correspondent au cœur des cartes wireless. Le chipset Hermes est principalement utilisé dans les cartes Orinico de Lucent, mais pas seulement. D'autre part, les références au chipset Prism2 d'Intersil incluent généralement les versions 2.5 et 3 dudit chipset.

Pour la suite, les outils classiques que sont arp-sk [25], la suite dsniff [26] et un outil d'extraction/manipulation de paquets IP (Netfilter fera fort bien l'affaire) permettront à notre intrus d'exploiter cet accès sans fil.

CONCLUSION

Les réseaux sans fil ont, de par la nature même de la méthode de transmission, à faire face à une problématique de sécurité qui tenait jusque-là de la sécurité physique : l'accès au médium de communication. Lorsque cet accès est possible pour un intrus, il entraîne la compromission possible de toutes les données qui y sont transmises (i.e. le domaine de broadcast Ethernet).

Bien que le WEP ait été développé pour apporter une solution à ce problème, il échoue naïvement dans sa tâche. Les techniques mises en place sont faibles et insuffisantes pour garantir un accès sûr, à tel point que même des ajouts comme le 802.1X par exemple voient leur utilité réduite à néant.

Il est donc clair, et ce malgré les efforts des constructeurs pour essayer d'améliorer la sécurité du 802.11 (suppression des IV faibles, support de clés 256 bits ou protocoles propriétaires comme LEAP [27] de Cisco par exemple), qu'un réseau sans fil 802.11 basé sur du WEP ne peut pas et ne doit pas être considéré comme sûr. Il en résulte que la sécurité ne peut venir que des couches supérieures, par la mise en place de méthodes d'authentification, de chiffrement et de contrôle d'intégrité au niveau réseau (liens IPSEC) ou applicatif (SSH, SSL, etc.).

Cédric "Sid" Blancher –
sid@miscmag.com

Daniel "Bozo" Polombo –
bozo@miscmag.com

RÉFÉRENCES

[1] **http://www.ethereal.com/**

[2] D. Polombo, *"Introduction technique aux réseaux 802.11", MISC 6*, mars-avril 2003

[3] Discussion "Wlan @ bestbuy is cleartext?" **http://online.securityfocus.com/archive/82/270364/2002-04-28/2002-05-04/2**

[4] É. Filiol et F. Raynal, *"Les faiblesses du WEP, MISC 6"*, mars-avril 2003

[5] W. Arbaugh, *An Inductive Chosen Plaintext Attack against WEP/WEP2*, mai 2001

[6] N. Borisov, I. Goldberg et D. Wagner, *"(In)Security of the WEP algorithm"*, 2000 **http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html**

[7] **http://www.cs.umd.edu/~waa/wireless.html**

[8] **http://www.wardriving.com/**

[9] **http://www.wifi-montauban.net/communaute/index.php/TrebucherSansFil**

[10] **http://www.wifimaps.com/**

[11] A. Wan, W. Arbaugh, N. Shankar et Y.C. Justin, *"Your IEEE 802.11 Wireless Network Has No Clothes", IEEE International Conference on Wireless LAN's and Home Networks*, Singapour, 2001 **http://www.cs.umd.edu/~waa/wireless.pdf**

[12] A. Mishra et W. Arbaugh, *"An Initial Security Analysis of the IEEE 802.1X Protocol"*, février 2002 **http://www.cs.umd.edu/~waa/1x.pdf**

[13] C. Blancher, É. Detoisien et F. Raynal, *"Jouer avec le protocole ARP", MISC 3*, juillet-août 2002

[14] **http://802.11ninja.net/**

[15] **http://airsnort.shmoo.com/**

[16] **http://airtraf.sourceforge.net/**

[17] **http://www.blackalchemy.to/project/fakeap/**

[18] **http://hostap.epitest.fi/**

[19] **http://www.kismetwireless.net/**

[20] **http://www.macstumbler.com/**

[21] **http://www.stumbler.net/**

[22] **http://www.cqure.net/tools.jsp?id=8**

[23] **http://www.remote-exploit.org/**

[24] **http://wifiscanner.sourceforge.net/**

[25] **http://www.arp-sk.org/**

[26] **http://naughty.monkey.org/~dugsong/dsniff/**

[27] **http://www.cisco.com/warp/public/cc/pd/witc/ao350ap/prodlit/1515_pp.htm**

4 La sécurité

Deux standards existent pour les réseaux mobiles : Bluetooth et 802.11. Ce dernier comprend notamment un protocole assurant sa sécurité, le Wired Equivalent Privacy (WEP) fondé sur le système de chiffrement RC4. Le WEP offre d'indéniables avantages en termes d'efficacité sur les longues distances et de débit. Il est désormais relativement répandu, implémenté dans un grand nombre de cartes réseau sans fil, et de nombreux édifices professionnels et privés sont équipés pour la connectivité WEP. Or, récemment, ce protocole a fait l'objet d'attaques opérationnelles réellement implémentées, qui ont montré que, dans sa forme initiale, ce protocole n'est absolument pas sûr. Cet article a pour objectif de présenter les faiblesses du WEP et les attaques qui les exploitent.

INTRODUCTION

La plupart du temps, on considère que l'accès physique à un système (réseau ou machine) est amplement suffisant pour le compromettre. Avec les réseaux sans fil (WLAN), c'est encore plus facile puisque les messages viennent à vous sans que vous n'ayez rien (ou presque) à faire. Dans ces conditions, il est nécessaire de prévoir des protections supplémentaires par rapport à ce qui existe pour les réseaux filaires ordinaires. Ainsi, des mesures de sécurité ont directement été intégrées dans le protocole 802.11 pour gérer les problèmes d'authentification (i.e. prouver qui on est), d'association (i.e. s'assurer qu'on discute avec les membres du même WLAN), et de confidentialité (i.e. restreindre l'accès des messages aux entités désirées, typiquement, le client et le serveur). Outre ses qualités de performances, le protocole WEP est également simple à administrer. Le principe de base est que chaque appareil d'un réseau (une carte de communication ou un point d'accès (PA)) contient une clé (en pratique, la mise à la clé se fait à partir d'un mot de passe fourni par l'utilisateur). Cette clé est identiquement présente dans tous les périphériques des machines appartenant au réseau mobile considéré. On parle alors de clé partagée (*shared key*). La communication dans le réseau mobile est protégée contre les machines étrangères au réseau, c'est-à-dire celles ne possédant pas cette clé. L'objectif d'un attaquant est de la récupérer afin de s'incruster dans le réseau mobile.

Cette simplicité s'est très vite révélée dangereuse et les attaques n'ont pas tardé à apparaître. Ce furent d'abord Borisov, Goldberg et Wagner **[1]** qui montrèrent qu'un défaut de réinitialisation des cartes de communications WEP (par suite d'un manque de spécification concernant l'initialisation de l'algorithme de chiffrement RC4) conduisait les utilisateurs à employer les mêmes suites clefs (les suites aléatoires produites par RC4 et combinées au texte clair aux fins de chiffrement), ce qui constitue une grave faiblesse cryptologique permettant très facilement de retrouver les messages clairs. De plus, l'espace des vecteurs d'initialisation, par sa taille réduite, obligeait avec une forte probabilité une réutilisation de ces vecteurs, qui se traduit par la réutilisation des "suites-clé" (le rôle fondamental de ces vecteurs est précisément d'interdire cette éventualité).

Fluhrer, Mantin et Shamir ont publié en 2001 **[3]** une autre attaque passive utilisant uniquement les paquets chiffrés interceptés. Cette attaque exploite une faiblesse du protocole WEP concernant l'utilisation des vecteurs d'initialisation par RC4. Toutefois, cette attaque, théorique, n'avait pas été testée par ses auteurs et de fait sa validité non confirmée. En 2001, Stubblefield, Ionnadis et Rubin on effectivement implémenté cette attaque et prouvé ainsi que l'attaque de Fluhrer, Mantin et Shamir était valide et opérationnelle. Une autre implémentation de cette attaque (avec quelques améliorations) **[5]** a apporté la même confirmation.

du WEP

LE PROTOCOLE WEP

Le protocole WEP repose sur l'algorithme de chiffrement par flot RC4 [6] de Donald Rivest. Ce système utilise une clef de longueur variable (jusqu'à 2048 bits mais à l'export hors USA, la clef est limitée à 128 bits comme dans Acrobat par exemple). Le chiffrement intervient après une initialisation du système avec la clef secrète par le module KSA (*Key Scheduling Algorithm*). Ce module comporte au moins deux faiblesses conséquentes qui ont permis l'attaque décrite dans cet article. Le lecteur en trouvera dans [3] une description technique détaillée.

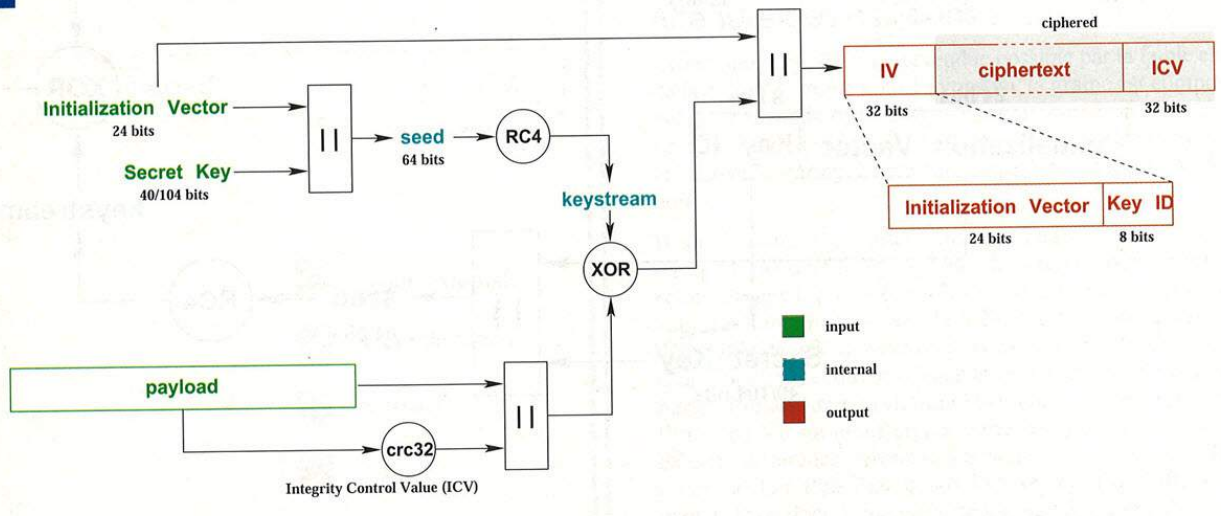
CONSTRUCTION DU MESSAGE CHIFFRÉ

Comment RC4 est-il utilisé dans le protocole WEP ? Le chiffrement dans ce protocole s'appuie sur une clef secrète K de 40 ou 104 bits, partagée par tous les éléments du réseau mobile. Pour le calcul d'une séquence (trame) WEP ("*frame*"), la séquence de clair correspondant M est d'abord concaténée avec une valeur de *checksum* sans clé ICV(M) de 32 bits (il s'agit ici d'une valeur calculée à partir de M pour en vérifier l'intégrité d'où le sigle ICV *Integrity Check Value*). La séquence M || ICV(M) est alors considérée (|| désigne l'opérateur de concaténation). Le code employé est le classique CRC-32 (*Cyclical Redundancy Check*). Cette checksum correspond en fait au reste dans la division (en binaire) du message par un diviseur fixé au préalable.

Un vecteur d'initialisation IV de 24 bits, généré pour chaque nouvelle séquence WEP, est accolé à la clef secrète selon le schéma IV || K. A noter que beaucoup de cartes WEP se contentent de générer IV à l'aide d'un simple compteur, d'autres à l'aide d'un

La figure 1 représente le schéma mis en place par le WEP. Précisons qu'en pratique, on dispose de 4 clés secrètes (dans le cas de clés de 40 bits), d'où la nécessité d'indiquer par le champ *Key ID* du paquet, laquelle a été réellement utilisée comme graine pour RC4. Il reste encore à ajouter les en-têtes appropriés pour obtenir un paquet 802.11 valide.

1



Chiffrement WEP

générateur aléatoire. La valeur IV || K est utilisée pour initialiser RC4 (on parle de graine, *seed*). La suite pseudo-aléatoire produite ensuite par RC4, appelée *keystream*, est additionnée bit à bit modulo 2 (simple opération de ou exclusif bit à bit) aux bits successifs qui composent M || ICV(M). On montre que si la suite possède des propriétés aléatoires fortes alors la combinaison de cette suite avec une suite non aléatoire, typiquement le texte clair, produit une suite qui, elle, est aléatoire donc non exploitable par l'attaquant. On définit une suite aléatoire par une suite de variables X_i, dites de Bernouilli, indépendantes et identiquement distribuées selon la loi uniforme, c'est-à-dire que $P[X_i = 1] = P[X_i = 0] = 0.5$ quel que soit i. Le résultat est la séquence chiffrée C donnée par :

$$C = (M||CRC32(M)) + RC4(IV||K)$$

où + ici désigne l'addition modulo 2 bit à bit, et C le chiffré qui transite sur le réseau.

DÉCHIFFREMENT DU MESSAGE

Pour le déchiffrement, c'est la même chose, mais à l'envers. Du paquet reçu, on extrait le vecteur d'initialisation qui est ensuite concaténé avec la clé partagée pour obtenir IV || K. On peut maintenant initialiser le RC4 pour générer la même suite pseudo-aléatoire qui a servi au chiffrement. Le texte clair est obtenu en refaisant un XOR entre les octets de cette suite pseudo-aléatoire et ceux, chiffrés, du paquet. La figure 2 décrit ce mécanisme.

DESCRIPTION DE RC4

Ce système est constitué de deux parties :

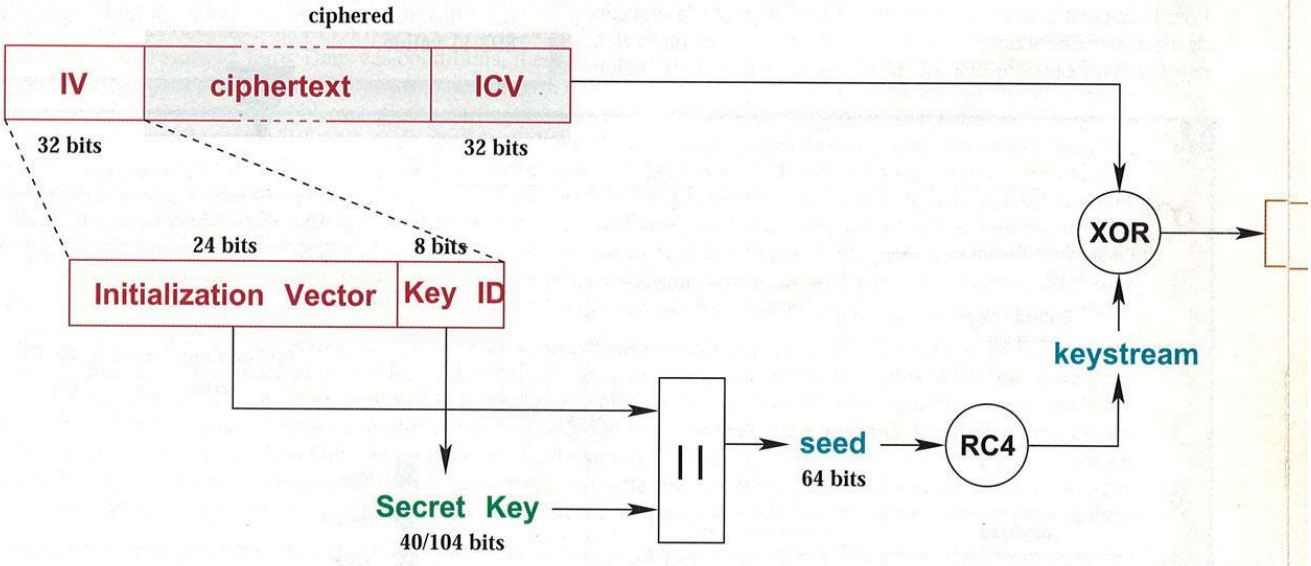
■ le module de mise à la clef KSA. Son rôle est de transformer une clef aléatoire (d'une taille usuelle allant de 40 à 256 bits) en une permutation S sur l'ensemble {0,...,N-1}, avec N = 256 en pratique. L'algorithme est le suivant pour une clef K' = IV || K (le paramètre l est égal au nombre de mots de la clef de $\log_2(K')$ bits) :

```
KSA(K')
Initialisation :
   pour i = 0,...,N - 1 faire
     S[i] = i
   finpour
   j = 0

Mélange :
   pour i = 0,...,N - 1 faire
     j = j + S[i] + K'[i mod l]
     échange(S[i], S[j])
   finpour
```

■ le module de génération d'aléa (*PRGA*) qui utilise ensuite la permutation S pour produire la suite pseudo-aléatoire. L'algorithme est le suivant :

2



Déchiffrement WEP

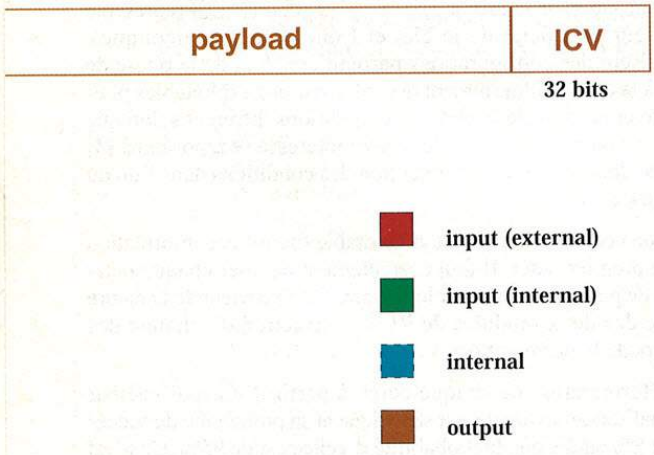
```
PRGA(K)
        Initialisation :
          i = 0
          j = 0
        Boucle de génération :
          i = i + 1
          j = j + S[i]
          échange(S[i], S[j])
          production de z = S[S[i] + S[j]]
```

A noter que la permutation `S` évolue très vite dans cette phase. La valeur `z` est alors combinée au texte clair via un XOR (OU-exclusif, i.e. une addition modulo 2).

LES MÉSAVENTURES DU WEP

Il n'a pas fallu beaucoup de temps pour que les gens s'intéressent au WEP, et en particulier à la sécurité qu'il offrait réellement ainsi qu'aux sources d'attaques, qui sont variées...

Signalons d'emblée que ces attaques n'ont rien de révolutionnaires. Elles sont toutes connues depuis très longtemps, mais appliquées à d'autres protocoles. Elles illustrent en tout cas que la mise en place d'un protocole, et en particulier quand il fait appel à de la cryptographie, ne peut pas se passer d'un expert du domaine, et que l'analyse de la communauté scientifique ne peut qu'être bénéfique sur la norme résultante.



PROBLÈMES DE HASARD OU CHOIX HASARDEUX

De la clé secrète... réellement secrète ?

La clé secrète doit être portée sur tous les systèmes qui veulent se connecter au WLAN. Par conséquent, elle est rarement changée. De plus, afin de s'en souvenir, la clé était, au moins dans les premiers temps, un mot de passe de 40 bits (soit 5 caractères !!!), avec toutes les faiblesses que cela induit (facilité pour les deviner, attaque par dictionnaire, etc.).

Pour remédier à cela, de nombreux équipements sont livrés avec un générateur de clés qui produit directement des valeurs hexadécimales. Pour cela, un générateur pseudo-aléatoire est initialisé avec une graine de 32 bits, cette graine étant construire à partir d'une *passphrase*. Toutefois, ces générateurs sont largement biaisés puisque la passphrase est composée uniquement de caractères issus du clavier de l'ordinateur, c'est-à-dire de caractères ASCII dont le bit de poids fort restera toujours nul. Ainsi, chaque octet de la graine appartient à l'ensemble {0x00, ..., 0x7F} et non {0x00, ..., 0xFF}. L'espace décrivant la graine du générateur aléatoire va donc de `00:00:00:00` à `7f:7f:7f:7f`.

En fonction du générateur utilisé, d'autres attaques sont possibles, mais nous nous éloignerions du sujet. Une étude assez complète [8], réalisée par T. Newsham, montre que cette méthode limite la taille de l'espace des clés à 2^{21}.

L'utilisation d'une clé secrète de 104 bits s'avère donc indispensable. Sa génération repose encore sur une passphrase, mais cette fois, la fonction MD5 est utilisée, et seuls les 104 premiers bits de la sortie sont conservés. La sécurité se situe donc ici encore une fois dans la passphrase, et dans la fonction MD5. La force brute n'est alors envisageable que si vous tenez à ce que vos arrières arrières petits enfants obtiennent le résultat. On préfère alors les attaques par dictionnaires.

Attendre des collisions

Une attaque simplissime est rendue possible par la faible entropie de la graine. Comme nous l'avons vu, la graine est composée de 64/128 bits, mais seuls 24 bits varient entre 2 paquets, ceux du vecteur d'initialisation ! Les 40 (ou 104) autres bits sont ceux de la clé secrète partagée entre tous les équipements. Ainsi, il existe donc 2^{24}.

Il suffit donc à un attaquant d'attendre tranquillement en enregistrant tout le trafic pour qu'une collision intervienne relativement rapidement, c'est-à-dire lorsque deux paquets capturés sont chiffrés avec la même clé (ce qui est facilement détectable dans la mesure où le vecteur d'initialisation passe en clair d'une part, et qu'on sait que la clé secrète est la même d'autre part). Lorsqu'une collision survient, on obtient alors de l'information sur la différence entre les clairs. La connaissance de cette différence permet des attaques statistiques qui donnent accès au clair. Plus l'attaquant dispose de clairs chiffrés avec la même clé (et donc le même vecteur d'initialisation), plus l'attaque porte ses fruits. Une fois un clair obtenu, l'attaquant est alors en mesure de récupérer tous les autres.

CONTRÔLE D'INTÉGRITÉ AVEC CRC 32 ?

L'utilisation de CRC pour le contrôle d'intégrité des messages est plus que douteuse. En effet, CRC est initialement conçu pour la détection d'erreur, mais certainement pas pour des tests d'intégrité. Ces derniers reposent sur des fonctions de hachage cryptographiques. Elles sont construites pour satisfaire un certain nombre de propriétés, et en particulier l'impossibilité de reconstruire le haché lorsqu'on modifie le message.

Pour cela, les fonctions sont généralement non linéaires (une fonction est linéaire lorsque `f(x+y)=f(x)+f(y)`), ce qui n'est pas le cas du CRC. Il est dès lors possible de modifier le contenu chiffré du paquet (c'est-à-dire à la fois le *payload* et la *checksum*) de sorte à ce que le récepteur du paquet ne remarque pas la modification.

Par exemple, si on considère un message `X`, le chiffrement est donné par `RC4(K)+ X||CRC(X)`. Un attaquant qui capture le trafic voit passer cette trame et décide de la modifier. Il ne connaît donc pas le clair, mais il peut quand même agir grâce à (ou "à cause de" au choix) à la linéarité de `CRC CRC(X+Y)=CRC(X)+CRC(Y)` où `Y` est une chaîne binaire quelconque. Quant au chiffrement RC4, il est réalisé par un `XOR (+)` entre le message en clair et une clé, et on a donc `RC4(K)+(X+Y)=(RC4(K)+X)+Y`.

Ainsi, un attaquant peut choisir de modifier certains bits du message chiffrés puis ajuster les bits correspondants dans le CRC, chiffrés de sorte que le récepteur ne remarque rien ! Il écoute le trafic et capture donc `RC4(K)+X||CRC(X)`. L'attaquant décide de modifier la trame capturée avec la chaîne binaire `Y` et dont la checksum est donnée par `CRC(Y)` :

```
RC4(K)+(X+Y||CRC(X+Y)) = RC4(K) + (X+Y||CRC(X)+CRC(Y))
                       = RC4(K) + X||CRC(X) + Y||CRC(Y)
                       = (RC4(K) + X||CRC(X)) + Y||CRC(Y)
```

où chaque terme situé à droite de l'équation est connu de l'attaquant.

Les paquets étant chiffrés, on peut se dire que cette attaque ne sert pas à grand-chose. Sauf que la nature est bien faite ;-) Revenons sommairement sur l'authentification d'un client auprès d'un PA. Le client contacte initialement le PA pour lui dire qu'il voudrait bien s'associer. Pour vérifier que le client connaît bien la clé WEP, le PA renvoie un message en clair. Le client le chiffre alors avec sa clé WEP et renvoie ce chiffré au PA. Si le chiffré est bien celui attendu, le PA valide l'association.

Outre l'attaque à clair connu ainsi envisageable (et ce d'autant plus facilement qu'un attaquant peut provoquer la désassociation du client et du PA, ce qui conduit à un nouveau challenge, et donc une nouvelle paire (clair, chiffré)), les problèmes de CRC prennent ici toute leur importance.

Supposons qu'un méchant pirate écoute sur le réseau, et capture une paire (clair, chiffré) (on notera `X` le clair, et `RC4(K)+X||CRC(X)` le chiffré correspondant à la trame renvoyée). Ensuite, il envoie une demande d'association au PA. Celui-ci lui retourne un message en clair en guise de défi (notons `Y` ce message en clair). L'attaquant doit alors retourner le message `RC4(K)+Y||CRC(Y)` au PA pour être associé, ce qu'il est en mesure de faire même s'il ne connaît pas la clé K :

```
RC4(K)+ Y||CRC(Y) = RC4(K) + Y||CRC(Y) + X||CRC(X) + X||CRC(X)
                  = (RC4(K) + X||CRC(X)) + Y||CRC(Y) + X||CRC(X)
```

Et justement, tous les éléments sont connus dans cette dernière équation :

```
RC4(K) + X||CRC(X) est le chiffré retourné par le client ;
X||CRC(X) est le clair envoyé en guise de challenge par le PA
Y||CRC(Y) est le clair reçu par l'attaquant pour sa propre authentification.
```

Le tour est joué ! Signalons que cela fonctionne parce que, outre le fait que l'intégrité est calculée sans dépendre d'une clé, le PA a la gentillesse de laisser le client choisir un vecteur d'initialisation. En effet, l'attaquant, pour forger sa réponse, utilise le même vecteur que celui qu'il a capturé dans le précédent challenge en écoutant le réseau.

Certes, l'attaquant ne peut pas communiquer sur le WLAN puisqu'il ne connaît pas la clé WEP, mais il est présent et associé sur ce WLAN. De plus, cette approche est adaptable pour certaines trames dont les paramètres sont bien connus (message ARP, demande/réponse DHCP, etc.).

L'ATTAQUE DE FLUHRER, MANTIN ET SHAMIR

L'attaque utilise seulement le premier octet `oc_0` produit par la suite de sortie. L'équation pour ce premier octet est la suivante :

```
S[S[1] + S[S[1]]] = oc_0
```

Après la phase de mise à la clef, `oc_0` ne dépend que de trois valeurs de la permutation :

```
(S[1], S[S[1]], S[S[1] + S[S[1]]])
```

L'attaque est alors fondée sur la capacité à déduire de l'information sur la clef secrète `K` en observant ces valeurs. Cette information sera obtenue grâce au vecteur d'initialisation `IV` composant la clef `K'`. Certains vecteurs `IV` particuliers, dits "favorables" (sans entrer dans les détails, il s'agit de vecteurs d'initialisation de la forme `(Octetcle + 3, 255, X)`, où `Octetcle` représente la valeur de l'octet de la clé, et `X` une valeur quelconque), produisent des configurations particulières lors de la phase de mise à la clef, qui fournissent des informations exploitables pour retrouver des bits de la clef `K` (les conditions diffèrent selon que `K' = IV || K` ou que `K' = K || IV` ; le lecteur intéressé se reportera à **[3]** pour le détail technique d'obtention des conditions dans l'un ou l'autre des cas).

Chaque vecteur d'initialisation favorable fournit une information sur le premier octet. Il faut correctement deviner chaque octet avant de pouvoir déterminer le suivant. Cela provient de la nature même des deux modules de RC4 (instruction d'échange des valeurs de la permutation `S`).

La détermination de chaque octet, à partir d'un seul vecteur d'initialisation favorable, est statistique et sa probabilité de succès est de 5% tandis que la probabilité d'échec est de 95%. Ce n'est donc qu'en considérant un grand nombre de vecteurs d'initialisation favorables, et donc de trames différentes, qu'on obtiendra la valeur exacte de chaque octet.

Comme l'attaque repose sur une estimation statistique, elle sera donc d'autant plus rapide que le trafic capturé sera important.

Le lecteur trouvera des exemples détaillés dans [5] de la technique de détermination des octets.

IMPLÉMENTATIONS PRATIQUES DE L'ATTAQUE DE BASE

Tout le problème est de pouvoir déterminer le nombre minimal de vecteurs d'initialisation nécessaires à la détermination certaine d'un octet, de capturer effectivement les trames concernées et de monter l'attaque en pratique. La faisabilité technique de l'attaque a été démontrée dans [4] par sa réalisation en grandeur réelle alors que l'attaque n'était décrite dans [3] que de manière théorique.

La capture des paquets a représenté selon les auteurs la partie la plus longue. Cela illustre bien le décalage fréquent entre la théorie et sa mise en application pratique. De nombreux logiciels (commerciaux pour la plupart) existent, qui permettent la capture et le décodage des trames 802.11. Ces logiciels étant très onéreux, et afin de montrer la faisabilité pratique d'une attaque réelle par un attaquant aux ressources limitées, les auteurs de [4] ont choisi la configuration suivante :

- une carte *Linksys* (100 euros) basée sur le chipset *Intersil Prism II*.
- le driver Linux *linux-wlan-ng-prism2* [7] et le patch pour l'observation des trames [8].
- le logiciel *ethereal* [9] pour la capture des trames chiffrées et leur décodage.

En final, l'expérience a montré que l'attaque de base nécessitait un peu plus de trames que prévu dans [3] : entre 5.000.000 et 6.000.000 au lieu de 4.000.000 pour retrouver la clef K. Cette quantité est relativement faible pour monter une attaque réaliste à partir de l'observation pendant quelques heures d'un réseau moyennement saturé. L'algorithme est le suivant :

```
DetCleWEP()
 Cle[0,...,TailleCle] = 0
 pour octetCle = 0,...,TailleCle faire
   Compteur[0,..,255] = 0
   pour chaque trame T
     si IV de T est de la forme (Octetcle + 3, 255, N) (valeur de N entre 0
et 255)
        alors Compteur[OctetSuppose]++
     finsi
   finpour
   Clef[octetCle] = IndexElementMax(Compteur);
 finpour
 retourner Cle
```

Les auteurs de [4], sur la base de constatations faites par les auteurs de [3], ont implémenté une version optimisée de cette attaque. Le nombre de trames nécessaires tombe alors à environ 1.000.000. Dans cette version, l'information fournie par la valeur CRC32(M) est utilisée.

Une autre amélioration apportée par Hulton [5] est de ne pas se limiter au premier octet de clé, mais également de tenir compte des octets suivants. Si ce premier octet est celui qui révèle le plus d'information sur la clé, les octets suivants dévoilent aussi parfois de l'information, et même si elle est moins fiable, elle n'en est pas moins exploitable.

A NOUS DE JOUER AUSSI

SNIFFER SUR L'AIR

La mise en place d'un sniffer n'est pas aussi simple que sur des réseaux filaires. En effet, passer l'interface en mode *promiscuous* ne suffit pas. Ce mode annule le filtrage de niveau 2 sur les adresses MAC, ce qui provoque la remontée de tous les paquets qui arrivent sur une interface réseau, sans tenir compte de l'adresse IP.

Qu'arrive-t-il si on passe une interface *wireless* en mode promiscuous ? Rien dans la plupart des cas ! Et c'est normal. En fait, les trames Ethernet sont encapsulées dans des trames 802.11b. Lorsqu'un paquet arrive sur l'interface, il est traité comme paquet 802.11b, et ensuite seulement comme paquet Ethernet. De plus, la carte doit être associée à un PA pour qu'elle reçoive bien des paquets. Pour qu'une interface avec ce mode capture des données, il est donc nécessaire d'agir préalablement au niveau 802.11.

Pour recevoir les paquets, il faut passer la carte wireless en mode *monitor*. Dans un tel état, la carte n'est plus capable de se comporter comme une interface réseau normale, et toutes les connexions qui transitent par elle tomberont. Toutefois, elle réceptionne alors tout le trafic qui passe dans son voisinage.

L'ENTRÉE DES ARTISTES

Les intervenants de cette histoire sont au nombre de deux. Tout d'abord, il y a `joker`, un iBook avec une carte Airport. Ensuite, il y a `twoface`, un PC sous Linux qui nous sert à sniffer. Les outils "classiques" supportent de mieux en mieux le 802.11. Pour nos expériences, nous utilisons un Linux 2.4.20 et, comme driver pour l'interface wireless, `linux-wlan-ng-0.1.16-pre8`. Certains logiciels ont besoin d'être patchés ou de patcher d'autres bibliothèques pour fonctionner correctement. Nous signalerons cela au cas par cas. Enfin, il y a un point d'accès, configuré pour du WEP 64 bits et qui connaît, codées en dur, les adresses MAC qui ont le droit de passer par lui.

La première chose à déterminer est le canal employé par le WLAN. Cela se fait assez simplement à l'aide de différents outils (voir l'article de C. Blancher et D. Polombo dans ce même numéro), ou bien "artisanalement" en écoutant sur les 13 canaux pour déterminer ceux intéressants. Dans notre configuration, le canal 11 est utilisé. Signalons que certaines cartes (type Cisco Aironet) supportent tout à fait bien de sniffer sur plusieurs canaux en même temps.

Regardons d'abord ce qu'il se passe :

```
root@twoface# wlanctl-ng wlan0 lnxreq_wlansniff channel=11 prismheader=false
wlanheader=false keepwepflags=false stripfcs=false enable=true
message=lnxreq_wlansniff
  enable=true
  channel=11
  prismheader=false
  wlanheader=false
  keepwepflags=false
  stripfcs=false
  packet_trunc=no_value
  resultcode=success
root@twoface# tcpdump -e -n -s 0 -X -i wlan0
tcpdump: listening on wlan0
11:29:21.350465 BSSID:0:9:5b:35:72:91 DA:ff:ff:ff:ff:ff:ff SA:0:9:5b:35:72:91 Beacon
() [ 11.0 Mbit] ESS CH: b , PRIVACY
0x0000          4052 976d 0f00 0000 6400 1100 0006 0000      @R.m....d.......
0x0010          0000 0000 0104 8284 8b96 0301 0b05 0400      ................
0x0020          0200 00ff ffff ff                            .....
11:29:21.452862 BSSID:0:9:5b:35:72:91 DA:ff:ff:ff:ff:ff:ff SA:0:9:5b:35:72:91 Beacon
() [ 11.0 Mbit] ESS CH: b , PRIVACY
0x0000          3de2 986d 0f00 0000 6400 1100 0006 0000      =..m....d.......
0x0010          0000 0000 0104 8284 8b96 0301 0b05 0401      ................
0x0020          0200 00ff ffff ff                            .....
...
```

Nous avons donc tout d'abord autorisé le *sniffing* sur l'interface `wlan0` pour le canal 11, ce qui provoque le passage en mode monitor. Ensuite, avec Tcpdump, nous récupérons le trafic. Ici, des *Beacon Frames* sont émises à intervalle régulier. Celles-ci nous indiquent que le WEP est activé (PRIVACY), que le canal est le 11 (CH: b, même si nous le savions déjà), que le débit est réglé sur 11 Mbits, que nous sommes sur un réseau type ESS (*Extented Service Set*) dont le PA a pour adresse MAC 0:9:5b:35:72:91).

Continuons à écouter jusqu'au moment où `joker` se connecte au réseau (nous enlevons le "bruit") :

```
11:30:41.698587 BSSID:ff:ff:ff:ff:ff:ff DA:ff:ff:ff:ff:ff:ff SA:0:30:65:25:35:3d
Probe Request (gotham) [ 11.0 Mbit]
0x0000   0006 676f 7468 616d 0104 0204 0b16 ffff        ..gotham........
0x0010   ffff                                           ..
11:30:41.699388 BSSID:0:9:5b:35:72:91 DA:0:30:65:25:35:3d SA:0:9:5b:35:72:91 Probe
Response (gotham) CH: b ,PRIVACY
0x0000   cafb 029f 0100 0000 6400 1100 0006 676f        ........d.....go
0x0010   7468 616d 0104 8284 8b96 0301 0bff ffff        tham............
0x0020   ff                                             .
11:30:41.699665 RA:0:9:5b:35:72:91 Acknowledgment
```

Et hop, on récupère tranquillement le SSID du réseau (`gotham`).

L'attaque est maintenant une question de patience et de chance. Comme nous l'avons vu, elle repose sur des estimations statistiques. Donc, plus nous capturons de paquets, plus l'attaque réussit vite. Les outils permettant cela sont nombreux (voir article de D. Polombo et C. Blancher dans ce dossier). Selon la taille de la clé utilisée, et le nombre de "clés faibles" dont on dispose, on s'appuiera sur une attaque par dictionnaire ou celle décrite par Fluhrer, Mantin et Shamir.

CONCLUSION

L'attaque du protocole WEP est intéressante car elle illustre parfaitement le fait que la sécurité cryptologique ne se résume pas à un algorithme de cryptographie fort. Bien souvent, des attaques relativement aisées parviennent à réduire la sécurité d'un protocole en miettes en visant le système au niveau de son implémentation ou de sa gestion. C'est l'image de "la porte blindée sur un mur en carton". Et dans ce contexte, la sécurité informatique prend une importance capitale car tout repose sur elle. Le protocole WEP est en cours de modification pour tenir compte des attaques récentes mais combien de systèmes intégrant de la cryptologie sont encore vulnérables à des attaques d'implémentation ? Pour le moment, les constructeurs intègrent directement sur le matériel des mécanismes de détection des clés faibles pour éviter qu'elles ne transitent sur le réseau.

Eric Filiol
Ecole Supérieure et d'Application des Transmissions
Laboratoire de cryptologie et de virologie
efiliol@esat.terre.defense.gouv.fr
http://www-ocq.inria.fr/codes/Eric.Filiol/index.html

Frédéric Raynal
pappy@miscmag.com
http://www.security-labs.org

RÉFÉRENCES

[1] N. Borisov, I. Goldberg et D. Wagner - *Intercepting mobile communications: the insecurity of 802.11*. In MOBICOM 2001, Rome, Juillet 2001.

[2] Wireless LAN medium access control (MAC) and physical layer(PHY) specifications - *IEEE Standard 802.11*. L.M.S.C. of the IEEE Society, 1999.

[3] S. Fluhrer, I. Mantin et A. Shamir - *Weaknesses in the key scheduling algorithm of RC4*. Proceeding of SAC'2000, Lecture Note in Computer Science, Springer, 2001.

[4] A. Stubblefield, J. Ioannidis et A.D. Rubin - *Using the Fluhrer, Mantin and Shamir attack to break WEP*, 2002.

[5] D. Hulton - *Practical exploitation of RC4 weaknesses in WEP environments*, - Février 2002. **http://www.dachb0den.com/projects/bsd-airtools.html**

[6] R.L. Rivest - *The RC4 Encryption Algorithm*. RSA Data Security, Inc, Mars 1992.

[7] Disponible sur **http://www.linux-wlan.com/**

[8] Disponible sur **http://www.lava.net/~newsham/wlan/**

[9] Disponible sur **http://www.ethereal.com/**

5 BlueTooth, Hiperlan : les autres normes Wireless

Relativement récents, les réseaux sans fil deviennent de plus en plus performants grâce notamment aux avancées de l'électronique et du traitement du signal. Nous allons particulièrement nous intéresser dans cet article à mieux comprendre les autres normes que sont Hiperlan, HomeRF et surtout Bluetooth.

UN PEU D'HISTOIRE

Le premier réseau commercial analogique sans fil, au niveau des opérateurs, a vu le jour en 1982 à Chicago. En 1986, France Télécom lance Radiocom 2000 en France. Les premiers réseaux GSM (numériques) apparaissent en France en 1992 et remportent le succès que nous connaissons. Que ce soit au niveau des opérateurs (GSM, GPRS, UMTS), au niveau local (WLAN), ou au niveau domestique (WPAN), de nombreuses applications intéressantes sont envisagées. Les terminaux s'acheminent vers un support indifférencié de plusieurs protocoles, passant de l'un à l'autre sans rupture de la connexion en fonction de l'endroit où l'on se trouve : GPRS, UMTS, WLAN, Bluetooth. Par exemple, lorsque l'on arrive dans un lieu public pour une conférence, on passe sur un WLAN plus rapide que l'UMTS. Autre exemple, actuellement lorsqu'un TGV passe d'une cellule GSM à une autre, les protocoles de changement de cellule se font simultanément pour un grand nombre de personnes téléphonant. Demain, un WLAN permettra de bénéficier du réseau à l'intérieur du TGV. Le software gérera le choix du protocole à un moment donné : plus de 80% de la R&D dans l'Internet se fait sur le logiciel et non le matériel, selon Gilles Kahn, directeur scientifique de l'INRIA. Les réseaux sans fil se développent très rapidement et devraient représenter un énorme marché en ce début de 21ième siècle. Les prix, jusque-là inaccessibles, deviennent de plus en plus abordables, les performances et les débits augmentent, les réseaux domestiques et la population de travailleurs mobiles également.

BLUETOOTH, HIPERLAN, HOMERF : DE L'EXOTISME AU PAYS DU SANS FIL ?

Très en vogue avec le développement de Bluetooth, les réseaux sans fil individuels de type WPAN devraient, avec le temps, augmenter leurs débits et leurs portées pour devenir de véritables concurrents des WLAN. C'est à ce titre qu'ils sont évoqués ici. Les deux noms qui sortent du lot des WPAN sont Bluetooth et HomeRF. Sur le site de Bluetooth, on peut lire que 2000 constructeurs les suivent, tandis que 100 seulement soutiennent le projet HomeRF. Sur le site de HomeRF, on réplique à cela en précisant que l'inscription au soutien de Bluetooth est gratuite alors que ce n'est pas le cas pour HomeRF. Les deux normes précisent clairement qu'elles mènent une véritable croisade pour unifier l'ensemble des constructeurs. Même les sacro-saints "White Papers" sont d'une mauvaise foi évidente… *Home Radio Frequency* (HomeRF) est une norme fondée sur les normes 802.11b et DECT. Elle permet indifféremment de faire transiter des flux audio ou des données. La norme autorise des portées de 50 mètres sans utiliser d'amplificateur. De même que Bluetooth, HomeRF travaille au niveau physique dans la bande des 2,4 GHz, avec la technologie FHSS, à raison de 50 sauts de fréquence par seconde. Le débit nominal est de 1,6 Mbps (soit 1 Mbps réel), mais les débits peuvent atteindre 10 Mbps dans la version 2. Les applications possibles de ces technologies laissent libre cours à l'imagination. Si celles-ci se développent, nul doute que notre vie en sera bouleversée, et que de nouveaux débats éthiques surgiront. Ces applications permettent en effet de surveiller et d'enregistrer la totalité des faits et gestes des personnes et la liberté de chacun pourrait être menacée.

HOMERF (HOME RADIO FREQUENCY)

Initiée par le *Home Radio Frequency Working Group* (WG) (comprenant Compaq, Hewlett-Packard, IBM, Intel et Microsoft), HomeRF, davantage positionné vers le marché domestique, est née en 1998, et permet de transporter indifféremment voix et données dans un rayon de 50 à 100 mètres avec un débit de 10 Mbits/s (en fait 3 à 4 MBits/s à se partager entre tous les utilisateurs de la cellule).

Elle permet de mettre en place des communications de données sans fil mais également une liaison DECT pour transporter la voix (le DECT est la norme utilisée pour transmettre la voix en mode numérique entre un téléphone sans fil et sa base fixe). Simple d'usage et souple d'utilisation, peu de produits existent à ce jour.

HomeRF est un protocole spécifique dérivé de l'OpenAir, du standard 802.11 de l'IEEE (*Institute of Electrical and Electronic Engineers* – Groupe de travail et de standardisation international) et des technologies de téléphonie sans fil TDMA adaptées du DECT (*Digitally Enhanced Cordless Telephone*, standard qui n'est pas disponible aux Etats-Unis). La HomeRF 1.2 utilise une technique d'étalement de spectre à sauts de fréquence dans la bande des 2,4 GHz et offre un débit maximal théorique de 1,6 Mbps, sur une portée réelle d'une trentaine de mètres (soit à peu près aussi bien que les premières technologies de réseau sans fil).

Elle supporte simultanément une topologie de réseau client/serveur (pour partager une connexion Internet, par exemple) et point à point (échange de fichiers entre deux postes). Elle diffère donc de la norme 802.11b, qui s'est imposée pour les réseaux locaux d'entreprise grâce à son débit (11 Mbps) et à sa portée utile (100 mètres environ), et qui a été aidée par une forte baisse des prix des composants.

Le 802.11, en revanche, fonctionne en séquence directe (longueur d'onde constante), ce qui le rend moins apte à traverser les murs, et la compatibilité entre des appareils de marques différentes reste souvent aléatoire. Il résiste également aux interférences. Néanmoins, cette norme fait partie depuis peu de l'histoire. Déjà moribond, le projet HomeRF a vu son groupe de travail se dissoudre : il y a quelques semaines, Intel a annoncé son retrait du projet pour se concentrer entièrement au 802.11.

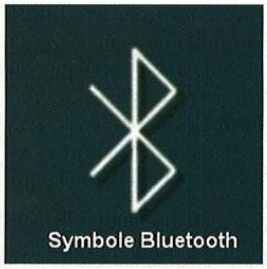
HIPERLAN 1 ET HIPERLAN 2 (HIGH PERFORMANCE RADIO LAN, VERSION 2)

Initiée et adoptée par l'*European Telecommunications Standards Institute* (ETSI), Hiperlan 1 permet un transfert autour de 20 Mbps/s. dans la gamme de fréquence de 5 Ghz. Hiperlan 2, son successeur, monte jusqu'à 54 Mbps/s. dans la même gamme de fréquence.

Cette norme a le défaut d'être uniquement européenne, les Américains ne travaillant pas à sa standardisation. Hiperlan utilise la même couche physique que l'IEEE 802.11, mais il n'est pas compatible avec lui. Ce standard de WLAN a été défini dans sa version 1 par le comité RES-10 du projet BRAN (*Broadband Radio Access Networks*) de l'ETSI le 16 juillet 1998.

Dans cette première version, les communications peuvent se faire sur 5 canaux distincts de priorité différente. L'adaptation du CSMA/CD appelée EY-NPMA (*Elimination Yield None Preemptive Priority Multiple Access*) consiste à scruter les canaux par ordre de priorité jusqu'à trouver un canal libre pour émettre. Le niveau 2 (liaison) du modèle OSI est divisée en deux sous-couches, la sous-couche CAC (*Channel Access Control*) qui correspond à la partie physique de la technique d'accès (gestion des problèmes liés au canal hertzien ainsi que toute la transmission et réception), et la sous-couche MAC qui correspond à la partie logique, soit la mise en forme de la trame, le routage interne, les algorithmes de confidentialité, la gestion de priorité et l'insertion et le retrait des stations.

Hiperlan 2 est soutenu par l'H2GF (*Hiperlan 2 Global Forum*), fondé en 1999 par Bosch, Dell, Ericsson, Nokia, Telia et Texas Instrument. Ils ont été rejoints un an après par d'autres industriels, tels Canon, Motorola ou encore Samsung. Les géants Cisco, Intel, Lucent ou Nortel sont toujours absents de ce forum. Cette deuxième version propose un débit de pointe à 54 Mbps et utilise, au niveau physique, le protocole OFDM (de la même façon que 802.11a). L'avenir de cette technologie, essentiellement européenne, sur les réseaux sans fil semble hypothéqué par le manque d'intérêt des industriels (surtout américains), même si ses spécifications doivent servir de base à des normes qui offriront un plus haut débit.

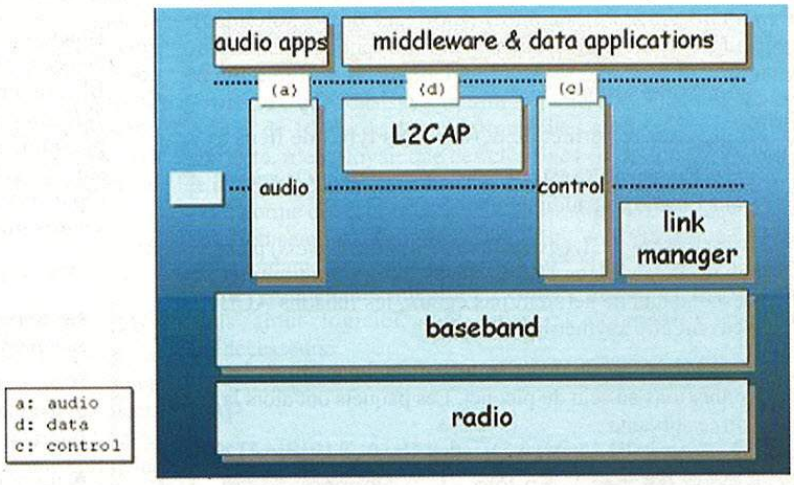

Symbole Bluetooth

BLUETOOTH

Cette technologie sans fil tient son nom d'un fait d'armes historique. Harald Blaatand (910 – 986), littéralement "Harald à la dent bleue", unifia le Danemark et la Norvège, royaumes vikings, dans une Europe divisée par des querelles de religion et de territoires. D'où le nom de Bluetooth en anglais. Initié par Ericsson en 1994, rapidement rejoint par IBM, Intel, Nokia et Toshiba au sein du "Bluetooth SIG" (*Bluetooth Special Interest Group*), le but de Bluetooth est d'unifier l'ensemble des constructeurs autour d'une seule norme sans fil : elle-même. Bien sûr, l'enjeu commercial à l'époque "d'Harald à la dent bleue" n'était pas le même, et il n'est pas certain que l'union soit encore possible dans le monde sans pitié des technologies de l'information et de la communication où la hache a été remplacée par les millions d'euros ! Aujourd'hui, 2400 constructeurs, dont 3Com, Motorola et Microsoft, en plus des multinationales précédemment citées, se sont ralliés à cette cause. Bluetooth est une technologie de moyen débit 720 kbits/s (1 Mbits/s en débit théorique) à basse consommation énergétique. Son rayon d'action est limité entre 10 et 30 mètres et ses composants de petite taille lui permettent d'être insérée dans des équipements très mobiles (mais pas seulement). Bluetooth permet de créer un réseau de 8 appareils en communication simultanée. Adaptée à la mobilité (montre, PDA, voiture, espaces publics...), son faible rayon d'action fait sa force et sa faiblesse. Les communications s'établissent sur la bande de fréquences 2400 – 2483,5 MHz. Le débit de base est de 1 Mbits/s. Les données sont envoyées par paquets, comme

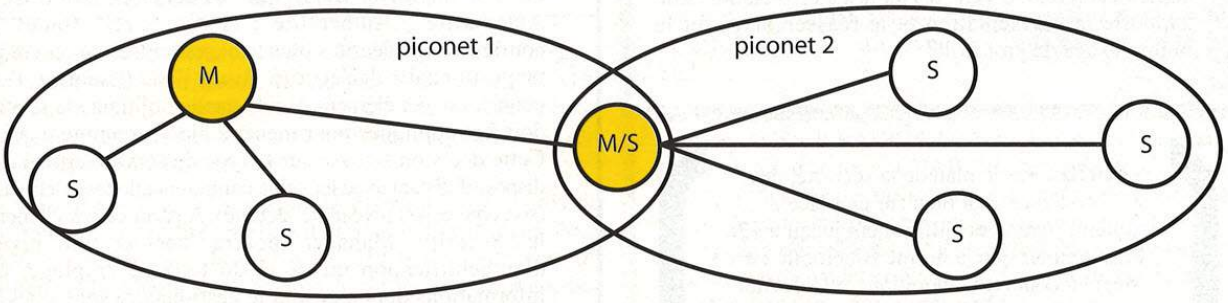
lors des communications par IP, encadrés de blocs de données de contrôle. En fonction de l'encombrement de la bande passante, émetteurs et récepteurs peuvent changer de canal jusqu'à 1600 fois par seconde. En plus de l'adresse du destinataire, les blocs de contrôle contiennent donc la fréquence sur laquelle se trouvera le paquet suivant. Ce besoin d'un contrôle permanent explique la différence entre le débit théorique de 1 Mbits/s et le débit pratique qui est au mieux de 2 x 432 = 864 Kbits/s. Une norme Bluetooth 2 est en cours de spécification : elle devrait permettre d'atteindre un débit théorique de 2 à 10 Mbits/s. En 2002, Ericsson estime à plus de 100 millions le nombre d'appareils équipés de cette puce. Cette dernière coûte actuellement 20□, mais Intel prévoit qu'elle ne coûtera bientôt plus que 5□. Enfin, et cela explique le coût encore élevé des puces, il existe 20 profils différents et trois types de liaisons. Les profils régissent le comportement des périphériques Bluetooth, les liaisons désignent les modes de transmission utilisés entre les appareils communicants. Le groupe de travail IEEE 802.15.1 travaille actuellement sur la normalisation de Bluetooth avec une norme permettant la compatibilité (802.15.1) entre Bluetooth 1.1 et le 802.11. Mais 802.15.1 est tellement compatible qu'elle remet même en question l'intérêt et l'avenir de Bluetooth. Le marché jugera si ces normes sont complémentaires ou concurrentes.

Architecture des protocoles



Couche physique : radio

Au niveau physique, Bluetooth utilise la technologie par saut de fréquence (FHSS) sur 79 canaux dans la bande 2,402 à 2,480 GHz. Le réseau fonctionne en mode maître/esclave, et le maître décide des sauts de fréquence de façon pseudo-aléatoire, 1600 fois par seconde. La norme 1.0A définie en juillet 1999 prévoyait un débit brut de 1 Mbps (soit 720 kbps). Nul doute que ce débit sera amené à augmenter par la suite, même si la version 1.1, approuvée en mars 2001, ne le prévoit pas encore. Les machines d'un réseau Bluetooth se rassemblent en sous-réseaux appelés *piconets*. Dans ce piconet, une des machines joue le rôle de maître, et gère à ce titre l'horloge et les sauts de fréquence. Chaque maître peut accueillir jusqu'à 7 esclaves actifs, soit 8 appareils actifs maximums par piconet. Plusieurs piconets adjacents constituent un *scatternet* et peuvent interagir. Une machine peut ainsi être esclave d'un piconet et maître d'un autre. Chaque piconet dispose d'un débit de 1 Mbps. 10 scatternets peuvent ainsi interagir, soit 72 appareils maximums (8 x 10 – 8 appareils).



Architecture des protocoles (suite)

Baseband layer

Cette couche permet de définir trois types de liens :

- les liaisons SCO (*Synchronous Connection-Oriented*) pour l'audio (ou audio et données) ;
- les liaisons ACL (*Asynchronous Connectionless*) pour les données. Dans le cas où les débits montants et descendants ne seraient pas égaux, les liaisons ACL peuvent être asymétriques ;
- les liaisons de base, pour toute la gestion des connexions au sein du piconet. Les paquets ont alors la forme suivante :

Code d'accès	En-tête	Données
72 bits	54 bits	0 à 2745 bits

Le code d'accès permet la synchronisation des composants Bluetooth. L'en-tête stocke le numéro de paquet, l'adresse source et destination, le type de paquet, le CRC…

Link manager protocol

Ce protocole est responsable de la supervision des différentes connexions, de l'authentification des appareils, et du chiffrement. Il gère également les mises en veille des différents appareils.

Link Layer Control & Adaptation (L2CAP)

Cette couche permet l'adaptation des protocoles supérieurs (comme TCP/IP) au réseau Bluetooth : elle supporte la segmentation et le réassemblage, et le multiplexage de protocole.

Nota : Sur le plan de la sécurité, des systèmes sont bien sûr en place : authentification et chiffrement jusqu'à 128 bits. A noter que la norme Bluetooth 2 est déjà en cours d'élaboration : elle devrait étendre la portée à une centaine de mètres et autoriser des débits de 10 Mbits/s.

BLUETOOTH, SÉCURITÉ = WEP SURCLASSÉ !

Bluetooth étant issu du monde de la téléphonie, son mode de sécurisation a été calqué sur celui des téléphones sans fil numériques DECT (*Digital European Cordless Telecommunications* – norme de téléphonie sans fil de proximité répandue surtout à travers l'Europe qui s'appuie sur la technique de transmission d'accès multiple à répartition dans le temps [AMRT]). Il s'agit d'une sécurisation point à point par échange de clés privées.

Par rapport à 802.11, Bluetooth présente des avantages certains en termes de sécurité et repose sur trois éléments :

- une adresse permanente et unique inscrite physiquement sur la carte Bluetooth (sur le modèle de l'adresse MAC des cartes Ethernet) ;
- un code personnel d'identification à 4 chiffres ;
- un dispositif de génération de nombres aléatoires sur 128 bits.

Ces éléments permettent de générer des clés de cryptage et d'authentification à 128 bits selon trois modes :

- mode 1, non sécurisé -> aucune fonction de sécurité n'est activée, tous les appareils Bluetooth peuvent se connecter librement ;
- mode 2, sécurisation au niveau de l'application -> la connexion des appareils est libre, le contrôle s'effectue au niveau applicatif ;
- mode 3, sécurisation au niveau de la connexion -> c'est le mode le plus sûr : il hérite des fonctionnalités du mode 2 en y ajoutant un contrôle préalable au moment de la tentative de connexion

En plus de ces trois modes de sécurité, Bluetooth offre deux niveaux de sécurité pour les dispositifs physiques (fiable ou non fiable) et trois pour les services (Autorisé et Authentifié ; Authentifié ; Accès libre). Toutes les configurations décrites plus haut sont mises en œuvre par un gestionnaire de sécurité. Avant l'établissement d'une connexion, cet élément décide quelle politique de sécurité doit être appliquée pour mener à bien la communication. Cette décision se base sur le type de service utilisé et le dispositif distant avec lequel la communication va s'effectuer (son type et son niveau de sécurité). À partir de ces éléments, le Security Manager pourra décider du niveau d'authentification utilisé et du type de cryptage. Les informations dont a besoin le gestionnaire sont stockées dans deux bases de données : la base de données des dispositifs physiques (qui stocke le type de dispositif, son

niveau de fiabilité et la taille de ces clés de cryptage) et celle des services (où l'on trouvera les informations relatives au niveau de sécurité service et au mode d'acheminement de l'information). La principale faiblesse de la sécurité Bluetooth à l'heure actuelle réside dans l'insuffisance du code personnel à 4 chiffres. Une autre faille, inhérente celle-là à tous les appareils mobiles, réside dans le déni de services. En effet, les appareils Bluetooth fonctionnant sur batterie, une forme d'attaque peut consister à surcharger de travail un dispositif ciblé de manière à épuiser ses batteries. Les concepteurs de Bluetooth travaillent actuellement sur des parades à ces limitations.

BLUETOOTH ET 802.11

Le 802.11b offre aujourd'hui une bande passante cinq fois plus importante ainsi que des portées plus élevées que Bluetooth, ce qui rend cette technologie plus indiquée dans une utilisation en réseau (les gros réseaux locaux d'entreprise et les environnements informatiques centralisés). Mais cet avantage a un coût : plus puissants, les émetteurs récepteurs 802.11b sont d'une taille plus importante et consomment beaucoup plus que leurs homologues Bluetooth. WiFi n'est donc pas une solution adaptée pour les outils nomades (PDA et autres GSM…) Si Bluetooth est limité en débit et en distance de propagation, cette technologie offre l'avantage d'une faible consommation d'énergie grâce à la compacité de la puce. De plus, le prix de cette solution est moins élevé. Aussi Bluetooth se révèle intéressant pour les produits nomades grand public qui ont l'inconvénient d'avoir une consommation d'énergie élevée. Il est évident que pour interconnecter sans fil un certain nombre de PC entre eux et à Internet, la technologie à utiliser est WiFi. En effet, 802.11b est une extension naturelle de Ethernet 10 Base T, et assure un débit qui lui correspond. Mais pour connecter des outils nomades, entre eux et à Internet, la technologie à utiliser est Bluetooth. En effet, Bluetooth consomme très peu de courant et permet donc de respecter l'autonomie du matériel sur lequel elle est implantée. Les outils nomades, pour la plupart, ont besoin de supporter de l'audio. Or, Ethernet et TCP/IP ne savent pas gérer la communication de la voix aussi bien que Bluetooth.

L'AVENIR DE LA SÉCURITÉ EN WIFI

L'IEEE a choisi le protocole WEP (*Wired Equivalent Privacy*) pour assurer la sécurisation des normes 802.11a et 802.11b. Contrairement à Bluetooth, le WEP fonctionne avec des clés de chiffrement 64 bits (les clés à 128 bits sont optionnelles) qui peuvent être facilement cassées.

De plus, il n'y a ni mécanisme de distribution des clés de chiffrement, ni véritable authentification de l'utilisateur. Pour améliorer la sécurité des WLAN, l'IEEE s'oriente vers un chiffrement réutilisant les techniques de chiffrement de protocoles point à point. De nouveaux protocoles sont apparus, parmi lesquels on distingue principalement :

WPA (WIFI PROTECTED ACCESS)

La Wi-Fi Alliance - qui regroupe notamment 3Com, Lucent Technologies, Nokia, Sony, Compaq ou encore Siemens – a récemment annoncé une alternative au WEP : le WPA. La norme WPA emploie le protocole TKIP (*Temporal Key Integrity Protocol*), qui consiste à fournir de nouvelles clés pour chaque paquet de 10 Ko de données transmises sur le réseau. WEP, de son côté, n'employait que des clés fixes. Toutefois, WPA utilise le même algorithme que le WEP. Les premières solutions intégrant cette norme de sécurité devraient apparaître dès cette année. Cette solution semble, néanmoins, n'être qu'une réponse temporaire en attendant l'arrivée de la norme 802.11i. Ce standard, encore en développement, intégrera d'emblée une solution sécurisée, sans ajout logiciel, tout en étant compatible avec ses prédécesseurs.

EAP (EXTENSIBLE AUTHENTICATION PROTOCOL)

Il existe également une nouvelle combinaison 802.1x/EAP qui permet de gérer et de distribuer des clés de cryptage par utilisateur et par session. Étant tous deux par nature des standards ouverts, leur implémentation est laissée à la discrétion des divers fournisseurs. En conséquence, quatre types d'implémentation EAP font figure de "standards". Ils reposent tous les quatre sur la même architecture 802.1x et EAP sous-jacente, mais se distinguent par la manière dont ils mettent en œuvre l'EAP.

LEAP

Cisco a été l'un des premiers fournisseurs à proposer, en décembre 2000, un produit doté de son "standard" LEAP (*Lightweight EAP*). Il s'agit d'une solution propriétaire qui ne fonctionnait initialement qu'avec les cartes Cisco clients 802.11, les serveurs RADIUS et les points d'accès Cisco. L'équipementier s'est récemment associé à d'autres fournisseurs pour assurer la compatibilité avec LEAP des équipements et logiciels. L'éventail de cartes PC clients 802.11 est désormais plus large, et au moins quatre autres solutions RADIUS prennent en charge LEAP. Certains fabricants d'ordinateurs portables supportent même cette solution en mode natif avec leurs adaptateurs intégrés 802.11.

L'implémentation de LEAP est relativement simple. Des serveurs ACS/AR RADIUS de Cisco peuvent être aisément reliés à un domaine LDAP ou Active Directory et l'authentification des utilisateurs est transparente. Le seul revers de cette approche est que les règles de mot de passe doivent être rigides, car LEAP est vulnérable à des attaques de dictionnaire par interception. Si les règles sont adéquates, LEAP est une solution plutôt pratique et sûre.

EAP-TLS (Transport Layer Security)

C'est un standard ouvert pris en charge par la quasi-totalité des fournisseurs. Basé sur EAP, sa puissance tient au fait qu'il exige l'utilisation de l'infrastructure à clé publique PKI. PKI fait d'EAP-TLS un standard extrêmement sûr puisque des clés privées et publiques asymétriques sont utilisées sur les clients et RADIUS. Seul inconvénient, l'implémentation d'une PKI peut sembler assez délicate. Microsoft en est un fervent adepte ; il a doté Windows XP d'un support client natif pour EAP-TLS. L'éditeur propose une prise en charge spécifique pour Windows 2000, NT, 98 et Pocket PC. Pour l'heure, il faut recourir à une solution tierce, telle que celle de Meetinghouse Data Communications (MDC), pour les systèmes d'exploitation autres que XP. Cisco recommande dorénavant de prendre en charge à la fois LEAP et EAP-TLS. EAP-TLS est un bon choix avec la version 3 de Cisco ACS RADIUS, car Cisco s'est rendu compte que tous les produits ne sont pas compatibles avec LEAP. Le coût de mise en œuvre d'EAP-TLS est pour ainsi dire négligeable si l'on utilise Microsoft RADIUS et la technologie PKI. En effet, RADIUS Internet Authentication Service (IAS) de Microsoft est intégré au système d'exploitation Windows 2000 Server et constitue une solution aussi stable que les autres. Comme Microsoft recommande d'implémenter IAS sur les contrôleurs de domaine, il n'y aucun coût additionnel lié à un nouveau serveur ou des licences supplémentaires. Pour disposer de l'infrastructure PKI requise, on peut mettre en œuvre le service Certificate Authority, qui est également intégré à Windows 2000 Server. La mise en place d'une infrastructure PKI dans l'entreprise est la seule opération à effectuer; en n'oubliant pas, toutefois, que les certificats PKI peuvent être utilisés à d'autres fins, par exemple avec un réseau privé virtuel (VPN) L2TP. Cette opération ne doit être réalisée qu'une seule fois.

EAP-MD5

C'est la version la moins sûre d'EAP, car si elle gère les noms d'utilisateurs et les mots de passe pour l'authentification, elle ne demeure pas moins vulnérable aux attaques par dictionnaire. De plus, cette version ne prend pas en charge les clés WEP dynamiques, ce qui constitue un défaut majeur.

EAP-TTLS (Tunneled Transport Layer Security)

Cette version EAP est signée par l'éditeur Funk. Elle est prise en charge sur ses produits Odyssey ou Steel-Belted RADIUS Server, ainsi que sur des logiciels clients tiers, tels que ceux proposés par MDC. L'argument commercial de Funk est que les certificats PKI ne sont nécessaires que sur les serveurs d'authentification, non sur les postes clients. Cette solution peut être considérée comme presque aussi sûre qu'EAP-TLS, tout en simplifiant le déploiement.

Les solutions de réseaux locaux sans fil sont aujourd'hui prêtes, tant en entreprise que pour le grand public. L'année 2002 aura été véritablement marquée par un début d'engouement et une croissance significative des ventes, WiFi en tête. Néanmoins, à l'heure où le secteur informatique traverse l'une de ses plus graves crises, le formidable espoir que représentent les technologies sans fil est tempéré par la faiblesse des mécanismes de sécurité des données transmises. Tant que le réseau est câblé, peu d'inquiétudes surgissent, mais dès que celui-ci passe par les airs, une certaine anxiété arrive. Après tout, généralement, on peut mieux surveiller quelque chose d'observable. Dès lors, d'importantes questions surgissent : quel est le niveau de sécurité d'un tel système ? Quelles sont les vulnérabilités induites dans mon système d'information par la mise en œuvre de telle ou telle norme ? Mes données sont-elles correctement protégées ? A l'origine, le standard définit un mécanisme par lequel le WEP peut être atteint. Ce protocole se fonde sur un chiffrement RC4 de 40 bits. Le WEP donc mis en œuvre, alors toutes les données transmises sont cryptées. Toutefois, une étude de l'université de Californie à Berkeley a démontré qu'il existait une faille de sécurité pouvant compromettre la confidentialité des données transmises sur le WLAN. On peut noter que la mise en œuvre de cette faille est simple et que la norme n'avait pas pour but de faire de ce type de réseau un système point à point complètement sécurisé. Comme nous avons pu le voir précédemment, une norme supérieure d'encodage des données est en train d'être mise en place pour remédier à ce problème. Concoctée

Références

http://www.bluetooth.com/
http://www.homerf.org/
http://www.etsi.org/technicalactiv/hiperlan1.htm
http://www.hiperlan2.com
http://www.ieee.org
http://www.tomshardware.fr
http://www.rd.francetelecom.fr/fr/technologies/ddm200207/index1.php

HiperLAN ou 802.11a : quelle norme pour les réseaux sans fil en Europe ? Peter Judge- ZDNet US.

Right Technologies Choices – First Quarter 2002 (White Paper) (**www.cysive.com**)

An Initial Security Analysis of the IEEE 802.1x Standard, Arunesh Mishra & William A. Arbaugh, Department of Computer Science, University of Maryland

Conclusion

par la WiFi Alliance – qui regroupe notamment 3Com, Lucent Technologies, Nokia, Sony, Compaq ou encore Siemens – la norme de sécurité WPA (*WiFi Protected Access*), devrait à terme remplacer le WEP. L'IEEE, déjà chargée des standards 802.11b et 802.11a, travaille en parallèle sur un nouveau standard 802.11 appelé 802.11i (ou EAP). Ce protocole permet de gérer et de distribuer des clés de cryptage par utilisateur et par session. Pour finir, la plupart des équipements réseau sont livrés avec des fonctions de sécurité élémentaires, au standard WEP. Par défaut, les ports IrDA et Bluetooth sont configurés pour se connecter automatiquement aux périphériques avoisinants, sans demander un mot de passe ou une authentification. Mais beaucoup d'entreprises ne les activent pas ou ne changent pas les paramètres ou les mots de passe par défaut. Un pirate, même débutant, peut ainsi aisément se connecter sur le réseau. La sécurité peut également être complètement remise en cause si des employés installent des points d'accès sans prévenir le service informatique. Ces deux points (paramètres par défaut des matériels, sensibilisation des utilisateurs) font partie des premiers réflexes que responsables informatiques et administrateurs doivent acquérir.

Eric Hazane / XP Conseil

ehazane@xpconseil.com
http://ehazane.free.fr

Communication ou la naissance

L'idée de départ de cet article était de vous montrer plusieurs méthodes pour créer des canaux de communication entre l'espace utilisateur et l'espace noyau. Il s'est vite avéré au cours des développements de nos modules que nous étions limités par les outils disponibles pour les tester (récupération d'informations du noyau, dissimulation de celles-ci..), mais aussi par les fonctionnalités de nos modules eux-mêmes. Nous avons donc décidé de développer un outil (kernsh) pour pallier au problème.

INTRODCTION

Une description plus précise sera faite par la suite mais vous pouvez d'ores et déjà trouver les sources de ce programme à l'URL suivante :
http://www.kernsh.org/kernsh-0.2.tgz.

Tous les codes sources qui suivent sont inclus dans l'archive.

L'article se décompose donc en deux parties. La première décrit de nouvelles méthodes de communication entre l'espace utilisateur et l'espace noyau. Nous expliquons auparavant les différents niveaux de privilèges implémentés par le processeur et utilisés par les systèmes d'exploitation : les *ringlevels*. Nous verrons aussi comment ils communiquent entre eux pour en tirer un bénéfice à l'aide des modules noyau. Dans la seconde partie, nous illustrons ces principes et les exploitons davantage à l'aide de *kernsh*.

A noter aussi que nous travaillons uniquement sur Linux processeur i386, mais cela peut certainement s'appliquer à d'autres architectures.

L'ESPACE D'ADRESSAGE D'UN PROCESSUS ET SES PERMISSIONS

Chaque processus dispose d'un espace d'adressage virtuel géré par les structures *mm_struct* (`linux/sched.h`) et *vm_area_struct* (`linux/mm.h`) qui se trouvent au sein même d'une autre structure, la *task_struct* (`linux/sched.h`). A chaque exécution d'un processus sont associées, entre autres, des zones de code et de données, et une pile. Nous parlons alors de segments mémoire. Le système de fichiers virtuel `/proc` nous donne des informations précises au sujet de ces segments (cf le fichier `/proc/<pid>/maps` ou réf. [2]).

Par quel procédé le noyau alloue cet espace mémoire et de quelle manière y sont gérés les privilèges, c'est ce que nous allons voir maintenant.

LES NIVEAUX DE PRIVILEGES

D'après la documentation INTEL, les mécanismes de protection du processeur connaissent quatre niveaux de privilèges, ou *ringlevels*, de 0 à 3 inclus, 0 étant le privilège le plus élevé, et 3 le moins élevé. Le *Current Privilege Level* (ou CPL) quant à lui désigne le niveau de privilège du segment de code qui contient le code de cette tâche, c'est-à-dire le niveau de privilège du code en cours d'exécution.

LA SEGMENTATION ET LA PAGINATION

L'adressage mémoire sous Linux fonctionne selon deux dispositifs matériels : la segmentation et la pagination. La segmentation transforme les adresses logiques en adresses linéaires et la pagination les adresses linéaires en adresses physiques. Nous ne détaillerons pas ces deux dispositifs (cf réf. [1]), seule la pagination qui organise les espaces mémoire des processus nous intéresse pour la suite de l'article. Tous les systèmes d'exploitation utilisant le mode protégé sur processeur x86 se comportent de la même manière.
Il faut savoir que 2 niveaux de privilèges seulement sont implémentés pour les pages et tables de pages. Ce sont le mode

userland/kernelland de kernsh

utilisateur et le mode noyau, contrôlés par le bit User/Supervisor. En mode noyau, il est à 0 (avec CPL < 3), et en mode utilisateur, il est à 1 (avec CPL=3). Si nous voulons utiliser les fonctionnalités du noyau, il faut arriver à faire exécuter du code pendant que le processeur est en ringlevel 0.

Ce niveau de privilège est représenté par le champ _PAGE_USER dans le fichier asm/pgtable.h :

```
/*
 * The 4MB page is guessing..  Detailed in the infamous "Chapter H"
 * of the Pentium details, but assuming intel did the straightforward
 * thing, this bit set in the page directory entry just means that
 * the page directory entry points directly to a 4MB-aligned block of
 * memory.
 */
[...]
#define _PAGE_PRESENT   0x001
#define _PAGE_RW    0x002
#define _PAGE_USER  0x004
[...]
```

Pour que ce soit plus clair, voici un récapitulatif des valeurs du bit U/S et de CPL en fonction du mode d'exécution :

- en mode noyau : bit U/S = 0 et CPL < 3 ;
- en mode utilisateur : bit U/S = 1 et CPL = 3.

LES ESPACES D'ADRESSAGE EN MODE UTILISATEUR/NOYAU

Lorsqu'un processus est en mode utilisateur, il ne peut accéder aux structures de données et au code du noyau du fait que le ringlevel ne correspond pas au CPL (ringlevel = 1 et CPL < 3). Si un tel accès mémoire est tenté, le processeur échoue et appelle le gestionnaire de l'exception de défaut de page ou page_fault. La mémoire du noyau est en fait "mappée" dans l'espace mémoire de chaque processus, mais le niveau de privilège du processeur empêche d'y accéder. En revanche, si le CPL est supérieur au ringlevel, c'est-à-dire si nous sommes en mode noyau (ringlevel = 0) et que nous souhaitons accéder à du code utilisateur (CPL = 3), l'accès mémoire est autorisé. Le noyau doit en effet avoir accès à l'ensemble des pages, par exemple pour modifier dynamiquement l'espace d'adressage d'un processus lorsque celui-ci requiert plus ou moins de mémoire. Il est donc obligé de connaître les régions de mémoire attribuées à un processus et d'avoir le contrôle total sur celles-ci.

Note : Une ancienne version du patch de sécurité PaX, met volontairement le drapeau *supervisor* sur les pages qu'il veut protéger. Le processeur échoue systématiquement lors des accès à ces pages, et appelle page_fault, qui est spécialement modifiée pour traiter ces différents cas et autoriser ou non l'accès à la page.

PASSER DU MODE UTILISATEUR AU MODE KERNEL

Comment un processus peut-il donc passer du mode utilisateur au mode kernel et vice-versa ? Le noyau est le cœur du système. Il est donc évident qu'il puisse être sollicité par le mode utilisateur pour effectuer une tâche. Il le sera d'au moins deux façons :

- lorsqu'un programme invoque un appel système ;
- lorsqu'une interruption (interruption asynchrone) ou une exception (interruption synchrone) intervient.

Les interruptions sont générées par des périphériques et les exceptions par le noyau lors d'une erreur de programmation par exemple. Par conséquent, nous nous intéresserons seulement aux appels systèmes car c'est la seule manière de contrôler ce que nous passons au noyau.

Note : Ne confondez pas IRQ et interruption. La première signifie *Interrupt ReQuest* ou Requête d'Interruption et est générée par un périphérique. La seconde est plutôt considérée comme l'évènement associé à l'IRQ. Là encore, il s'agit de fonctionnalité du processeur, indépendante du système d'exploitation.

La table de descripteurs d'interruption ou IDT

Même si nous avons dit à l'instant que seuls les appels systèmes nous intéressent, présentons succinctement la table des interruptions.

Chaque interruption est identifiée par un numéro ou vecteur entre 0 et 255. Elles sont répertoriées en trois catégories, dont la catégorie des interruptions logicielles (du vecteur 48 à 255), chacune possédant son propre descripteur d'interruptions associé (codé sur 8 octets).

Ces descripteurs sont dans la table des descripteurs d'interruptions (IDT). Le descripteur d'interruptions précise le type d'interruption, et entre autres le DPL (*Descriptor Privilege Level*). Le DPL correspond au ringlevel nécessaire pour appeler l'interruption.

L'IDT est initialisée au lancement du système par la fonction `trap_init()` (`arch/i386/kernel/traps.c`). Le processeur ayant besoin de connaître l'adresse de l'IDT, celle-ci se trouve dans le registre `idtr`, et peut être lue/écrite grâce à l'instruction `lidt`.

Le vecteur 128 ou interruption $0x80

La fonction `set_system_gate()` (toujours dans le fichier `arch/i386/kernel/traps.c`) met en place le descripteur d'interruptions pour une interruption logicielle, avec le DPL = 3. Cette fonction est appelée quatre fois dans `trap_init`, ce qui signifie qu'il y a quatre interruptions de ce type.

On remarque notamment dans le fichier `traps.c` :

```
[...]
set_system_gate(SYSCALL_VECTOR, &system_call);
[...]
avec SYSCALL_VECTOR = 0x80 (arch/i386/kernel/irq.h)
```

Ce qui signifie que l'interruption 128 (0x80) est une interruption logicielle que nous pouvons appeler depuis le mode utilisateur et dont le gestionnaire d'interruptions est la fonction `system_call()`.

L'interruption 0x80 peut être levée en exécutant l'instruction `int $0x80` (`int` est l'instruction émettant un signal d'interruption). Le processeur lit le descripteur correspondant à l'entrée 0x80 dans la table des descripteurs d'interruption.

L'instruction `int` permet d'invoquer les appels systèmes en mode utilisateur. Pour identifier l'appel système, un numéro est passé en tant que paramètre via le registre `%eax`. Ces numéros représentent l'index du tableau de pointeur sur fonctions `sys_call_table[]` et peuvent être retrouvés dans le fichier `/usr/include/asm/unistd.h`.

Si nous souhaitons aussi passer des paramètres aux appels systèmes, nous le faisons par l'intermédiaire des registres `%ebx`, `%ecx`, `%edx`, `%esi`, `%edi` et `%ebp`, soit 6 paramètres au plus. Il faut aussi faire attention à ce que la taille d'un paramètre ne dépasse pas celle d'un registre, soit 32 bits (sur une architecture x86 de 32 bits). Notez que sous *BSD, les paramètres sont passés sur la pile.

Une fois l'interruption 0x80 levée, le contrôle est donné au noyau (nous sommes en mode noyau) et plus précisément à la fonction `system_call()` (`arch/i386/kernel/entry.S`) qui traite l'appel système en exécutant la fonction qui lui est associée. La fonction `ret_from_sys_call()` est ensuite appelée et nous retournons en mode utilisateur.

La fonction system_call()

Cf `arch/i386/kernel/entry.S` :

```
[...]
#define SAVE_ALL \
        cld; \
        pushl %es; \
        pushl %ds; \
        pushl %eax; \
        pushl %ebp; \
        pushl %edi; \
        pushl %esi; \
        pushl %edx; \
        pushl %ecx; \
        pushl %ebx; \
        movl $(__KERNEL_DS),%edx; \
        movl %dx,%ds; \
        movl %dx,%es;
[...]


ENTRY(system_call)
        pushl %eax          # save orig_eax                    [1]
        SAVE_ALL
                                                               [2]
        GET_CURRENT(%ebx)
        cmpl $(NR_syscalls),%eax                               [3]
        jae badsys

        testb $0x20,flags(%ebx)     # PF_TRACESYS
        jne tracesys
        call *SYMBOL_NAME(sys_call_table)(,%eax,4)             [4]
        movl %eax,EAX(%esp)     # save the return value
        ALIGN
        .globl ret_from_sys_call
        .globl ret_from_intr
ret_from_sys_call:
        movl SYMBOL_NAME(bh_mask),%eax
        andl SYMBOL_NAME(bh_active),%eax
        jne handle_bottom_half
ret_with_reschedule:
        cmpl $0,need_resched(%ebx)
        jne reschedule
        cmpl $0,sigpending(%ebx)
        jne signal_return
```

```
restore_all:
        RESTORE_ALL
[...]
badsys:

        movl $-ENOSYS,EAX(%esp)
        jmp ret_from_sys_call
```

■ A la ligne **[1]**, la fonction `system_call()` sauvegarde sur la pile `%eax`, soit le numéro d'appel système. Chaque processus a en réalité deux piles différentes afin qu'elles n'interfèrent pas entre elles : la pile noyau et la pile utilisateur. Elles sont respectivement utilisées lorsque nous sommes en mode noyau ou en mode utilisateur. Dans le cas de cette fonction, étant donné qu'une interruption a été levée, c'est sur la pile noyau que `%eax` est sauvegardé.

■ La macro `SAVE_ALL` **[2]** s'occupe entre autres de sauvegarder sur la pile noyau les autres registres contenant les paramètres de l'appel système.

■ La valeur du numéro de l'appel système est vérifiée **[3]**. Si elle est supérieure ou égale à `NR_syscalls`, la fonction `system_call()` se termine en renvoyant le code d'erreur `-ENOSYS` (en résumant un peu la chose :).

■ La fonction correspondant au numéro d'appel système `%eax` est ensuite appelée **[4]**. Cette fonction est la nième entrée (où 'n' est donc la valeur de `%eax`) du tableau de pointeur sur fonction `sys_call_table[]`.

EXEMPLE

Un exemple valant mieux qu'un long discours, voici un programme simple qui utilise l'appel système `sys_open()` et qui permet d'ouvrir le fichier `/boot/System.map` :

```
$ /bin/cat sys_open.c
#include >linux/unistd.h>
#include >sys/types.h>
#include >sys/stat.h>
#include >fcntl.h>

#define FILENAME "/boot/System.map"

int
main( void )
{
        long __res;

        __asm__ volatile ("int $0x80"
                : "=a" (__res)
                : "0" (__NR_open), "b" (FILENAME), "c" (O_RDONLY) );
}
$ make sys_open
cc     sys_open.c   -o sys_open
$ strace ./sys_open
[...]
munmap(0x40014000, 14874)               = 0
getpid()                                = 354
open("/boot/System.map", O_RDONLY)      = 3

_exit(3)                                = ?
```

Nous avons ici déclenché l'appel système `sys_open()` qui ouvre le fichier `/boot/System.map` avec succès à en juger par la valeur de retour. Si vous avez bien suivi jusque-là, vous savez que nous ne pouvons pas lire la mémoire noyau à cause du CPL qui diffère. En revanche, nous pouvons appeler des fonctions qui s'exécuteront en mode noyau. Peut-être en existe-t-il une qui lise la mémoire et nous la renvoie ? Nous pourrions ainsi lire la mémoire noyau du mode utilisateur.

LES MODULES

Linux supporte un système de modules, c'est-à-dire qu'une partie du noyau peut être rajoutée alors que celui-ci est déjà lancé, et ce sans redémarrer le système. Ce mécanisme est souvent utilisé pour charger des pilotes de périphériques. Mais cela signifie qu'il est possible depuis le mode utilisateur de faire exécuter du code en mode noyau.

Les modules utilisés en tant que *rootkit* permettent donc notamment de :

- rendre invisible des fichiers, comme ceux produits par un *sniffer* ;
- filtrer le contenu d'un fichier (supprimer son IP des logs, les numéros de ses processus...) ;
- sortir de prison (*chroot*) ;
- dissimuler l'état du système (mode *promiscuous*) ;
- dissimuler des processus ;
- sniffer ;
- ménager des *backdoors*
- durcir les appels système...

Cf référence **[3]** pour plus détails.

Bref tout ce que votre imagination et vos connaissances vous permettent de faire, aussi bien pour un pirate que pour un administrateur. Ils nécessitent cependant beaucoup de développement pour faire effectuer au noyau ce que nous voulons : détournement de nombreux appels système, passage de paramètres difficile entre le mode utilisateur et le mode noyau... Et modifier le fonctionnement du système peut s'avérer fatal (beaucoup le savent :).

Nous allons donc essayer de mettre en place des moyens de communication entre le mode noyau et utilisateur. Certes, les exemples qui suivent implémentent des fonctionnalités qui existent déjà dans les rootkits actuels (obtenir les privilèges root par exemple). Mais le but ici est plutôt de vous démontrer qu'il existe plusieurs moyens (des fois plus pratiques) de faire exécuter du code au noyau en tant qu'utilisateur.

SIMULATION D'UN DEBORDEMENT DE BUFFER DANS LE NOYAU

L'idée principale est d'ordonner au système de faire ce que nous voulons avec un minimum de code dans l'espace noyau.

Nous créons donc un module implémentant un appel système qui contient un débordement de *buffer* en modifiant une entrée de la `sys_call_table[]` simplement de cette manière :

```
#define SYS_BOF 250
[...]
        sys_call_table[ SYS_BOF ] = ( void * )overflow;
[...]
```

Un programme client charge alors un *shellcode* en mémoire (espace utilisateur) et appelle le `syscall` du module. La fonction `overflow()` est appelée et par conséquent, le *buffer overflow* se déclenche et écrase l'adresse de retour de la fonction associée au `syscall` avec l'adresse du shellcode situé dans l'espace utilisateur. Pourquoi le shellcode se lancerait-il ? Puisque, au moment du débordement, nous sommes en mode noyau, il n'y a aucun problème pour revenir dans l'espace utilisateur et accéder à ce que nous voulons.

Comme nous l'avons spécifié au début de l'article, les sources des modules sont inclus avec les sources de *kernsh* (dans le répertoire `modules/`). Vous y trouverez aussi le script `bd_of_modules.sh` qui gère une pseudo base de données de modules. Celui en question est `buffer_overflow_lkm_1.c` et son "client" : `client_buffer_overflow.c`.

Note : Les tests sont effectués sur une Debian 3.0 avec un noyau 2.2.19 et 2.4.19.

Après avoir compilé et chargé le module (`insmod`) en tant que root, nous lançons le client `client_buffer_overflow` en tant que simple utilisateur. Nous sommes alors en ringlevel 3. L'interruption 0x80 déclenche l'appel système 250, nous passons en ringlevel 0 et la fonction `overflow()` du module s'exécute en mode noyau, avec les privilèges de ce ringlevel.

```
$ ./client_buffer_overflow
[+] shellcode: 0x80494e0..
sh-2.05a# id
uid=0(root) gid=0(root) groups=1000(blaat)
sh-2.05a# exit
exit
$
```

Nous obtenons un shell avec les privilèges root car nous nous sommes arrangés pour que le module, avant qu'il ouvre le shell, change les droits du processus courant.

Nous aurions tout aussi bien pu placer le shellcode dans le module. Mais le mettre dans la partie cliente permet déjà d'avoir un module beaucoup plus léger, et surtout de contrôler plus facilement le code à exécuter. Autre point qui peut être important pour certains, Openwall ne détecte rien car il *patche* la pile utilisateur et non la pile noyau, tout comme PaX, car le "page fault" ne se déclenche pas puisque nous sommes en mode noyau (CPL=0).

A L'AIDE D'UNE STRUCTURE

L'idée est toujours de passer du code à exécuter dans l'espace noyau en appelant un appel système que nous avons pris soin de créer. Mais au lieu de simuler un débordement dans l'espace noyau, nous lui passons une structure qui contient l'adresse de la fonction à exécuter et les arguments qui vont bien. Nous lui indiquons le type de chaque argument (`char * fmt`), et les paramètres dans différents tableaux de la structure. Le module n'a plus qu'à récupérer la structure passée en paramètre, mettre les arguments comme il convient sur la pile et appeler la(les) fonction(s) :>

Les sources du module sont au même endroit. Le module est `struct_lkm_1.c` et le client est `client_struct.c`.

Dans l'exemple fourni, nous n'appelons que la fonction `printk()`. Nous trouvons son adresse dans le fichier `System.map` généré lors de la compilation du noyau (il est tout à fait possible de calculer l'adresse de `printk()` automatiquement mais cela n'est pas le sujet de l'article). Ce fichier contient les symboles du noyau et leurs adresses.

Le principe pour exécuter du code en mode noyau est le même que dans l'exemple précédent. Nous créons un appel système que nous déclenchons avec l'interruption $0x80 pour passer du ringlevel 3 au ringlevel 0. Simplement, au lieu d'utiliser un shellcode au moment où nous sommes dans le ringlevel privilégié, nous utilisons une structure.

Notez que nous pouvons cumuler les appels à différentes fonctions en les enchaînant grâce au pointeur `next` de la structure, comme dans une banale liste chaînée.

Essayons :

```
$ ./client_struct
[!] client_struct launched !! You have to be sure that
[!] module is inserted and that syscall 250 is redirect on the
[!] function our_handler of struct_lkm !
clienstruct% call printk("toto %d\n", 42)
toto 42
clienstruct% quit
$ /bin/dmesg
[...]
[+] loaded..
toto 42
```

Ça a marché ! Ici, il n'y avait pas d'intérêt particulier à faire exécuter un `printk()`, mais c'était simplement pour illustrer le principe. Nous aurions pu par exemple faire appel à `do_execve()` et lancer le programme que nous voulions ou encore *kmalloc()er* de la mémoire.

LA NAISSANCE DE KERNSH

Nous souhaitons maintenant faire évoluer nos modules et nos méthodes de communication entre l'espace utilisateur et l'espace noyau. Par exemple, en se mettant à la place d'un pirate, il faudrait qu'ils ne soient pas visibles dans la liste des modules installés (`/sbin/lsmod`). Autre exemple, nous pourrions, plutôt que d'utiliser l'interruption $0x80 détourner une quelconque interruption pour la faire pointer sur la fonction `init_module()` du module `buffer_overflow_lkm_1.o`, déclencher cette interruption et la restaurer. Le code du module serait alors exécuté.

Cette seconde partie montre un exemple d'utilisation de *kernsh* et le module qui simule un buffer overflow dans l'espace noyau. Plutôt que d'expliquer tous les moyens qui existent pour passer de l'espace utilisateur à l'espace noyau, nous nous focalisons sur deux modules qui présentent de nouvelles méthodes de dialogue entre le *userland* et le *kernelland*. Cependant, si vous voulez connaître la manière d'insérer un module dans `/dev/kmem` ou bien détourner une interruption (pour ne citer que ces deux exemples, regarder les références **[4]** et **[6]**).

UN BREF DESCRIPTIF DE KERNSH

Avant toute chose, nous tenons à préciser que *kernsh* est en constante évolution. Il se peut donc qu'il y ait quelques modifications entre la version disponible au moment où nous avons écrit l'article et celle qui le sera au moment où il paraîtra.

Pour compiler *kernsh*, modifiez la variable `IFLAGS` dans le fichier `modules/Makefile` puis un simple `make` suffit. Vous n'aurez plus qu'à lancer *kernsh* en tant que root. *kernsh* se présente sous la forme d'un shell. La commande `help` affiche toutes les commandes et `help [commande]` affiche le descriptif de la commande en question.

Voici, au moment où nous écrivons cet article, les commandes présentes dans *kernsh* :

```
% help
display : System information
symbol  : Description of the symbols
syscall : Description of the syscall
module  : Management of the modules
kmem    : Use of /dev/kmem
idt     : Handling of the IDT
help    : Display this help
exit    : Quit kernsh
```

Il reste beaucoup à développer, en particulier la gestion de `/proc` et l'insertion de *kpatch* (cf référence **[5]**). A savoir que *kpatch* est un utilitaire qui permet d'insérer un module dans une image kernel (`/boot/vmlinuz` par exemple). Donc, toute aide sera la bienvenue :>

Un exemple d'utilisation de Kernsh

Voici un exemple d'utilisation de *kernsh* avec le module qui simule un débordement de buffer où nous allons décrire chaque étape. Nous insérons le module directement dans `/dev/kmem` (prendre le module `buffer_overflow_lkm_2.o`) :

```
# ./kernsh
[...]
% module -s modules/buffer_overflow_lkm_2.o
buffer_overflow_lkm_2.o size: 172 octets
```

Nous commençons par regarder la taille du module, c'est la taille qu'il prendra une fois qu'il sera chargé en mémoire.

```
% kmem -m 172
module modules/lkm-kmalloc.o inserted in /dev/kmem..
memory malloc()ed at 0xc04cbea0
```

Nous allouons de la mémoire avec la fonction `kmalloc()` selon la taille obtenue, soit dans notre cas 172 octets.

```
% kmem -i modules/buffer_overflow_lkm_2.o 0xc04cbea0 overflow
overflow at 0xc04cbea0
module modules/buffer_overflow_lkm_2.o inserted in /dev/kmem..
% module -l
No module installed..
```

Nous insérons ensuite le module à l'endroit précédemment alloué. Nous pouvons voir (à l'aide de la commande `module -l`) que le module `buffer_overflow_lkm_2.o` n'est pas listé parmi les modules installés. En revanche, en éditant `/dev/kmem` et en comparant avec ce que nous obtenons en éditant (avec l'outil `bvi`) le module, il apparaît clairement qu'il y est inséré :

```
% kmem -d 0xc04cbea0 128

[ 0xc04cbea0 -> 0xc04cbf20 ]
55 89 e5 81 ec 08 01 00  00 8b 45 08 89 45 04 c6  | U.........E..E..
45 08 00 31 c0 c9 c3 90  55 89 e5 83 ec 08 83 c4  | E..1....U.......
f4 68 fe be 4c c0 e8 01  66 c4 ff 31 c0 c9 c3 90  | .h..L...f..1....
55 89 e5 83 ec 08 83 c4  f4 68 0c bf 4c c0 e8 e9  | U........h..L...
65 c4 ff c9 c3 8d 76 00  6b 65 72 6e 65 6c 5f 76  | e.....v.kernel_v
65 72 73 69 6f 6e 3d 32  2e 32 2e 31 39 00 5b 2b  | ersion=2.2.19.[+
5d 20 6c 6f 61 64 65 64  2e 2e 0a 00 5b 2b 5d 20  | ] loaded....[+]
75 6e 6c 6f 61 64 65 64  2e 2e 0a 00 00 47 43 43  | unloaded.....GCC
```

Notez qu'un module inséré par `/sbin/insmod` aura ses 4 premiers octets systématiquement égaux à 0x60 0x00 0x00 0x00 sur noyau 2.4.x et 0x54 0x00 0x00 0x00 sur noyau 2.2.x. Cela peut être utile pour détecter une

UN EXEMPLE D'UTILISATION DE KERNSH

structure module valide qui aurait été retirée de la module_list, et ne serait donc pas visible avec la commande /sbin/lsmod ou dans /proc/modules.

```
% syscall -h 250 0xc04cbea0
hijacked syscall : 250
    with address : 0xc04cbea0

% syscall -l
[...]
syscall# 250: 0xc04cbea0        syscall# 251: 0xc01129e4
syscall# 252: 0xc01129e4        syscall# 253: 0xc01129e4
syscall# 254: 0xc01129e4        syscall# 255: 0xc01129e4

% exit
#
```

Nous détournons (ou *hijack()ons*) ensuite le syscall 250 (ou n'importe quel syscall qui n'est pas utilisé dans la table des syscall) et nous le faisons pointer sur l'adresse de la fonction overflow() du module buffer_overflow_lkm_2.o. La fonction overflow() sera alors appelée en déclenchant le syscall détourné. Comme tout à l'heure, c'est le client client_buffer_overflow qui va se charger de déclencher le syscall 250 et donc le débordement de buffer.

```
$ ./client_buffer_overflow
[+] shellcode: 0x80494e0..
sh-2.05a# exit
$
```

Rappelez-vous, le module implémentait lui-même un syscall (plus précisément le syscall 250) qui déclenchait un débordement de buffer :

```
[...]
#define SYS_BOF 250
[...]
sys_call_table[SYS_BOF] = ( void * )overflow;
[...]
```

Cette fois-ci, c'est différent, *kernsh* nous permet de modifier directement la table des syscall. La fonction init_module() ne joue donc plus aucun rôle. Faites alors le test de charger en mémoire un fichier objet ne contenant que la fonction overflow() du module, nous obtiendrons de la même manière que précédemment un shell.

Nous vous laissons le soin de faire les mêmes manipulations avec le second module (struct_lkm_2.o et client_struct).

Les résultats s'avèrent aussi concluants :>

CONCLUSION

Nous avons introduit dans cet article les notions fondamentales de niveau de privilèges (ringlevel), d'espace noyau et utilisateur, les moyens de passer de l'un à l'autre grâce aux interruptions. Le principal objectif était de vous démontrer qu'il était possible de faire communiquer l'espace utilisateur et l'espace noyau de différentes manières et celles que nous vous avons montrées en étaient deux parmi tant d'autres.

Nicolas "sauron" Brito - unixlabs@noos.fr

Samuel "zorgon" Dralet - zorgon@mastersecurity.fr

RÉFÉRENCES

[1] Le livre "*Understanding the Linux kernel*" Daniel P. Bovet et Marco Cesati

[2] "*Les patchs kernel*" - Frédéric "pappy" Raynal, Samuel "zorgon" Dralet
http://www.miscmag.com/articles/index.php3?page=212

[3] "*Root-kit et intégrité*" - Frédéric "pappy" Raynal
http://www.security-labs.org/index.php3?page=404

[4] "*RUNTIME KERNEL KMEM PATCHING*" - Silvio Cesare
http://ouah.sysdoor.net/runtime-kernel-kmem-patching.txt

[5] "*Static Kernel Patching*" - jbtzhm
http://www.phrack.org/phrack/60/p60-0x08.txt

[6] "*Handling Interrupt Descriptor Table for fun and profit*" - kad
http://www.phrack.org/phrack/59/p59-0x04.txt

[7] "*Linux on-the-fly kernel patching without LKM*" - sd et devik
http://www.phrack.org/phrack/58/p58-0x07

LA CONFÉRENCE 100% SÉCURITÉ

Au programme :

- Guerre de l'information
- Cryptographie
- Sécurité des systèmes
- Protection des réseaux

SYMPOSIUM SSTIC
SUR LA SÉCURITÉ DES TECHNOLOGIES DE L'INFORMATION ET DES COMMUNICATIONS

Renseignements et inscriptions :
http://www.sstic.org

SYMPOSIUM SUR LA SÉCURITÉ DES TECHNOLOGIES DE L'INFORMATION ET DES COMMUNICATIONS

10 - 12 juin 2003 À RENNES

Supélec - École Supérieure d'Électricité

Supélec · CGBI Cartel sécurité · misc multi-system & internet security cookbook · cea

Sécuriser

Contrairement à une idée largement répandue par les médias généralistes, les systèmes Unix libres ne se limitent pas à un certain Linux (pardon GNU/Linux !). Depuis à peu près la même époque, à savoir le début des années 90, le monde BSD (Berkeley Software Distribution) propose également une panoplie de systèmes Unix libres. Il existe trois grandes familles, FreeBSD, NetBSD et OpenBSD, auxquelles on peut ajouter Darwin, le cœur du Mac OS X d'Apple.

FreeBSD est probablement le plus répandu, puisque, comme pour Linux, le processeur de prédilection est l'incontournable Intel, autrement dit, celui qui équipe sans doute pas loin de 90 % des ordinateurs de la planète.

NetBSD est de loin le plus versatile, puisqu'il doit fonctionner à peu près sur tout ce qui ressemble à un ordinateur, y compris des "rossignols" (non, pas les oiseaux, mais les vieilles choses démodées), dont une grande partie des lecteurs n'a certainement jamais entendu parler. Bien sûr, ça ne l'empêche pas de reconnaître les incontournables processeurs Intel.

Enfin, *last but not least*, OpenBSD se veut (à juste titre) la référence en matière de serveurs pour ce qui concerne la sécurité.

Alors pourquoi traiter de FreeBSD plutôt que de ses semblables ? Parce que !

Plus sérieusement, parler de l'un amène inévitablement à parler des autres. Ils sont issus du même point de départ : 4.4BSD au tout début, et BSD Lite ensuite. Les différents "parfums" sont nés de certaines dissidences entre concepteurs, mais ils continuent à partager l'essentiel.

FreeBSD va nous permettre de considérer des cas différents. Autrement dit, nous prendrons comme exemples successifs une machine destinée à un usage personnel, un hôte destiné à faire office de serveur dans un réseau local et enfin un ordinateur utilisé comme passerelle vers le monde extérieur.

Dans cet article, nous utiliserons la version stable en cours au moment d'écrire ces lignes, la 4.7. Nous n'aborderons pas la version 5.0, qui n'existera sans doute pas dans la branche stable avant les versions 5.1 ou 5.2. La branche stable reste encore l'apanage des versions 4 : une 4.8 devrait apparaître en mars (2003 !) et suivront peut-être une 4.9, voire même une 4.10.

Oui, le monde BSD est très actif tout en sachant rester discret...

FREEBSD CHEZ SOI

Se procurer un CD-ROM de la dernière distribution stable ne devrait pas poser de nombreux problèmes. Un CD parce que vous n'avez pas de connexion à Internet : votre machine possède un disque dur désespérément vide... et il n'y a même pas Windows, n'est-ce pas ?

J'ai l'air de plaisanter, mais vous êtes en train de lire une revue dédiée à la sécurité informatique, et le "dual-boot" (ou le "multi-boot") n'est pas forcément ce qui se fait de mieux dans ce domaine, loin s'en faut.

Donc, l'ordinateur en question, va se voir offrir un système tout neuf...

Nous ne nous attarderons pas sur l'installation : FreeBSD possède un excellent manuel livré avec la distribution. Ce ne sera pas une installation "tout graphique" comme c'est la mode ailleurs, mais ce sera pourtant très simple. Suivez les instructions fournies par l'interface *ncurses*.

Ça y est, FreeBSD est sur votre disque dur, il ne reste plus qu'à le configurer... et à recompiler son noyau ! Ben oui, il faut bien qu'il réponde à vos besoins.

Mais nous n'en sommes pas encore là. Commençons donc par le commencement.

COMPTES UTILISATEURS

Avant toute autre chose, sécurisez le compte root et les comptes utilisateurs. Ce n'est pas une obligation, mais vous pouvez éditer le fichier `/etc/login.conf` afin de modifier l'algorithme de cryptage des mots de passe. Dans la ligne `passwd_format`, remplacez la valeur par défaut `des` par `blf` pour Blowfish (la raison de ce choix, c'est que Blowfish est plus "sûr" que DES). N'oubliez pas ensuite de reconstruire la base par la commande `cap_mkdb /etc/login.conf`.

FreeBSD 4.7

Si vous le souhaitez, appliquez cette modification à la commande adduser en éditant /etc/auth.conf pour que blf devienne le cryptage par défaut. La ligne concernée doit contenir : crypt_default=blf. Avec le cryptage Blowfish, les mots de passe commencent par $2.

Supprimez les comptes "inutiles" tels que toor, uucp, games... Les commandes chpass, adduser et rmuser sont bien sûr incontournables pour ce qui concerne la gestion des utilisateurs.

Il est également possible d'utiliser S/key pour obtenir l'équivalent d'un "OTP" (*One Time Password*). Voir la page de manuel skey.

Enfin, vous pouvez choisir un autre mode d'authentification par l'intermédiaire de Kerberos, par exemple. En règle générale, cela concerne plus une machine membre d'un réseau qu'un ordinateur individuel, mais on fait comme on veut...

Précisons que le fichier /etc/login.conf permet de définir de nombreux paramètres tels que les quotas utilisateurs, certaines variables d'environnement et différentes restrictions.

Comme toujours, choisissez de ne pas vous loger directement en tant que root en préférant la commande su. Pour ce faire, l'utilisateur autorisé à se servir de ladite commande doit faire partie du groupe wheel : ajoutez-le dans le fichier /etc/group. Le groupe wheel ne doit en aucun cas être son groupe primaire. Lors de la création de l'utilisateur par la commande adduser, choisissez par exemple le groupe staff comme groupe primaire, si vous n'avez pas défini cela par défaut dans le fichier /etc/login.conf.

Une autre possibilité consiste à utiliser la commande sudo.

SERVICES

Parmi les indispensables vérifications, regardons les services actifs... et inutiles. FreeBSD a déjà gagné un bon point : les services pilotés par inetd sont tous désactivés par défaut. Pourquoi tout le monde ne fait pas la même chose ?

Si vous n'avez besoin d'aucun des services pilotés par inetd, vous pouvez le désactiver en ajoutant inetd_enable="NO" dans /etc/rc.conf. C'est théoriquement le réglage défini par défaut, mais sysinstall l'active si le niveau de sécurité choisi est "medium". Si vous y tenez vraiment, vous pouvez aussi remplacer inetd par xinetd.

Ajoutons que si vous arrêtez définitivement inetd, il est évident que vous ne bénéficierez plus de TCPWrappers.

Le fichier /etc/rc.conf permet de configurer ou de définir de nombreux éléments. Il peut servir à lancer des démons tels que Sendmail, SSHd, NFSd, etc. Si vous avez choisi les "réglages modérés" lors de l'installation (*medium* dans la fenêtre de profil de sécurité par défaut), ces démons sont actifs.

A vous de les arrêter en remplaçant YES par NO dans les lignes correspondantes.

Vous pouvez également définir les niveaux de sécurité du noyau dans ce même fichier rc.conf. Le niveau par défaut, -1, est le mode non sécurisé permanent.

Pour une machine utilisée chez vous, le niveau 0 est acceptable : c'est le mode non sécurisé. Au-delà de ce niveau, vous pourrez commencer à rencontrer des problèmes de fonctionnement.

Le niveau 1 correspond au mode sécurisé, le niveau 2 au mode hautement sécurisé, et le niveau 3 au mode sécurisé réseau. La page de manuel de init vous expliquera tout en détail.

Il n'est pas nécessaire de se focaliser sur ces niveaux : l'essentiel est de savoir à quoi ils correspondent, ou si vous préférez, ce qu'ils produisent sur le système. D'une certaine manière, ce sont de "faux amis" qui donnent une impression de sécurité pas toujours fondée, au moins pour ce qui concerne les niveaux les plus bas. Il suffit d'en être conscient : comme disait ma mémé, il vaut mieux y mettre les deux mains ;-)

PERSONNALISATION

Autre élément essentiel avec lequel nous allons "jouer" : *sysctl*. Cet outil (ou son équivalent) se retrouve dans tout système Unix, propriétaire ou libre. Sous Linux, il porte le même nom, sous Solaris vous disposez de ndd, sous Irix de systune, etc. Il sert à un tas de choses : améliorer les performances, modifier le comportement de la pile TCP/IP, gérer la mémoire virtuelle, etc., contribuant ainsi à une meilleure sécurité. Pour information, *sysctl* possède plus de 500 options : à vous la page de manuel !

sysctl peut fonctionner de manière interactive, en tapant la commande dans un shell ou s'utiliser par l'intermédiaire d'un fichier de configuration /etc/sysctl.conf, dans lequel vous écrirez la totalité des paramètres souhaités. Nous pouvons difficilement nous attarder sur toutes les options de *sysctl* et nous nous contenterons de quelques exemples susceptibles de peupler votre fichier sysctl.conf.

net.inet.ip.check_interface=1 permet de contrôler que les paquets arrivent bien de l'interface d'où ils prétendent venir.

net.inet.tcp.blackhole=2 et net.inet.udp.blackhole=1 permettent de transformer votre système en "trou noir" lors d'éventuelles tentatives de *scan* sur votre machine.

net.inet.ip.sourceroute=0 contribue à réduire les risques de "spoofing" en empêchant les paquets d'emprunter la même route en entrée et en sortie.

Il n'en reste plus que quelques 500... et vous les trouverez dans les fichiers "include" des sources de sys, netinet et vm. Voir la page de manuel de *sysctl*.

CONNEXION À INTERNET

Configurer la connexion est très bien expliqué dans la documentation, que vous utilisiez PPP, PPPoE ou PPPoA. Nous ne nous y attarderons pas.

Le fait de ne pas avoir recompilé le noyau signifie que vous n'avez pas, par exemple, de pare-feu puisque aucun n'est actif par défaut. Si vous faites partie de la catégorie des "même pas peur", vous avez heureusement la possibilité d'utiliser des filtres avec PPP, ou celle d'utiliser d'autres outils disponibles sur le CD. Les filtres sont bien sûr moins efficaces qu'un pare-feu, mais ils permettent malgré tout de limiter les dégâts. Cela dit, rappelons au passage qu'un pare-feu n'est pas la panacée universelle, mais bon...

Les filtres en question s'ajoutent au fichier /etc/ppp/ppp.conf sous la forme :

```
set filter alive 0 deny icmp
set filter alive 1 deny udp src eq 53
set filter alive 2 deny udp dst eq 53
```

Vous pouvez définir jusqu'à 20 règles. Vous trouverez plus ample information dans le "livre" *PPP Primer* qui fait partie de la documentation fournie.

Si au contraire, vous faites partie de la catégorie "je suis parano mais je me soigne", nous allons enfin en arriver à ce fameux noyau.

CONFIGURATION ET RECOMPILATION DU NOYAU

Comment configurer et recompiler le noyau est très bien expliqué dans le manuel de FreeBSD (oui, encore !). En conséquence, nous nous attarderons surtout sur les options de configuration. Au passage, n'oubliez pas de travailler sur une copie du fichier GENERIC que vous pourrez baptiser comme bon vous semble !

La référence à utiliser pour ces options de configuration n'est autre que le fichier LINT. Vous pourrez y prélever ce qui vous intéresse et qui ne figure pas dans votre propre fichier GENERIC (encore une fois, pas l'original !).

La raison majeure de cette recompilation du noyau n'est autre que l'activation de certains éléments... comme un pare-feu, par exemple. Le pare-feu de FreeBSD est le "vieux" *ipfw*. Sinon, vous pouvez choisir d'installer IPFilter. Pour activer ipfw, il suffit de décommenter les lignes d'options IPFIREWALL, sauf celle qui contient IPFIREWALL_DEFAULT_TO_ACCEPT ; sinon, à quoi bon activer le pare-feu s'il doit tout accepter :-(

Si vous utilisez PPPoE, par exemple, et que vous n'activiez pas les options NETGRAPH, les nœuds seront chargés automatiquement sous forme de modules.(Le système netgraph propose des objets pour le noyau destinés à différentes fonctions réseau. Ces objets sont appelés "nœuds", ou "nodes" in english.)

Dans une optique "moins on me voit, mieux je me porte", activez l'option TCP_DROP_SYNFIN. Elle rend votre machine non conforme à la norme TCP/IP, mais l'empêche d'être identifiée par des scanners. Pour une machine personnelle, c'est acceptable mais ne choisissez pas cette option pour un serveur HTTP !

Activez bien évidemment les options RANDOM_IP_ID (le nom suffit à expliquer de quoi il s'agit) et ICMP_BANDLIM. Ce dernier aide à protéger contre les attaques DoS (*Denial of Service*).

Dans un autre genre, certains documents "recommandent" de désactiver le pseudo-device *bpf* (*Berkeley Packet Filter*) si votre connexion à Internet n'utilise pas DHCP.

Sachez cependant que certains outils, Snort ou Tcpdump par exemple, ne fonctionneront pas dans ce cas, puisqu'ils doivent être capables de "passer" votre carte réseau en mode *promiscuous*. Vous risquez aussi quelques problèmes si vous utilisez les nœuds Netgraph puisque l'un d'entre eux dépend de *bpf* (ng_bpf). D'autres protocoles peuvent aussi se trouver altérés, voire incapables de fonctionner.

Alors, est-ce que ça vaut vraiment la peine ? A vous de voir !

Voilà pour l'essentiel, mais c'est à vous d'étudier les différentes possibilités en fonction de votre configuration et selon que vous ayez ou non un petit réseau local.

Il ne reste plus qu'à compiler votre nouveau noyau et c'est tout fini. Enfin presque !

PARE-FEU

Puisque nous avons maintenant un noyau prêt à gérer *ipfw*, c'est le moment de définir quelques règles de filtrage. Le nombre de ces règles étant pour le moins conséquent, nous nous contenterons d'utiliser le fichier exemple /etc/rc.firewall qu'il vous faudra adapter à votre cas. Selon le type de connexion, la gestion est légèrement différente, au moins pour ce qui concerne les interfaces et les protocoles. Cela se définit dans le fichier /etc/rc.conf.

Par exemple, si vous utilisez une connexion par câble avec PPPoE, les définitions seront différentes de celles applicables à une connexion par modem avec DHCP. C'est particulièrement important si vous avez un petit réseau local sur lequel les machines sont susceptibles de bénéficier de cette connexion et que vous souhaitiez les masquer.

Le problème ici vient de ce qu'il faudrait un ou plusieurs magazine(s) pour traiter le sujet. L'équipe de MISC ayant déjà fait le travail dans deux numéros hors série de Linux Magazine, nous allons faire le plus bref possible.

Dans le cas d'une connexion par modem, vous pourrez par exemple utiliser natd_enable="YES", natd_interface="tun0" et natd_flags="-dynamic.

Ensuite, vous n'aurez plus qu'à choisir le type parmi les options "open", "client" ou "simple". Éditez le fichier rc.firewall en conséquence.

Dans le cas d'IPv6, il faudrait faire la même chose avec le fichier rc.firewall6.

Pour information, l'évolution *ipfw2* est maintenant disponible dans la version stable. Pour en bénéficier, vous devez ajouter `IPFW2=TRUE` dans `/etc/make.conf` avant de reconstruire votre système. La page de manuel de *ipfw* décrit les améliorations apportées.

PERMISSIONS ET INTÉGRITÉ

Un petit `chmod 700` sur le répertoire `root` ne peut pas faire de mal. Dans `/etc` un `chmod 600` sur la majorité des fichiers de configuration n'aura rien de répréhensible.

Selon le niveau de sécurité choisi (*a priori* 0, dans notre cas), vous pourrez aussi rendre certains fichiers non modifiables sauf par `root`. La commande `chflags` et son option `schg` feront ça très bien avec en plus la possibilité d'appliquer la restriction à un répertoire entier grâce à la récursivité (`-R`). Quelques candidats, parmi d'autres, sur lesquels assigner ce drapeau peuvent être les répertoires contenant les exécutables comme `/bin` ou `/sbin`.

Signalons au passage que le script `/etc/security` lancé par `cron`, permet de vérifier (entre autres choses) l'intégrité des fichiers et des périphériques SUID.

Vous disposez aussi d'un excellent outil nommé `mtree` capable de comparer la hiérarchie du système de fichiers à une spécification qu'il aura lui-même générée. Une fois cette spécification créée, vous pourrez lancer un comparatif de manière cyclique par `cron`. Voir le répertoire `/etc/mtree` et la page de manuel pour en savoir plus.

LOGS

Puisque nous parlons d'une machine personnelle, il est probable que les logs seront gérés localement et non sur une machine distante. Lancez donc `syslogd` avec l'option `-ss`. Cela offrira l'avantage de fermer le port UDP 514. Une petite ligne dans `/etc/rc.conf` contenant `syslogd_flags="-ss"` suffira. Puisque nous sommes dans les logs, profitons-en pour taper `chmod -R 600` sur le répertoire `/var/log`.

Pour limiter la taille des logs et gérer leurs rotations, utilisez `newsyslog` dans une tâche `cron`. Cette commande possède son propre fichier de configuration `/etc/newsyslog.conf`. Vous pourrez le modifier afin de le faire correspondre à vos besoins.

CONSOLE, TERMINAUX ET X11

Editez le fichier `/etc/ttys` et remplacez le mot `secure` par `insecure` sur la ligne concernant la console afin d'exiger le mot de passe de `root` pour le passage en mode *single user*.

A ce propos, de nombreux terminaux sont actifs par défaut dans ce fichier : si vous n'en avez pas besoin, n'hésitez pas à en passer quelques-uns à `off` pour les désactiver... mais gardez-en au moins un !

Si vous utilisez `startx` pour lancer X et qu'aucune machine n'a besoin de se connecter à la vôtre par son intermédiaire, fermez le port, ça ne vous empêchera pas d'utiliser votre interface graphique. Pour ce faire, éditez le script `startx` et modifiez-le afin d'obtenir la ligne `serverargs="-nolisten tcp"`. Maintenant, après redémarrage de `startx`, le port 6000 n'apparaîtra plus lorsque vous taperez la commande `sockstat -4` (ou `lsof` qui la remplacera avantageusement).

APPLICATIONS

Que serait un système d'exploitation sans applications ? FreeBSD propose deux manières d'installer des logiciels : les paquetages et les ports. Les paquetages peuvent être assimilés à ce qui se pratique ailleurs (Linux avec les *rpm* ou les *deb*, Solaris avec ses *pkg*, etc.). Les ports, eux, sont une particularité de la famille BSD.

Les ports ont un intérêt certain dans la mesure où ils gèrent les dépendances. Toutefois, ils ont un inconvénient majeur (ceci n'engage que moi !), il faut se connecter en tant que `root` pour les télécharger et les installer. Objectivement, ce n'est pas ce que je préfère. Alors, certes, les ports sont censés vérifier les signatures des paquetages, mais tout se passe sans votre intervention et, répétons-le, en étant connecté en tant que `root`. Heureusement, il reste les classiques archives `tar.gz`, mais ce n'est pas non plus une garantie... pas plus que les paquetages d'ailleurs. Alors acceptons-en l'augure :-(

Profitons-en pour mentionner l'outil de mise à jour *CVSup* qui permet de synchroniser le code source. Il devient ainsi possible d'avoir un système théoriquement exempt de vulnérabilités connues. Bien évidemment... il faut être `root` pour l'utiliser :-(

Pour en terminer avec ce chapitre, précisons qu'il est intéressant d'activer la compatibilité Linux pour bénéficier de l'émulation binaire avec FreeBSD. Celle-ci vous permet de profiter de "presque" toute la logithèque disponible. Pour ce faire, une simple ligne `linux_enable="YES"` dans le fichier `/etc/rc.conf` suffit si vous avez sélectionné l'option `COMPAT_LINUX` pendant la configuration du noyau.

Il reste encore beaucoup à faire, mais vous disposez maintenant d'une machine moins vulnérable. A vous d'approfondir pour découvrir tout ce qu'il est possible d'améliorer. Vous pourrez aussi appliquer certaines des restrictions qui vont suivre.

FREEBSD DANS UN RÉSEAU LOCAL

Parmi les nombreux avantages des systèmes libres et souvent presque gratuits, le coût est évidemment la première chose qui vient à l'esprit. Il existe un autre avantage important qui découle du précédent, c'est la possibilité de multiplier le nombre de machines dans un réseau en recyclant des ordinateurs relativement anciens et en les équipant de ces systèmes. L'intérêt majeur c'est d'arriver à ce que chaque machine ne joue qu'un seul rôle, ou si préférez, qu'elle n'exécute qu'une seule tâche importante. En d'autres termes, il s'agit de ne pas mettre tous ses oeufs dans le même panier. Dernier point et non le moindre : vous devez savoir à quelle fonction vous allez affecter ce serveur avant d'installer le système. Selon le rôle attribué, vous installerez uniquement ce dont il a besoin pour fonctionner correctement. Par exemple, dans la majorité des cas, il sera inutile d'installer le X Window System.

FreeBSD, c'est "le pouvoir de servir". Il n'est pas gourmand en ressources et peut par conséquent se contenter d'un processeur dont la fréquence se mesure encore en Mhz et d'une quantité de mémoire "raisonnable". Il est bien évident que la fonction qu'il aura pourra tempérer ces propos. Je m'explique : si vous voulez faire de cette machine un serveur de bases de données avec 100 connexions simultanées, vous allez au-devant de quelques soucis pour plusieurs raisons.

SERVEUR DE BASES DE DONNÉES

Si vous comptez installer un serveur Oracle ou MySQL sur une machine équipée de FreeBSD, vous allez vous heurter à certaines limitations qui concernent aussi bien les ressources du système que certaines de ses caractéristiques. Oublions le côté ressources pour l'instant pour ne traiter que de la partie "caractéristiques".

Si nous prenons le cas d'Oracle, ce sera le parcours du combattant. Si vous arrivez à le faire fonctionner correctement (bonne chance !), ce sera grâce à l'émulation binaire puisqu'il s'agira d'une version Linux. Autant le dire tout de suite : je vous déconseille fortement cette "solution".

Si nous prenons le cas de MySQL, nous allons rencontrer d'autres problèmes : les "threads". FreeBSD utilise les threads à l'échelle de l'utilisateur. En clair, tout se passe dans l'espace utilisateur et non dans l'espace noyau. Une solution consiste à compiler MySQL en le "liant" à la bibliothèque LinuxThreads. Ça fonctionne mieux mais c'est loin d'être parfait. De plus, il sera préférable que le nombre de connexions simultanées soit du genre "restreint". Si c'est le cas, vous pouvez parfaitement envisager un serveur MySQL sous FreeBSD que vous vous empresserez d'installer dans une "jail".

Enfin, si nous revenons au point de départ, il faudra malgré tout une machine "musclée". En conséquence, préférez Linux pour ce genre de serveur.

Pour rester dans les bases de données, PostgreSQL sera peut-être mieux adapté à FreeBSD que les deux SGBDR ci-dessus. La version 5 de FreeBSD devrait apporter la solution à ces problèmes.

FreeBSD peut aussi devenir, par exemple, un excellent serveur SSH, un serveur HTTP pour votre intranet ou un serveur de courrier sérieux. Ce n'est bien évidemment pas limitatif, vous pouvez choisir bien d'autres rôles.

SERVEUR SSH

Ici, nous sommes beaucoup plus dans le cadre des qualités de FreeBSD. Celui-ci bénéficie du travail effectué sur son "petit frère" OpenBSD, qui accessoirement, se charge aussi du développement d'OpenSSH. Nous ne reviendrons pas sur les avantages et inconvénients d'OpenSSH (voir le hors série de Linux-Magazine qui a servi de rampe de lancement à MISC).

Considérons SSH pour ce qu'il est : un remplacement incontournable des R-Commands (`rsh`, `rcp`, `rlogin`, etc.), et donc un moyen de faire circuler les données de manière cryptée. Et pas autre chose !

Cela ne signifie pas qu'il faille le traiter de façon désinvolte. Sous FreeBSD, nous disposons de plusieurs possibilités pour faire de notre serveur une machine "sérieuse".

Tout d'abord, nous installons un système réduit à sa plus simple expression. Pas besoin de X et de tout ce qui va avec, par exemple. N'installez que le strict nécessaire... et vous obtiendrez un système de très petite taille, et par conséquent, aucun problème de ressources, même si SSH est exigeant dans ce domaine.

Nous pouvons passer le niveau de sécurité à 1 dans `/etc/rc.conf`. Cela signifie qu'il n'y a plus de possibilité d'écriture sur `/dev/mem` et `/dev/kmem`. Les modules du noyau ne peuvent plus être chargés ou déchargés. Nous appliquons ce niveau 1 parce que nous n'utilisons pas X. Si X était installé, il est probable que nous aurions beaucoup de difficultés à nous en servir.

Pour ce qui concerne la configuration, nous n'allons pas réinventer la roue, mais simplement mentionner l'essentiel. Toute cette partie se passe dans le fichier `/etc/ssh/sshd_config`. Parmi les suggestions, passez le protocole par défaut à 2, ou si vous préférez, décommentez la ligne `Protocol 2,1` et supprimez le 1. Pas de ForwardAgent : vous devez donc avoir une ligne spécifiant `ForwardAgent no`. Pas de X11Forwarding non plus : X11 n'est pas installé ! Votre fichier doit contenir la ligne `X11Forwarding no`. Pas d'authentification par mot de passe : il vous faut une ligne `PasswordAuthentication no`. L'authentification se fera par clés DSA ou RSA. Théoriquement, tous ces réglages sont définis par défaut dans les versions récentes d'OpenSSH et de FreeBSD.

Passons maintenant à l'une des caractéristiques intéressantes de FreeBSD : les *jails* (ou prisons, si vous préférez). Pour faire court, *jail* c'est un *chroot* amélioré. Il s'agit de recréer un environnement système dans un répertoire spécifique et protégé. Le problème majeur posé par une *jail*, c'est la gestion des processus qui y sont lancés. Une des solutions consiste donc à limiter au maximum le nombre de ceux qui seraient activés en dehors de celle-ci.

Nous n'allons pas décrire la création d'une "jail", les pages de manuel font ça très bien et vous trouverez de nombreux liens dans la section Références à la fin de l'article.

Insistons simplement sur la manière de procéder. Créez une "jail" pour chaque utilisateur autorisé à se connecter au serveur SSH, et bien sûr, une autre pour SSH proprement dit. Paradoxalement, lorsque vous créez une "jail", n'essayez pas de faire "minimaliste". Il vaut mieux enlever ce qui est en trop *a posteriori* qu'ajouter des éléments petit à petit.

sysctl possède également des options spécifiques aux "jails". N'hésitez pas à les utiliser. Encore une fois, tout est dans les pages de manuel.

Vous pouvez aussi modifier `/etc/syslog.conf` et `/etc/newsyslog.conf` afin d'obtenir des logs spécifiques à SSH. Par exemple, ajoutez la ligne :

```
auth.info /var/log/auth
```

Cela permettra de ne pas utiliser le "logging" par défaut dans `/var/log/security`.

Vous disposez maintenant d'un nouveau serveur qui n'est là que pour recevoir les connexions des clients SSH de votre réseau local. Non seulement il ne vous a pratiquement rien coûté, mais il a déchargé une autre machine de ce rôle superflu, puisqu'elle avait déjà une autre activité. De plus, si vous avez bien fait votre travail, ce nouveau serveur ne devrait pas être parmi les plus vulnérables de votre parc, ne serait-ce que par le nombre de ports ouverts (en l'occurrence, un seul... tout au moins quand il n'y a pas de connexion !) et le peu de services actifs (le strict minimum).

Nous pourrions aller plus loin encore, en envisageant l'utilisation d'IPSec pour créer un réseau privé virtuel (*Virtual Private Network*). Le sujet étant plutôt vaste, il nécessiterait un article entier, en conséquence nous ne l'aborderons pas. Rassurez-vous, la littérature concernant IPSec est très fournie.

SERVEUR HTTP

Malgré quelques soucis plus ou moins récents, Apache reste le serveur HTTP de référence pour FreeBSD comme pour les autres OS. Avec les mécanismes de *jail* déjà cités, le système d'hôtes virtuels et les modules du style `mod_ssl`, il y a vraiment moyen d'obtenir un serveur HTTP digne de ce nom.

Le premier travail consiste à bien définir la configuration dans `httpd.config`... ce qui n'entre pas vraiment dans le cadre de cet article. De plus, "comment sécuriser Apache" se trouve en de nombreux exemplaires sur Internet.

Ensuite, mettez le serveur "en prison"... et `jail` c'est mieux que `chroot`. Définissez les éventuels hôtes virtuels dans `/etc/rc.conf`. Voir le chapitre "Virtual hosts" de la documentation FreeBSD.

Par défaut, le serveur doit fonctionner sous propriétaire et groupe `www`. Le fait d'utiliser `mod_ssl` implique également l'ouverture du port 443. Le niveau de sécurité peut aussi dans ce cas être défini à 1.

Comme précédemment, le système doit contenir le minimum requis... et si le site doit utiliser une base de données, celle-ci sera plus à sa place sur une autre machine !

SERVEUR DE COURRIER

Arbitrairement, nous choisirons un serveur *Postfix*. Comme pour le serveur HTTP, une "hénaurme" littérature est disponible pour Postfix. Nous recommanderons simplement l'utilisation des domaines virtuels. Comme toujours, `jail` contribuera à la sécurisation de l'ensemble. Vous pourrez aussi envisager l'utilisation de TLS pour encore améliorer les choses.

Précisons que Postfix est l'un des serveurs de courrier parmi les plus (sinon le plus) robustes et qu'il se sentira très bien dans l'environnement FreeBSD

Encore une fois, ce qui précède ne peut en aucun cas être exhaustif. Tout ceci peut être obtenu de la même manière avec les autres BSD libres ou avec Linux. Il ne s'agit que d'exemples. Pour certains serveurs (HTTP, Postfix, etc), OpenBSD sera encore mieux adapté.

FREEBSD EN TANT QUE PASSERELLE

Voici un autre moyen de recycler efficacement une machine relativement ancienne : offrez une passerelle vers le monde extérieur à votre réseau local tout en améliorant sa protection.

Une nouvelle fois, nous allons installer un FreeBSD minimal : uniquement l'absolu strict nécessaire. Il nous faudra quand même les sources (à jour et avec les éventuels patches de sécurité) et un compilateur... pour recompiler le noyau :-(

Cette machine doit posséder deux cartes réseau. Configurez la première avec une adresse IP non routable (192.168.0.1 ou 10.0.0.1 par exemple) dans `/etc/rc.conf`. Pour la deuxième (celle qui ouvre vers le monde extérieur), considérons que vous avez la chance d'avoir un prestataire qui vous a attribué une adresse IP fixe. Définissez-la également dans `/etc/rc.conf`. Si ce n'est pas le cas, il faudra malheureusement activer "l'infâme" DHCP par une ligne du style `ifconfig_xl1="DHCP"` (remplacez `xl1` en fonction de votre type de carte réseau). A ce moment-là, il faudra aussi éditer le fichier `/etc/dhclient.conf`.

Désactivons la totalité des services, s'il en reste : pas de *sendmail*, *inetd* ou autre *portmap*. Vérifiez par des commandes comme `lsof` ou `sockstat`. La liste des ports à l'écoute devrait être vide.

Configurons maintenant notre nouveau noyau, comme précédemment. Cette fois, arbitrairement et pour varier les plaisirs, nous choisissons d'activer *IPFilter* à la place d'*ipfw* (décommentez les options `IPFILTER` dans votre copie de "`GENERIC`"). N'hésitez pas à sélectionner les options mentionnées plus haut comme `RANDOM_IP_ID` ou `ICMP_BANDLIM`, mais soyez encore plus restrictifs en ajoutant par exemple, `NO_LKM` et `NO_KLD`.

Enfin, compilez le nouveau noyau.

Modifiez vos fichiers `/etc/rc.conf` et `/etc/sysctl.conf`. Le premier doit contenir les lignes :

```
ipfilter_enable="YES"
ipfilter_flags=""
ipnat_enable="YES"
ipmon_enable="YES"
```

Vous pouvez ajouter `ipmon_flags` en indiquant les options choisies (voir la page de manuel d'*ipmon*).

Dans le second fichier, vous devrez inscrire `net.inet.ip.forwarding=1` en plus de tout le reste (voir la partie "personnalisation"). Vous pouvez bien sûr y ajouter des options destinées à améliorer les performances réseau, du style `net.inet.tcp.recspace` et `net.inet.sendspace` en leur affectant des valeurs égales à 65535.

C'est le moment de "durcir" l'ensemble avec la commande `chflags schg` et ses options `-R` (pour la récursivité) et `-H` (pour que la restriction s'applique aux liens symboliques) sur la plupart des répertoires.

Une précision "évidente" : lors d'une mise à jour du système, il faut supprimer ces restrictions par la commande inverse `chflags noschg` sous peine de la voir échouer.

Le niveau de sécurité peut être maintenu à 1, mais ce serait encore mieux de le passer à 2. Testez les deux possibilités. N'oubliez pas non plus d'utiliser `mtree` comme déjà mentionné.

Maintenant, il ne reste plus qu'à définir les règles pour *IPFilter* et *ipnat*.

■ Quelques exemples pour IPFilter :

Pour bloquer tous les paquets entrants et sortants non autorisés ultérieurement :

```
block in log all
block out log all
```

Pour laisser circuler le trafic interne :

```
pass in quick on xl0 all
pass out quick on xl0 all
```

■ Un exemple pour *ipnat* :

Pour "mapper" sur l'adresse externe tout le trafic qui n'est pas implicitement défini :

```
map xl1 "adresseIP/netmask" -> "adresseIP xl1/netmask"
```

Enfin, les machines du réseau local devront bien évidemment avoir comme *route* l'adresse IP locale de la passerelle (dans notre exemple, celle de la première interface), et comme adresse DNS celle fournie par le prestataire.

Reste l'essentiel : testez de l'intérieur à partir d'une autre machine du réseau local équipée de *nmap* ou de *nessus*. Si possible, testez aussi de l'extérieur par l'intermédiaire de votre prestataire, par exemple... si vous pouvez lui faire confiance !

Tout ceci reste très basique et vous pourrez aller beaucoup plus loin avec IPFilter, selon la configuration de votre réseau. IPFilter

ET ALORS ?

Cet article se limite malheureusement à un survol de toutes les possibilités offertes par la famille des BSD libres pour ce qui concerne la sécurité. L'apport d'OpenBSD à la communauté Unix, libre ou propriétaire, est indéniable, et c'est particulièrement applicable à FreeBSD, NetBSD... et Darwin (et à Mac OS X par extension).

C'est parfaitement comparable à l'apport de BSD au monde Unix. Que serait TCP/IP sans les "sockets" de BSD ?

Au risque de faire hurler les foules, la famille des BSD libres est certainement plus avancée que Linux sur le plan de la sécurité "par défaut". De plus, sa caractéristique majeure se nomme stabilité. Un nouveau noyau n'apparaît pas tous les 15 jours et ça n'empêche en rien le développement et les améliorations qui vont avec... et ce noyau n'a pourtant rien de monolithique.

C'est aussi dans la famille en question qu'IPv6 est probablement à son stade le plus avancé. Pour ce qui concerne IPSec et les VPN, c'est pareil. Le projet KAME fait vraiment un superbe travail dans ce domaine.

La panoplie d'outils est aussi particulièrement conséquente. Voir la section `security` dans `/usr/ports`. Bien évidemment, les outils spécialement développés pour Linux sont pour la plupart utilisables grâce à l'émulation binaire.

Tout ceci s'applique bien sûr à la famille BSD libre dans son intégralité.

Un autre avantage considérable vient du nombre de plates-formes susceptibles de supporter les systèmes libres de BSD. Quelle que soit la machine et son processeur, vous trouverez toujours (ou presque) une version de NetBSD, FreeBSD ou OpenBSD capable de lui servir de système d'exploitation. Il peut s'agir de la plus performante des stations Alpha comme d'une pièce de musée équipée d'un vieux Motorola 68***. Il en sera de même pour une ancienne station SPARC Ultra1 ou Ultra2 de Sun, ou d'une "vieille" HP300 ou 700 à processeur PA-RISC.

Puisque le sujet de l'article est quand même FreeBSD, ajoutons que la version 5 devrait apporter de nombreuses améliorations à un produit déjà pétri de qualités. Si vous aimez la découverte, rien ne vous empêche de le tester dès maintenant.

Un autre sujet qui passionne les mêmes foules que ci-dessus concerne les "fameuses" licences. Les licences BSD sont trop laxistes aux yeux des "intégristes" de la GPL. Soit. Dans les faits, ça change quoi ? Combien de lignes de code GPL se trouvent dans des produits "propriétaires" sans que quiconque le sache ? Quel est le "petit" développeur qui tentera de faire un procès à une grosse société dont il soupçonne qu'elle utilise son travail sans le chanter sur les toits ? Quelle chance de gagner ce procès selon le pays concerné ? A part une solution à l'amiable grâce à l'appui de la communauté, que reste-t-il ? Même chose pour les licences concernant la documentation. Alors, la GPL existe et c'est très bien sur le fond : quid de la forme ? Arrêtons donc de nous voiler la face : la liberté c'est

pourrait par exemple servir de proxy "transparent" pour le serveur de courrier évoqué plus haut. Il suffirait alors d'ajouter `rdr` (pour redirection) dans `/etc/ipnat.conf` et `pass in` dans `/etc/ipf.conf` de cette manière :

`rdr xl1 "adresseIP/netmask" port 25 -> "adresseIP xl0" port 25` (remplacer ce qui est entre guillemets par les adresses réelles)

```
pass in quick on xl1 proto tcp from any to any port = 25 flags S keep frags keep
state
```

Enfin, pour bénéficier d'un contrôle plus important, vous pourriez envisager d'utiliser des outils de connexion client/serveur comme *LineControl* ou le "vieux" *mserver*, par exemple. Tout dépendra des systèmes d'exploitation des clients, du type de connexion pour pouvoir s'adapter à votre cas. Et, reconnaissons-le, c'est plutôt artisanal comme méthode (ce qui n'a rien de péjoratif envers ces outils).

En résumé, vous verrez, il y a de quoi s'occuper.

CONCLUSION

formidable, mais elle a un prix, celui de ses inconvénients ! Enfin, le mot "honnêteté" n'est-il pas tombé en désuétude depuis longtemps ?

Après avoir fait les beaux jours de certains éditeurs d'Unix propriétaires, comme Sun par exemple, BSD a été remplacé par System V dans la plupart des cas. Le monde BSD fait aujourd'hui partie des "oubliés", et dans un sens, c'est tant mieux. La médiatisation outrancière de Linux, qui dit tout et n'importe quoi, ne me paraît pas très saine. Le logiciel libre, les systèmes en particulier, n'ont pas été prévus à l'origine pour devenir un modèle économique, même si ceux qui veulent faire croire le contraire sont de plus en plus nombreux. Si vous voulez savoir comment transformer un état d'esprit, voire une philosophie, en "machine à fric", suivez la saga du pingouin ! Pourtant, il se pourrait bien qu'il finisse tout seul sur la banquise quand la communauté de développeurs en aura marre de travailler pour que d'autres gagnent de l'argent sur son dos.

Sans compter que si l'ogre de Redmond sent vraiment un danger, il pourrait bien fermer les vannes. Il en a largement les moyens...

Wait and see !

Georges Tarbouriech - gt(at)linuxfocus.org

RÉFÉRENCES

- A tout seigneur tout honneur : **http://www.freebsd.org**
- Le reste de la famille : **http://www.netbsd.org** **http://www.openbsd.org** **http://developer.apple.com/darwin/**
- Sans doute le document le plus complet sur la sécurisation de FreeBSD et en français qui plus est : **http://minithins.net/papers/FreeBSD.txt**
- Une très bonne source d'informations : **http://www.daemonnews.org/**
- Le réseau d'O'Reilly possède toute une section sur la famille des BSD libres. Vous y trouverez un document intéressant sur IPSec avec FreeBSD : **http://www.onlamp.com/bsd/**
- Si vous n'êtes pas abonnés au magazine SysAdmin, certains articles sont consultables en ligne. En particulier, si vous allez à la rubrique "archives", dans May 2001, vous trouverez un article "Securing FreeBSD using jail". **http://www.samag.com**
- Une série de "HOWTO" pour FreeBSD : **http://www.freebsd-howto.com/HOWTO/**
- Un document comparant MySQL sous Linux et FreeBSD :**http://jeremy.zawodny.com/blog/archives/000203.html**
- Enfin, on n'est jamais si bien servi que par soi-même... mais j'ai honte : FreeBSD, une véritable alternative **http://www.linuxfocus.org/Francais/September2002/article260.shtml**
- Unix libre : les BSD(s) **http://www.linuxfocus.org/Francais/January2003/article276.shtml**
- Strictement rien à voir avec le sujet... quoique ! Pour certains, un délire paranoïaque, pour d'autres une des éventualités à envisager : **http://www.pbs.org/cringely/pulpit/pulpit20010802.html**
- Deux exemples d'outils de connexion client/serveur adaptés à une passerelle : **http://linecontrol.sourceforge.net/** **http://w3.cpwright.com/mserver/index.html**

Réseaux sans fil : menaces,

Fiche sécurité élaborée par le CLUSIF

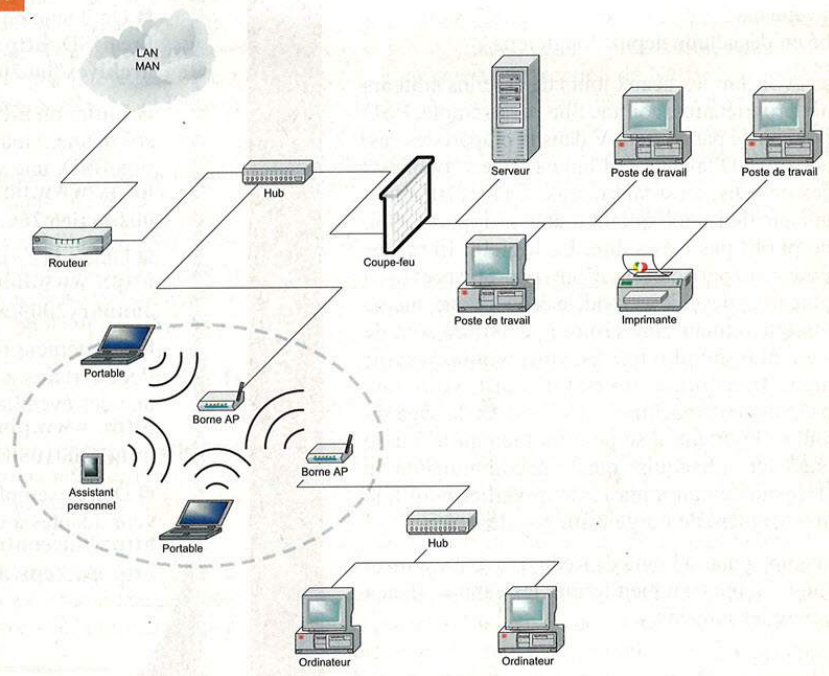
***Le présent document a un double objectif : d'une part sensibiliser les entreprises et les administrations aux risques inhérents au déploiement d'un réseau sans fil (RSF) de type 802.11b et d'autre part, présenter les solutions qui permettent d'augmenter la sécurité pour ce mode de connexion. Ce document ne traite pas des autres normes, telles que : 802.11a, 802.11i, 802.1x, Hiperlan, etc. ni des infrastructures personnelles ou des* hot spots.**

INFRASTRUCTURE

Le RSF de type 802.11b repose sur une norme. Il s'agit d'un protocole de transmission radio de proximité sur des canaux préétablis. Le RSF présente des intérêts immédiats : simplicité de déploiement, mobilité des employés, interopérabilité des équipements, faible coût en comparaison de l'installation et de la maintenance d'un réseau câblé. Il faut le considérer *a priori* comme un réseau ouvert au public mais également comme une "bulle de risque" en 3D : en effet, la connexion peut se faire de la rue, d'un autre étage, voire d'un avion (*warflying*, cf Références). En conséquence, toute connexion non maîtrisée peut remettre en cause la sécurité de l'ensemble du système d'information.

La facilité et le faible coût de mise en œuvre sont à mettre en perspective avec l'indispensable renforcement de la configuration préexistante de sécurité. Cette démarche sera identique pour tout équipement portable récent qui possède, par défaut, la possibilité de connexion à un réseau environnant ou à d'autres équipements ayant la même ressource (connexion *ad hoc*).

1



Dessin d'Infrastructure typique

enjeux et parades

MENACES ET IMPACTS

Menaces	Vecteurs	Moyens	Finalités	Exemples
Interception	Flux de connexion et de contrôle	■ Crypto-analyse (*crack*, *brute force*) ■ Man in the Middle (MiM) ■ Mascarade/Rejeu ■ Altération de message	■ Atteinte DIC* contre les données du S.I. ■ Atteinte ressources internes du S.I. ■ Atteinte ressources de connexion du S.I.	■ Consultation des serveurs de données ■ Internet gratuit
	Flux de données	■ Captation ■ Fausse borne AP (MiM) ■ Altération de message	■ Divulgation ■ Exploitation frauduleuse	■ Données financières de caisses enregistreuses
Disponibilité	Equipements	■ Déni de service (logique) ■ Brouillage par susceptibilité électromagnétique	■ Indisponibilité du RSF ■ Indisponibilité de ressources véhiculées via le RSF	■ Caméra de vidéo-surveillance ■ Sondes de détection
War-Xing	■ War driving ■ War driving + géolocalisation (GPS) ■ War chalking	■ Détection de proximité	■ Opportunité ultérieure d'interception ou d'indisponibilité	■ Cartes sur Internet ■ Marquage au sol
Attaques ad hoc	■ Flux de connexions ■ Flux de données	■ ARP cache poisoning ■ Man in the Middle (MiM)	■ Génération de pertes de paquets réseau ■ Crash de machine	■ Accès au réseau filaire ■ Attaques DoS classiques

*Disponibilité, Intégralité, Confidentialité.

Commentaires

Le war-Xing ne constitue une menace qu'à partir du moment où le RSF est mal sécurisé.

Les technologies de détection de proximité ou d'interception s'appuient sur des logiciels de scanning. Les scanners actifs engagent un dialogue avec le réseau : requêtes d'identification ou de connexion. Les scanners passifs se contentent d'écouter les différents canaux. Ces équipements sont extrêmement portables, il peut s'agir d'un PDA et d'une antenne bricolée d'apparence anodine.

Tous les secteurs d'activité sont concernés par cette exposition aux risques. A titre d'exemple, les PME de par leurs faibles engagements en moyens de sécurité ; les institutions financières, le monde médical et celui de la recherche-développement en raison du caractère confidentiel-stratégique des données traitées.

CONTRE-MESURES

Ces contre-mesures sont nécessaires en raison de la faiblesse de l'implémentation du protocole. Aujourd'hui, un système de firewall (DMZ) doit s'intercaler entre le RSF et le réseau local préexistant.

Il serait vain de refuser systématiquement cette technologie. S'il y a pression des utilisateurs ou de la direction, il est préférable d'en contrôler le déploiement pour éviter le constat ultérieur d'installations "sauvages" (comme ce fut le cas pour les modems au début de l'Internet grand public).

Soulignons que certains environnements industriels et médicaux ne se prêtent pas à l'utilisation des cartes et bornes AP en raison de susceptibilité électromagnétique ou d'environnement trop contraignant (chaleur, poussière, etc.). Les équipements grand public ne sont pas suffisamment "durcis" et il serait extrêmement risqué de déployer de telles ressources sur des systèmes temps réel ou pour des systèmes de détection (sécurité incendie). Partout ailleurs, des problèmes de couverture géographique et/ou de brouillage peuvent survenir en présence de matériaux entravant la propagation (béton, treillis métallique) ou d'équipements rayonnants (machines outils, four à micro-onde).

Si une architecture RADIUS ou VPN est déjà en place, le déploiement d'un réseau sans fil peut être envisagé sans un surcoût important tout en maintenant la protection du réseau local. Dans les autres cas, et notamment pour les PME, les compétences sécurité et les investissements complémentaires doivent être pris en considération en début de projet. Une solution intermédiaire consisterait à seulement commencer le déploiement du RSF à partir de normes plus récentes, par exemple 802.11i.

RECONFIGURATION DES ÉQUIPEMENTS

Les mesures de reconfiguration sont généralement peu coûteuses. Elles concernent essentiellement un re-paramétrage des équipements

1 Architecture et topologie

- Deux architectures s'opposent. Soit chaque borne AP est reliée au réseau filaire : les contraintes sont identiques à celles d'un réseau câblé. Soit toutes les bornes AP sont reliées entre elles et une seule est reliée au réseau. Dans ce cas, le trafic d'authentification va dégrader le débit. Quelle que soit la solution choisie, la liaison avec le réseau filaire doit se faire *via* une architecture firewall.
- La disposition physique des bornes AP a des conséquences. Lorsque c'est possible, choisir des antennes au lobe de rayonnement directif et placer les AP en des lieux qui réduisent les propagations en dehors du volume souhaité, par exemple en hauteur ou dans le coin d'une pièce. C'est ainsi qu'il faudra être vigilant quant aux infrastructures métalliques ou gaines de ventilation qui agissent comme des guides d'onde et propagent le signal au-delà du périmètre souhaité.
- Attention aux interférences entre réseaux dans un même espace. Il est possible de répartir l'attribution des bandes de fréquence entre les différentes infrastructures. Le RSF est aussi susceptible de brouillage par d'autres équipements (par ex. Bluetooth, four à micro-onde, répartiteurs TV... toute transmission sur la bande 2,4 GHz).

2 Points d'accès (bornes AP)

- Réduire la puissance d'émission par une commande logique.
- Privilégier le paramétrage en mode local (ex. par le port série) et donc, éviter les commandes à distance. Ce choix doit être fait au moment de l'achat ; le surcoût est d'environ 5 %. Ce mode d'administration reste très contraignant si on opte pour des authentifications *via* l'adresse MAC ou un système RADIUS en raison des mises à jour de listes (ACL). Des problèmes d'interopérabilité peuvent se poser pour des matériels de technologies différentes.
- Activer le WEP dans sa version 104 bits ou même dans sa version 40 bits. Ce dernier constitue déjà un durcissement de la sécurité même s'il existe quelques logiciels permettant la crypto-analyse des clefs, y compris sur WEP2.
- Choisir un SSID : éviter la diffusion publique qui implique un *broadcast* plus lointain. Choisir un nom de réseau qui ne rappelle pas celui de l'entreprise ou de l'activité pour ne pas susciter la tentative d'intrusion. Ne pas laisser le SSID par défaut de l'AP qui renseigne alors sur la marque de fabrique (et ses éventuelles vulnérabilités). Attention également à l'emploi d'un serveur DHCP qui invite à la connexion toute station appartenant au groupe SSID.
- Filtrer par l'adresse MAC : solution seulement envisageable pour un parc réduit de stations connectées car sinon, la mise à jour des tables d'adresses devient rapidement ardue.

3 Equipements portables

- Activer par une commande logique les modes BSS et ESS pour prévenir la connexion sur une fausse borne AP (scénario *man in the middle*).
- Si un portable n'est pas censé se connecter à un réseau sans fil, il est préférable de désactiver la ressource pour prévenir une connexion accidentelle ou sur une fausse borne AP qui se situerait à proximité de l'entreprise ou dans un lieu public. En fonction des équipements, déplacer un cavalier sur la carte mère, désactiver la ressource au niveau du BIOS ou désactiver le pilote de périphérique au sein du système d'exploitation (par exemple, Windows XP intègre par défaut la gestion des cartes sans fil).

RENFORCEMENT DES RESSOURCES DE SÉCURITÉ

Les actions à mener sont des domaines humains, organisationnels et techniques.

1 Sensibilisation de tous les utilisateurs quant aux enjeux

■ Les plus concernés sont souvent les informaticiens, les commerciaux, les consultants, la Direction, susceptibles d'être attirés par une technologie très conviviale. Il est recommandé de formaliser la politique de déploiement.

■ Le déploiement d'un réseau sans fil domestique est un autre point de compromission pour des équipements à usage professionnel qui s'y connecteraient. En l'absence d'un périmètre de sécurité, les répertoires en mode partagé sont, par exemple, un moyen pour accéder à des informations confidentielles de l'entreprise ou de l'administration.

2 Changement dynamique des clefs WEP

Il est possible de provoquer le changement périodique de la clef WEP pour prévenir ou réduire les possibilités d'interception. Si le poste portable se prête souvent à une telle modification, d'autres équipements, comme les tablettes, risquent de poser des problèmes. L'interopérabilité de tous les équipements peut être dégradée.

3 Authentification durcie

■ Mise en place d'un système RADIUS, d'une architecture VPN, d'IPSEC, etc.

■ Emploi du protocole 802.1x (LEAP, PEAP) qui nécessite une expertise plus forte.

→ Le RFC (*Request for Comments*) 2284 pour EAP (*Extensible Authentication Protocol*) explique ce standard pour l'authentification, le contrôle de l'utilisateur et le changement à la volée des clés d'encryptage.

→ PEAP et TTLS sont 2 standards concurrents poussés par différents acteurs pour améliorer EAP.

4 Géolocalisation des bornes

Cette cartographie peut être envisagée périodiquement. Elle nécessite l'installation d'un équipement GPS pour réaliser le positionnement des bornes.

5 Enregistrement des journaux de connexion

Certaines bornes permettent la transmission de logs pour un traitement centralisé. Ils seront ainsi sauvegardés pour analyser des situations atypiques.

6 Correction des bogues logiciels

Des dysfonctionnements logiciels, voire des faiblesses de sécurité, peuvent être corrigés par la mise à jour des programmes des bornes AP.

CADRE RÉGLEMENTAIRE

Il est nécessaire de consulter périodiquement l'ART car le cadre légal évolue.Attention aux actions d'audit : le risque d'intrusion sur un autre RSF à proximité existe. Il est donc impératif d'identifier le réseau à auditer et de manipuler avec circonspection des scanners actifs qui, par définition, vont s'introduire sur tous les réseaux accessibles dans l'espace environnant.

Rappelons également que l'attaque "par rebond" reste une opportunité : le RSF va servir de premier point de connexion pour une attaque sur Internet. L'entreprise ou l'administration se trouve donc à l'origine de l'acte de malveillance.

A ce jour, il n'existe pas encore de jurisprudences concernant l'application des règles de déploiement ou les actions malveillantes réalisées.

pascal.lointier@clusif.asso.fr

RÉFÉRENCES

RÉGLEMENTATION ET CONSEILS DE SÉCURITÉ

http://www.etsi.org
http://www.art-telecom.fr/
http://standards.ieee.org/wireless
http://www.wi-fi.com/
http://www.hsc.fr/ressources/presentations/
http://www.certa.ssi.gouv.fr/site/CERTA-2002-REC-002/

LOGICIELS ET PROCÉDURES D'ATTAQUE

http://www.netstumbler.com/
http://www.bretmounet.com/ApSniff/
http://wepcrack.sourceforge.net/
http://www.warchalking.org/

Protégez votre infrastructure

Pourquoi parler d'IPv6 [4] dans un dossier sur la sécurité réseau ? Les objectifs premiers d'IPv6 sont d'agrandir l'espace d'adressage de manière conséquente (*), de rendre l'en-tête IP plus simple, et également de supporter l'authentification et le chiffrement. Après les projets plutôt académiques, les opérateurs et les entreprises commencent ces derniers mois à investir plus de temps et de ressources dans les projets liés à IPv6. Les risques de sécurité ne changent pas grandement avec IPv6 mais la période de transition, les "bricolages" mêlant IPv4 et IPv6, et les applications (non-) "IPv6 ready" permettront de contourner certaines sécurités en place depuis bien longtemps, de profiter de nouveaux canaux cachés, ou encore de (re)découvrir problèmes et failles qui donneront une impression de déjà-vu.

INTERNET PROTOCOL VERSION 4

LA FIN D'IPv4 ?

Il y de ça presque une dizaine d'années maintenant, la pénurie d'adresses IP commençait déjà à préoccuper la communauté. Pour donner à IPv6 le temps de se mettre en place, des mesures plus strictes pour l'allocation d'adresses ont été introduites. L'approche "classful" a été remplacée par l'approche CIDR [2] (*Classless Internet Domain Routing*) et les réseaux locaux d'entreprise ont été placés derrière un équipement effectuant une traduction d'adresse (NAT) entre une ou plusieurs adresses publiques et des adresses dites privées ("RFC1918"). L'introduction de la traduction d'adresse a mis fin au concept de communication "end-to-end" ; applications et protocoles ont dû être adaptés pour continuer à fonctionner lorsque cela était possible (IPsec par-dessus UDP ou TCP par exemple). La traduction d'adresses dite *many-to-one* (N→1) n'est adéquate que lorsqu'un client initie une communication "sortante" vers un serveur, et n'autorise pas les connexions dites "entrantes". Les protocoles courants et plus complexes (FTP, H.323, etc.) sont traités de manière spécifique lors de la traduction et cela introduit une complexité sans cesse croissante.

La pénurie des adresses IP n'est que l'élément le plus visible de la croissance de l'Internet. Les numéros de systèmes autonomes (ASN) et l'explosion des tables de routage BGP ont aussi un effet sur les réseaux des opérateurs et l'évolution des protocoles : le champ contenant l'AS n'est codé que sur 16 bits, et une table de routage complète contient entre 110 000 et 120 000 routes à ce jour, ce qui a obligé bon nombre d'opérateurs à augmenter la mémoire sur une bonne partie de leurs routeurs.

(*) Déjà, à l'époque de la création de l'Internet, l'adressage sur 32 bits dépassait les prévisions les plus optimistes et certaines personnes trouvaient même cela quelque peu fantaisiste, tout comme la barrière des 640 Ko sur les PC dans les années 80... Et puis, qui se souvient encore des numéros de téléphones à moins de 8 chiffres ? ;-)

réseau IP : IPv6 et IP anycast

IPSEC

Le support d'IPsec, qui est "optionnel" dans IPv4, est obligatoire dans IPv6. Les informations relatives à IPsec se trouvent dans l'en-tête d'extension AH (*Authentication Header*) et/ou ESP (*Encapsulating Security Payload*). Les modes tunnel et transport sont supportés. Les implémentations IPv4+IPsec et IPv6 ne sont généralement pas conçues pour être interopérables, et la majorité des attaques contre IPsec sont toujours valides avec IPv6. Le chiffrement entre hôtes se retrouve également simplifié avec IPv6, alors qu'avec IPv4+IPsec, la tendance était plutôt de chiffrer entre deux passerelles ou entre un hôte et une passerelle. Pour plus d'informations sur IPsec, je vous invite à lire l'article sur le sujet dans MISC 2.

L'adresse source des datagrammes générés par les agents utilisés lors de larges dénis de services (voir l'article sur ce sujet dans MISC 4) est souvent modifiée et ne reflète pas la source réelle. L'arrivée d'IPv6 et le support intégré d'AH ne change(ra) pas grand-chose à ce problème.

MOBILE IP

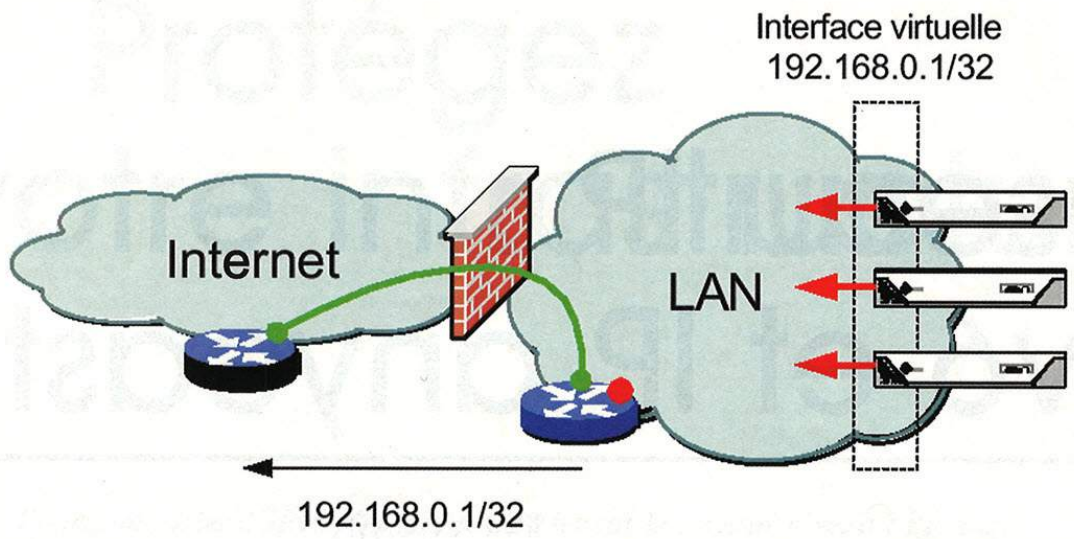
Le concept de mobilité IP (MIPv6 pour Mobile IPv6) peut se comparer à la fonction plus connue de "roaming" des téléphones cellulaires de type GSM ou xDMA. La mobilité IP est déjà possible avec IPv4, mais reste complexe. IPv6 simplifie la mise en place : l'équipement obtient une adresse temporaire dans le réseau étranger, mais conserve son adresse "de base" et informe un agent sur son réseau d'origine ("Home Agent") de sa nouvelle adresse. Le trafic ne passe plus systématiquement par l'agent, mais transite directement entre l'hôte distant et l'équipement ("Direct Routing"). Cette optimisation de route se fait grâce à l'échange de messages MIPv6 BU (*Binding Update*). Les différentes attaques (faux BU, interception, déni de service, etc.) peuvent se contrer en utilisant différentes méthodes et technologies (IPsec, PKI, Return Routability). La sécurité de MIPv6 est le sujet de nombreux *drafts* et autres propositions.

IPv4 ANYCAST

Contrairement à *IP multicast*, *IP anycast* ne repose pas sur l'utilisation d'une plage d'adresses spécifique, n'est pas un protocole IP particulier, et ne nécessite pas d'avoir des fonctionnalités spécifiques, que ce soit au niveau réseau ou au niveau des équipements clients ou serveurs. En fait, IP anycast est plus un concept de configuration (adressage et routage) qui, bien que cité dans beaucoup de documentations et RFC datant de vingt ans, ne commence qu'à être utilisé depuis quelques années maintenant, en particulier pour les réseaux de distribution de contenu ou les serveurs DNS.

L'approche est la suivante : plusieurs serveurs, en général distribués au sein du réseau de l'opérateur, voire distribués au niveau de l'Internet "partagent" la même adresse IP. Le routage s'occupe du reste : le trafic à destination de cette adresse IP est normalement "routé" vers le serveur le plus proche au sens réseau. Couramment, ce n'est pas un sous-réseau complet qui est annoncé de cette manière mais uniquement une adresse d'hôte (un /32 en notation CIDR IPv4), qui est l'adresse la plus spécifique possible. Pour éviter les trous noirs lorsqu'un serveur est indisponible, on évite de placer des routes statiques, mais on utilise plutôt un IGP [1] (en rouge dans le schéma ci-après) entre le serveur et un routeur qui lui redistribue la route dans BGP (en vert), par exemple. Ce routeur, pour limiter les risques ainsi que l'introduction d'une complexité inutile (autoriser ou relayer un IGP au niveau d'un équipement de filtrage), devrait se trouver au niveau de la ferme de serveurs, derrière un pare-feu. Pour aller encore plus loin au niveau de la redondance, et si les serveurs supportent une fonctionnalité de type IP MultiPathing [12], un cluster de ces derniers peut également partager cette adresse IP et voir la charge répartie entre les différents serveurs. Contrairement à IP multicast, il n'y a pas de notion d'appartenance à un groupe ("Group Membership Management") ni d'authentification dans IPv4 anycast. "N'importe qui" pourrait annoncer cette même adresse et le routeur n'a pas de moyen de valider cette information.

1



Un effet de bord de l'approche *anycast* est qu'elle agit comme un aimant pour les dénis de service. En fonction de la distribution des sources au niveau de l'Internet, soit un seul serveur et l'infrastructure réseau frontale sera attaquée et affectée si les sources sont concentrées sur une plaque, soit l'impact sera bien moindre ou nécessitera un nombre encore plus conséquent de sources si celles-ci sont vraiment distribuées.

Les challenges pour le déploiement d'un service reposant sur IP anycast sont, entre autres, de placer correctement ses serveurs au plus proche de la demande, d'arriver à gérer la distribution du trafic et du contenu, et pour les protocoles plus orientés "session" et les connexions longues, d'arriver à maintenir les informations de session en environnement distribué. Et finalement, d'arriver à identifier le serveur qui cause problème lorsque l'utilisateur n'est pas capable de fournir un *traceroute* ou donner son adresse IP ;-)

INTERNET PROTOCOL VERSION 6

L'EN-TÊTE IPv6

IPv6 ne se limite pas à passer de 32 à 128 bits ainsi qu'à une représentation héxadécimale de l'adresse pour répondre aux besoins (ordinateurs, mais également réseaux cellulaires de 3ième génération, etc.). Au-delà du changement de l'en-tête pour en accélérer le traitement, faciliter la gestion de la qualité de service et la mobilité, un nombre conséquent de changements ou de versions spécifiques à IPv6 de la majorité des protocoles courants ont été développés : ICMPv6, OSPFv3, BGP4+, DHCPv6, etc.

Au risque d'énerver les puristes, une description simplifiée des champs et leur(s) fonction(s) :

- La Version est 6 (pour IPv6).
- Traffic Class est comparable au champ ToS (*Type of Service*) d'IPv4.
- Flow Label identifie un paquet faisant partie d'un flux.
- Payload Length donne la taille du datagramme.
- Next Header identifie le protocole encapsulé ou l'en-tête d'option chaîné.
- Hop Limit est comparable au TTL d'IPv4.

A la différence d'IPv4, les routeurs IPv6 ne fragmentent plus les datagrammes et n'inspectent certaines options que si elles les concernent.

ADRESSAGE IPv6

L'adresse de la carte Ethernet (64 bits) peut être placée dans l'adresse IPv6 d'un hôte, ce qui simplifie l'auto-configuration vu que ces adresses sont censées être uniques. Cela pose également un problème de "vie privée", l'adresse IPv6 pouvant servir à tracer un utilisateur dans le temps (un peu à l'image d'un cookie HTTP). Mais il est possible, au contraire, vu le nombre d'adresses disponibles d'en changer plus souvent dans le temps pour contrer cette traçabilité (voir également [7]). Il y a fort à parier que d'autres identifiants "uniques" seront utilisés pour tracer un utilisateur (numéro de série, code IMEI [5], identifiant IEEE EUI64 [6], etc.).

Version (4 bits)	Traffic Class (8 bits)	Flow Label (20 bits)	
Payload Length (16 bits)		Next Header (8 bits)	Hop Limit (8 bits)
Source Address (128 bits)			
Destination Address (128 bits)			

En-tête IPv6 (aligné sur 32 bits)

Les adresses IPv6 unicast introduisent également la notion de portée : *link-local* est une adresse locale uniquement, configurée automatiquement (FE80:0:0:0:<identifiant d'interface basé sur l'adresse MAC> ou <adresse annoncée en multicast>:<ID MAC>), *site-local*, dont la portée se limite à un "site" n'est pas activé par défaut. Dans le premier cas, le trafic n'est pas routé ; dans le deuxième cas, il n'est pas routé hors du "site". Le risque est que des équipements vont se fonder sur cette notion de portée "géographique" pour les mécanismes de contrôle d'accès à une ressource, à une application, voire à l'interface d'administration. Il ne va pas sans dire qu'en fonction de la taille et du découpage du réseau, cela peut devenir rapidement quelque peu abstrait.

Il est aussi à prévoir que les interfaces réseaux des équipements, tels que les serveurs qui ne sont de nos jours habituellement configurés qu'avec seule adresse IP, se retrouvent avec, par exemple, une adresse IP par service ou assignée par le système d'exploitation de manière dynamique (i.e. "à la demande"). Le concept TARP [11] repose également sur l'utilisation d'adresses multiples par hôte pour améliorer le filtrage au niveau des pare-feux. Ces derniers, et les autres équipements de sécurité (IDS par exemple), devront pouvoir comprendre et analyser les en-têtes chaînés, s'adapter à un environnement où le chiffrement des échanges sera plus simple, et espérons-le, plus courant.

L'adresse de diffusion (*broadcast*) n'existe plus dans IPv6 et est "remplacée" par l'adresse *multicast*, ce qui introduit une meilleure utilisation des ressources. A la différence du multicast avec IPv4, où la portée est limitée par le TTL des datagrammes, avec IPv6, celle-ci est géographique (local, site, global, etc.).

Certaines fonctionnalités comme le routage (défini) par la source, que l'on a l'habitude, pour des raisons de sécurité, d'interdire par défaut sur les routeurs, est nécessaire pour Mobile IPv6.

Comme il existe différentes manières d'écrire une adresse IPv6 (pas de case, les 0 "en tête" sont facultatifs, un champ où tous les bits sont à zéro peut être noté ::, etc.), il y a fort à parier que cela permettra une nouvelle fois de contourner certains IDS ou encore de "masquer" l'adresse (URL) d'un site dans un message électronique publicitaire non sollicité…

IPv6 NEIGHBOR DISCOVERY

La découverte des voisins s'effectue grâce au protocole IPv6 ND (ICMPv6 Neighbor Discovery, voir [8]) et comprend l'acquisition des routeurs, des adresses des équipements (mécanisme ARP d'IPv4), la gestion de l'auto-configuration, la détection des conflits, ainsi que des informations relatives à leur disponibilité et à leur configuration. IPsec est préconisé pour sécuriser ce protocole et authentifier les échanges ainsi que les hôtes, mais les problèmes (administratifs) de gestion des clés se révèlent être, comme rencontré couramment, une lacune. Un grand nombre de *drafts* proposent d'utiliser GCA (*Cryptographically Generated Addresses*) à la place d'IPsec pour apporter la fonction d'authentification. Le champ Hop Limit doit être à 255 (valeur maximale), cela limite les échanges ND au brin local. Cette technique (TTL à 255) est également suggérée dans un draft concernant les mécanismes de sécurité de BGP : cela évite qu'un routeur, qui n'est pas un voisin direct, puisse interférer avec la session BGP (voir l'article sur les protocoles de routage dans MISC 4). Une attaque pourrait par exemple consister à encapsuler le datagramme erroné pour contourner le décrémentation du TTL par chaque routeur traversé.

IPv6 ND faisant partie des fonctionnalités de base, il y a fort à parier que beaucoup de politiques ou de règles de filtrage laissent passer ce protocole "par défaut", un peu à l'image des règles

001	TLA	NLA	SLA	Interface ID

Adresse IPv6 unicast

TLA (*Top Level Aggregator*) et NLA (*Next Level Aggregator*) sur 45 bits

SLA (*Site Level Aggregator*) sur 16 bits et Interface ID sur 64 bits

"implicites" longtemps pas vraiment documentées de Checkpoint FW-1 (faille RDP par exemple). Les ACL IPv6 sur les routeurs Cisco ont un "permit" implicite pour certains messages de ND.

Les "jouets" de sécurité réseau (comme arp-sk [9] par exemple) auront encore un bel avenir avec IPv6, mais beaucoup d'outils devront encore être adaptés pour supporter IPv6 et les nouveaux protocoles associés.

MIGRATION IPv4 VERS IPv6

Un nombre conséquent de mécanismes de transition d'IPv4 vers IPv6 sont disponibles : certains sont plus destinés aux hôtes comme DSTM (*Dual Stack Transition Mechanism*), d'autres concernent plus la partie réseau et architecture (tunnels, 6to4, etc.).

Un hôte avec deux piles utilisera l'une d'elles en fonction de la demande. Cela implique que, du point de vue de la sécurité, la politique de filtrage soit homogène. L'encapsulation d'IPv6 dans IPv4 (numéro de protocole IP: 41) est une alternative à l'encapsulation GRE (IP dans IP), ou via MPLS, pour mettre en place un tunnel entre deux hôtes IPv6 via un réseau IPv4. Beaucoup d'équipements de sécurité laissent passer ou ne traitent pas les protocoles "non connus", ou qu'ils ne savent pas gérer. Le protocole IP 41 en est un, et l'expérience récente d'un *honeypot* [10] en est la confirmation.

6to4 permet de connecter de façon plus dynamique deux îlots IPv6 via un réseau IPv4. Le préfixe 2002::/16 est réservé pour 6to4 et l'adresse IPv4 est placée dans l'adresse IPv6: 2002:<adresse IPv4>::/48. L'hôte n'implémente qu'une pile IPv6, le routeur quant à lui a deux piles et une adresse IPv4 "externe", qui est utilisée. Les relais 6to4 sont sensibles aux attaques de type déni de service ainsi que celles reposant sur des datagrammes avec des adresses IP trafiquées. Ils peuvent également servir de réflecteur.

Un autre protocole, TSP (*Tunnel Setup Protocol*) devrait permettre de rendre l'activation de tunnels à la demande plus simple qu'en passant via un "broker" de tunnels, ou encore passer outre les limitations introduites par NAT-PT [3] (*Network Address Translation - Protocol Translation*).

CONCLUSION

Le réseau Internet IPv6 est encore très laxiste au niveau de la sécurité : quasiment aucun FAI ne filtre les routes ou différencie *transit* et *peering*. Cela commence (heureusement) à changer, mais rappelle les débuts de l'Internet/ARPANET lorsque tout le monde faisait confiance (aveugle) à tout le monde. Bien qu'il existe des applications commerciales fondées sur IPv6, cela donne encore l'impression d'un réseau de laboratoire à l'échelle de la planète.

Au-delà des changements et des nouveautés au niveau de la pile TCP/IP, les applications clientes ou serveurs (DNS par exemple avec AAAA et A6), interfaces (API et sockets) doivent, elles aussi, être adaptées, recompilées, testées, etc. L'expérience et l'histoire prouvent que des erreurs de programmation, bien que connues depuis longtemps, sont sans cesse refaites et les failles réintroduites dans les nouveaux développements.

IPv6 introduit beaucoup de nouvelles fonctionnalités et de nouveaux protocoles, ce qui le rend plus complexe qu'IPv4, un peu à l'image de la complexité parfois inutile d'IPsec.

Nicolas Fischbach (**nico@securite.org**)

IP Engineering Manager – COLT Telecom AG

Senior Manager – European IP Engineering/Security – COLT Telecom

http://www.securite.org/nico/

RÉFÉRENCES

[1] MISC numéro 3 : *Protection de l'infrastructure réseau IP, les protocoles de routage et MPLS.*

[2] RFC 1519 : *Classless Internet Domain Routing.*

[3] RFC2766 : *Network Address Translation - Protocol Translation.*

[4] RFC 2460 : *Internet Protocol Version 6 (IPv6) Specification.*

[5] *International Mobile Equipment Identity* : **http://www.gsmworld.com/technology/gsm.shtml**

[6] EUI64 : Extended Unique Identifier (EUI)-64. Représentation: <identifiant vendeur>:FFFE:<adresse MAC>

[7] RFC3041 : *Privacy Extensions for Stateless Address Autoconfiguration in IPv6.*

[8] RFC2461 : *Neighbor Discovery for IP Version 6 (IPv6).*

[9] arp-sk : **http://www.arp-sk.org/**

[10] Honeynet : **http://www.honeynet.org/**

[11] *Transient Addressing for Related Processes : Improved Firewalling by Using IPV6 and Multiple Addresses per Host* – **http://research.att.com/~smb**

[12] IPMP : **http://docs.sun.com/db/doc/806-4075/6jd69oabv?a=view**

Bulletin d'abonnement

misc
MULTI-SYSTEM & INTERNET SECURITY COOKBOOK

A renvoyer avec votre règlement à Diamond Editions -
Service des abonnements/commandes
6, rue de la Scheer - 67600 Sélestat

☐ Oui, je souhaite m'abonner à Misc
☐ Oui, je souhaite profiter des offres de couplage

Je coche le type d'abonnement choisi :

Durée de l'abonnement	1 AN (6 N°) France	1 AN (6 N°) Etranger et DOM-TOM
Mode de Paiement	Chèque CB	CB Mandat postal
Magazine	☐ 33 Euros	☐ 45 Euros
Offres de couplage		
Magazine 11 N°s Linux Mag.+ 6 N°s LM Hors Série	☐ 79 Euros	☐ 128 Euros
Magazine 11 N°s Linux Mag+ .6 N°s Misc	☐ 83 Euros	☐ 128 Euros
Magazine 11 N°s Linux Mag +6 N°s Misc. + 6 N°s LM Hors Série	☐ 105 Euros	☐ 173 Euros

☐ Je règle par chèque bancaire ou postal
à l'ordre de Diamond Editions
☐ Paiement C.B.
N° Carte |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|
Expire le |_|_|_|_|
Date et signature obligatoires :

Nom .....
Prénom .....
Adresse .....
.....
code postal .....
VILLE .....

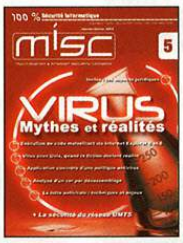
OFFRES DE COUPLAGE

+


11 N°s Linux Magazine + 6 N°s LM Hors Série

~~663,50 Frs~~ ~~101,15 €~~ En kiosque

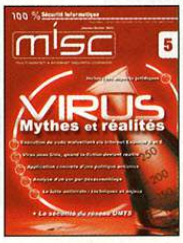
79 €

+


11 N°s Linux Magazine + 6 N°s Misc

~~722,54 Frs~~ ~~110,15 €~~ En kiosque

83 €

+


+


6 N°s Misc + 11 N°s Linux Magazine + 6 N°s LM Hors Série

~~956,71 Frs~~ ~~145,85 €~~ En kiosque

105 €

COMMANDEZ NOS ANCIENS NUMEROS

Magazine	Prix N°	Quantité	Total
Misc 1	5,95 euros		
Misc 2	7,45 euros		
Misc 3	7,45 euros		
Misc 4	7,45 euros		
Misc 5	7,45 euros		
Linux HS 8	5,95 euros		
Linux HS 9	5,95 euros		
Linux HS 10	5,95 euros		
Linux HS 11	5,95 euros		
Linux HS 12	5,95 euros		

Frais de port : France métropolitaine 3,81 euros
U.E. plus Suisse, Liechtenchtein, Maroc, Tunisie, Algérie 5,34 euros

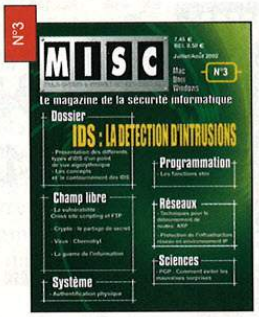
Total	
Frais de port	
Total de la commande	

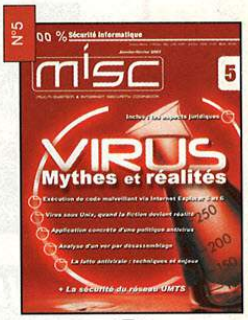
Nom .....
Prénom .....
Adresse .....
.....
code postal .....
VILLE .....

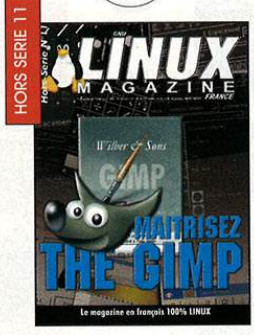
Mode de règlement

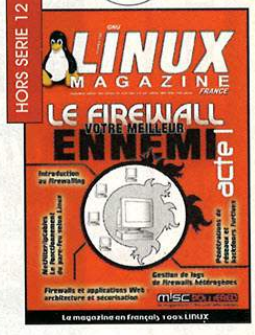
- ☐ Carte bancaire
- ☐ Chèque bancaire
- ☐ Chèque postal

Numéro :
_____/_____/_____/____
Date d'expiration____/____
Signature :









Récupérez votre code PIN ou une clé RSA avec un chronomètre

La sécurité d'un système dépend souvent de petits riens ou d'effets de bord imprévus permettant une fuite non désirée d'information. La fuite d'information que nous allons étudier maintenant est une fuite à caractère temporel et est donc mesurable à l'aide d'un chronomètre.

VÉRIFIER UN MOT DE PASSE OU CODE PIN PAR COMPARAISON SIMPLE

La technique la plus simple (ou la plus immédiate) pour vérifier un mot de passe ou un code PIN est d'utiliser, par exemple, la fonction C `strcmp(3)`. Par exemple en utilisant le programme suivant :

```
int is_password_correct(char *password, char *submitted)
{
  if (strcmp(password, submitted) != 0)
    return FALSE;
  else
    return TRUE;
}
```

Dans les sources de la bibliothèque C de GNU (`glibc-2.3.1`) on trouve le code de la fonction `strcmp(3)` (j'ai légèrement édité le code pour le faire rentrer plus facilement).

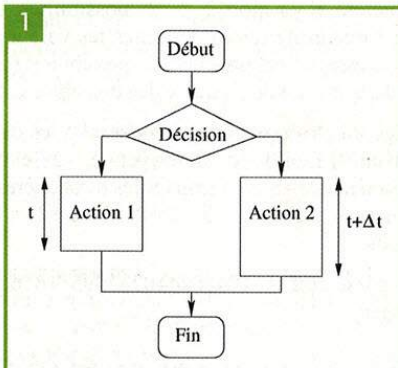
```
int strcmp (char *p1, char *p2)
{
  unsigned char *s1 = (unsigned char *) p1;
  unsigned char *s2 = (unsigned char *) p2;
  unsigned char c1, c2;

  do
  {
    c1 = (unsigned char) *s1++;
    c2 = (unsigned char) *s2++;
    if (c1 == '\0')
      return c1 - c2;
  }
  while (c1 == c2);

  return c1 - c2;
}
```

On remarque assez facilement que la durée d'exécution de la fonction `strcmp(3)` dépend des données qu'on lui passe (le temps d'exécution dépend de la longueur de la partie commune de début de chaîne). Le but du jeu maintenant est d'utiliser cette fuite d'information pour optimiser une recherche de mot de passe.

1



MODÉLISATION

Une attaque en temps (ou *timing attack*) est possible lorsqu'une opération a un temps d'exécution variable en fonction des données secrètes manipulées (voir Figure 1).

La variation du temps d'exécution est due au fait que l'algorithme utilisé fait un test sur une partie des données secrètes. Suivant le résultat de ce test, deux branches différentes du programme sont possibles. Les deux branches ont des temps d'exécution différents.

L'ALGORITHME D'ATTAQUE

Dans le cas de la vérification de mot de passe, les deux branches sont les cas `(c1 == c2)` et `(c1 != c2)`.

Si les deux premiers caractères de `password[]` et `submitted[]` sont différents, la fonction `strcmp()` sort immédiatement après avoir comparé ces deux caractères. Si, par contre, les deux premiers caractères sont égaux, la fonction `strcmp()` fera au moins une comparaison sur les deux caractères suivants et aura donc une durée d'exécution plus longue.

L'attaque consiste à faire varier le premier caractère du mot de passe soumis (de `submitted[]`) et de mesurer le temps d'exécution de la fonction `is_password_correct()`. Pour une seule valeur du premier caractère, le temps sera plus long que pour les autres cas. Cette valeur correspond au cas : le premier caractère du mot de passe est correct et la comparaison continue sur le deuxième. Nous avons donc trouvé la valeur du premier caractère. Il suffit de répéter la même opération sur les caractères suivants du mot de passe et de les trouver un par un.

COMPLEXITÉ DE L'ATTAQUE

Imaginons un PIN code (ou mot de passe) de 8 caractères. Imaginons aussi que tous les caractères peuvent être utilisés, donc 256 possibilités pour chaque caractère. Il y a alors $2^{8*8} = 2^{64}$ possibilités. Une attaque par force brute (essai de toutes les valeurs possibles) devra essayer, en moyenne, 2^{63} possibilités (en moyenne, seule la moitié de l'espace des possibles est parcouru).

Avec un chronomètre, on recherche les caractères un par un. Il faut donc, en moyenne, 128 essais (2^7) par caractère. Donc, pour trouver les 8 caractères, l'un après l'autre, il faut $2^7+2^7+2^7+2^7+2^7+2^7+2^7+2^7 = 8 \times 2^7 = 2^{10}$ essais.

On passe de $2^{63} = 18446744073709551616$ à $2^{10} = 1024$ essais.

CHRONOMÉTRAGE EFFICACE

Cette attaque s'applique difficilement sur un ordinateur actuel. Il faut mesurer un temps de l'ordre de quelques cycles machine. C'est très difficile avec un chronomètre externe, mais c'est faisable en utilisant le compteur de cycle d'un CPU Intel à partir du Pentium (voir l'instruction assembleur `rdtsc` [1]).

Cette attaque est en revanche beaucoup plus facile à réaliser sur une carte à puce lors de la vérification du code secret. Le CPU d'une carte à puce est cadencé à une fréquence de l'ordre de 5 MHz. Le plus petit des oscilloscopes actuels a une fréquence d'échantillonnage de 100 MHz, c'est-à-dire qu'un cycle CPU de la carte est représenté par 20 échantillons. Une variation d'un cycle est immédiatement visible sur l'oscilloscope.

3 CONTRE-MESURES

Il existe plusieurs techniques pour éviter ce problème.

1 TEMPS CONSTANT

Il est possible d'avoir un algorithme de comparaison qui s'exécute en un temps constant quelles que soient les données. Par exemple, l'algorithme suivant calcule la somme arithmétique du résultat du ou-exclusif des caractères deux à deux. La somme est nulle uniquement si tous les caractères de même rang de `password[]` et de `submitted[]` sont égaux deux à deux. Il faut faire attention à ce que la variable `s` qui contient la somme ne repasse pas à zéro par débordement de capacité (ce qui serait le cas si la variable s est sur 8 bits seulement).

```
int is_password_correct(char *password, char *submitted)
{
  int i, s;

  s = 0;
  for (i=0; i<PIN_SIZE; i++)
    s += password[i] ^ submitted[i];

  if (s)
    return FALSE;
  else
    return TRUE;
}
```

2 COMPARER LES HACHÉS

Une autre solution est de ne pas manipuler directement les données secrètes. Par exemple, il est possible d'utiliser une fonction de hachage (fonction à sens unique) et de comparer le haché du mot de passe soumis avec le haché du mot de passe de référence. Puisque la fonction est à sens unique, même s'il est possible de connaître la valeur du haché, il n'est pas possible de remonter à la valeur du mot de passe.

Le mot de passe de référence est haché lorsqu'il est entré et est stocké uniquement sous forme de haché. Il n'est ensuite plus jamais manipulé en clair.

Pour des raisons différentes (mais pas tant que ça), cette solution est utilisée par les systèmes Unix pour stocker et comparer les mots de passe utilisés au login (voir la fonction `crypt(3)` ou l'article [2]).

3 COMPTEUR DE RATIFICATION

Une contre-mesure classique dans le monde de la carte à puce est d'utiliser un compteur de ratification. C'est ce qui se passe avec les trois essais possibles d'un code secret de carte bancaire →

➔ par exemple. Le compteur est décrémenté à chaque mauvaise authentification. Lorsque le compteur est nul, l'authentification est rejetée même si le code PIN est correct.

Il faut cependant faire attention à la façon d'implanter la gestion du compteur. Avant de vérifier le code PIN, il faut vérifier que le compteur n'est pas déjà à zéro. La façon évidente est d'effectuer la comparaison du code PIN puis, si le code PIN est incorrect, de décrémenter le compteur. Le problème avec cet algorithme est qu'il est relativement aisé de se rendre compte que le processeur de la carte à puce effectue une écriture dans sa mémoire non volatile (EEPROM). Cette écriture consomme plus de courant et prend un temps non négligeable. Un attaquant muni d'un dispositif matériel adéquat peut couper l'alimentation de la puce dès qu'il se rend compte que la puce veut écrire en EEPROM (elle veut écrire la valeur décrémentée du compteur de ratification). Ainsi, le compteur de ratification n'est jamais mis à jour et l'attaquant peut essayer un autre code PIN.

La bonne façon de faire est de rejeter l'authentification si le compteur de ratification est nul (classique), puis de pré-décrémenter le compteur de ratification. Le code PIN est ensuite vérifié, et s'il est correct, le compteur de ratification est incrémenté (pour retrouver sa valeur antérieure) ou remis à la valeur maximale autorisée (par exemple 3 pour une carte bancaire). L'attaque précédente ne fonctionne plus.

L'attaque et la solution semblent évidentes une fois qu'on les connaît. Mais les cartes à puce programmables en Java (en fait, un sous-ensemble de Java appelé JavaCard [3]) seront utilisées par des programmeurs qui ne seront, en général, pas sensibilisés à ce problème. Ils vont donc tomber dans le piège et écrire des routines de vérification de mot de passe ou de code secret non sécurisées.

RSA

L'algorithme utilisé pour effectuer un calcul RSA est, en général, en temps non constant, et surtout avec un temps d'exécution qui dépend des données (secrètes) manipulées.

Il y a donc, ici aussi, matière à monter une attaque en temps et récupérer la clé privée RSA.

Rappel : l'algorithme de chiffrement RSA permet de chiffrer un message m par l'opération $c = m^e \bmod n$. n est le module public et d est l'exposant privé. Le but de l'attaque [4] sera de retrouver la valeur de d (exposant privé).

L'EXPONENTIATION MODULAIRE

L'algorithme classique d'exponentiation modulaire est appelé "carré et multiplié" (ou *square and multiply*) [5]. Il est simple et efficace.

```
z = 1
y = m
for i=0 to t-1 do
  if (d[i] == 1) do
    z = z * y mod n
  fi
  y = y * y mod n
od
c = z
```

L'algorithme est appelé carré et multiplié, car à chaque itération, le carré de y est calculé, et en fonction du i-ème bit de d, une multiplication (zxy) est effectuée.

Notation : z et y sont des variables intermédiaires, t est la taille de l'exposant d en bits (si d est un nombre de 512 bits alors $t = 512$), $d[i]$ est la valeur du i-ème bit de d ($d[0]$ est la valeur du bit de point faible de d), $z \times y \bmod n$ est une multiplication modulaire (c'est-à-dire le reste de la division de (zxy) par n).

Le problème vient du fait que la multiplication est effectuée en fonction de la valeur d'un bit de l'exposant privé d, et qu'une multiplication modulaire a une durée d'exécution qui dépend des données manipulées. En effet, si le produit (zxy) est inférieur à n, alors $(zxy) \bmod n = (zxy)$. Sinon, il faut effectuer une réduction modulaire qui revient à soustraire de (zxy) autant de fois n que nécessaire, jusqu'à ce que (zxy) soit strictement plus petit que n.

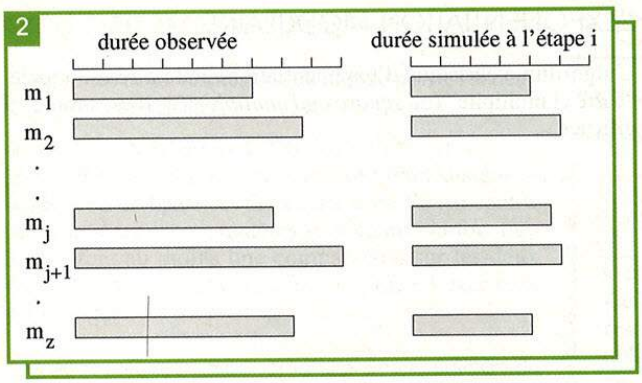
L'ATTAQUE

En observant les communications, l'attaquant a mesuré le temps de calcul du RSA pour z messages m différents. Il a donc un ensemble de z couples (m_j, t_j) avec t_j la durée d'exécution du RSA pour le message m_j (par exemple, les messages m_j sont des messages à signer envoyés à une carte à puce).

L'attaque va permettre à l'attaquant de trouver le bit d'indice i de l'exposant en connaissant les bits d'indice 0 à $i-1$. Pour trouver tous les bits de l'exposant, il suffit de commencer au bit 0 et de trouver les bits l'un après l'autre.

On suppose que l'attaquant connaît les i premiers bits de l'exposant (bits 0 à $i-1$). Pour chaque message m_j, il peut donc effectuer l'algorithme RSA et calculer y_j et z_j, qui sont les valeurs de y et z à l'étape i (avec $y_0 = m$ et $z_0 = 1$). L'attaquant calcule $z_j = z_{(i-1)} \times y_{(i-1)} \bmod n$ et mesure le temps d'exécution de cette multiplication modulaire.

Pour chaque message m_j, l'attaquant calcule le temps nécessaire pour aller jusqu'à l'étape $(i-1)$ et y ajoute le temps de calcul de z_j (branche de l'algorithme effectuée si $d[i] = 1$). Il a donc deux listes



de durées : la liste des durées t_j observée pour les messages m_j, et les durées qu'il vient de calculer par simulation pour ces mêmes messages m_j.

L'attaquant va ensuite comparer les durées mesurées et les durées qu'il a simulées de son côté (voir Figure 2). L'outil mathématique pour effectuer cette comparaison est le calcul du coefficient de corrélation. Le coefficient de corrélation donne le "degré de similitude" entre les des deux ensembles de données.

Si le coefficient de corrélation est élevé (90% ou plus), il y a une forte probabilité pour que les calculs simulés soient effectivement les calculs effectués par la carte à puce. La simulation a été faite pour le cas `d[i] == 1`, c'est-à-dire le bit `i` de l'exposant privé est à `1`. Si le coefficient de corrélation est faible, il y a une forte probabilité pour que les calculs simulés soient différents des calculs réels. Il y a de grandes chances pour que le bit `i` de l'exposant soit à `0` et que le calcul $z \times y \bmod n$ n'ait pas eu lieu.

L'attaquant connaît maintenant le bit `i` et peut passer au bit suivant avec la même méthode.

JUSTIFICATION

Le fonctionnement de la méthode peut se justifier de façon informelle en observant que la durée totale de l'algorithme RSA est la somme des durées de calculs élémentaires. Chaque calcul élémentaire a une durée d'exécution variable qui dépend du message et d'un bit de l'exposant.

Si le calcul élémentaire a une durée très variable, la durée totale va être très influencée par quelques-uns de ces calculs élémentaires. Si la similitude entre l'ensemble des durées mesurées et l'ensemble des données simulées à la première étape est faible, c'est que soit aucun des messages `m` n'a la propriété de provoquer un calcul lent (peu probable avec un grand nombre de messages `m` différents), soit le premier bit de l'exposant est à zéro.

Et ainsi de suite pour les bits suivants de l'exposant privé.

Une démonstration mathématique se trouve dans l'article [4].

AUTO-CORRECTION

Si l'attaquant obtient plusieurs fois de suite un coefficient de corrélation faible, c'est que soit plusieurs bits contigus de l'exposant sont à zéro, soit qu'il s'est trompé dans le choix d'un précédent bit de l'exposant. Il peut donc revenir en arrière dans ses choix et modifier une décision dont il n'était pas sûr (coefficient de corrélation moyen). Si l'attaquant retrouve de bons coefficients de corrélation, c'est qu'il a fait le bon choix.

L'exposant privé `d` est composé de `512`, `1024` ou `2048` bits en fonction de la taille de la clé. En moyenne, l'exposant privé `d` a autant de bits à zéro que de bits à un. Il y a donc très peu de chance pour que l'exposant contienne de grandes plages de bits à zéro.

COMPLEXITÉ DE L'ATTAQUE

L'attaque permet de trouver les bits de l'exposant un par un. La complexité est donc linéaire en fonction du nombre de bits de la clé.

En pratique, la difficulté de mise en œuvre de l'attaque est liée à la mesure précise du temps et au nombre d'échantillons (message, temps) récupérés. Plus l'attaquant possède d'échantillons, plus sa mesure du coefficient de corrélation sera précise.

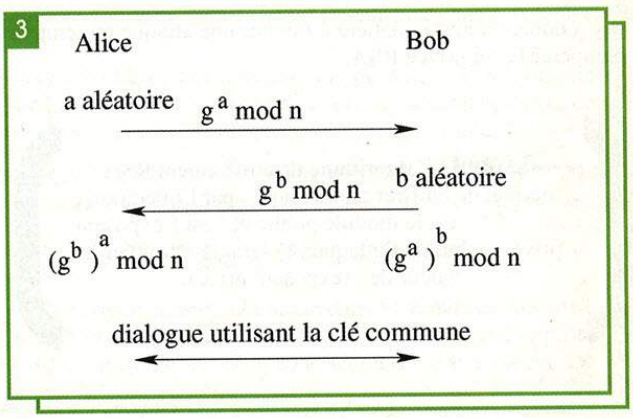
DIFFIE-HELLMAN, DSS ET LES AUTRES

RSA n'est pas le seul algorithme vulnérable par cette attaque. L'attaque fonctionne sur la réduction modulaire et peut donc s'appliquer aux autres algorithmes utilisant une réduction modulaire.

L'algorithme Diffie-Hellman permet à Alice et Bob de partager une clé de session commune $g^{ab} \bmod n$ sans avoir de secret commun au préalable (Figure 3).

Les messages $g^a \bmod n$ et $g^b \bmod n$ circulent en clair, donc l'attaquant peut les récupérer. Par contre, les secrets `a` et `b` sont aléatoires et changent à chaque nouvelle génération de clé de session. L'attaquant ne peut plus obtenir plusieurs exponentiations modulaires sur des messages différents et le même exposant inconnu. L'algorithme Diffie-Hellman est, par construction, insensible à cette attaque en temps.

L'algorithme de signature DSS (*Digital Signature Standard*) utilise aussi une réduction modulaire dont les données dépendent de la clé privée. Il s'agit d'un algorithme de signature, donc la clé utilisée est constante. Cet algorithme est vulnérable par cette attaque.



3 CONTRE-MESURES

Dans ce cas aussi, il existe plusieurs techniques pour éviter ce problème.

1 TEMPS CONSTANT

Cette solution, qui semble évidente n'est, en pratique, pas facile à réaliser. En effet, il est courant que les opérations arithmétiques complexes (multiplication, division) ne s'exécutent pas en temps constant au niveau de l'instruction CPU. Il faudrait alors recoder à la main l'opération de multiplication de façon à l'effectuer en un nombre constant de cycles CPU.

Cela peut devenir rapidement très compliqué s'il faut générer un code portable, mais qui prenne en compte les optimisations éventuelles du compilateur, les effets des caches de données et d'instructions, les branchements conditionnels et leurs effets sur le pipeline du CPU, etc.

2 MASQUAGE DES DONNÉES

Plutôt que de manipuler les données directement, il est possible, dans le cas de RSA, d'utiliser les données sous une forme différente. Le chiffrement n'est pas effectué sur m, mais sur $m' = m \times r^e \bmod n$, avec r, un (petit) nombre aléatoire tiré pour l'occasion et donc inconnu de l'attaquant.

On obtiendra donc à la sortie de l'algorithme de chiffrement $m'^d \bmod n = m^d \times r^{ed} \bmod n$. Il suffit de multiplier modulo n, le résultat, par l'inverse modulaire de $r^d \bmod n$ pour obtenir le résultat attendu, $c = m^d \bmod n$ (chiffrement du message m par la clé privée). Il est ainsi possible d'obtenir le chiffré de m sans jamais manipuler m dans la routine RSA.

Puisque r est inconnu de l'attaquant, m' est aussi inconnu de l'attaquant. Il ne peut plus simuler les résultats intermédiaires car l'algorithme utilise maintenant m'.

Cette modification n'a pas d'impact en dehors de la routine RSA puisque le résultat obtenu est toujours le même (le chiffré de m par la clé privée). L'algorithme de chiffrement est juste un peu plus lent, mais beaucoup plus sûr.

3 BOURRAGE

Une autre technique est de compléter le message à chiffrer avec des données aléatoires ou dépendantes d'un nombre aléatoire. C'est ce qui est utilisé pour le padding OAEP (*Optimal Asymmetric Encryption Padding*) [6] pour RSA.

Ce bourrage (*padding*) permet en plus d'éviter la propriété de multiplicativité de RSA qui permet de forger (sans connaissance de la clé privée) un chiffré valide à partir de deux chiffrés connus. Avec RSA, le chiffré de $m1 \times m2$ est simplement le produit $c1 \times c2$ avec $c1$ et $c2$ les chiffrés de $m1$ et $m2$. Donc, si $(m1, c1)$ et $(m2, c2)$ sont connus, il est facile de trouver le chiffré de $m1 \times m2$, de $m1 \times m1 \times m2$, de $m1 \times m2 \times m2$, et de façon générale de toute combinaison de $m1$ et $m2$.

CONCLUSION

L'écriture de routines de sécurité n'est pas chose facile. Des problèmes auxquels on ne pense pas *a priori* doivent être pris en compte et résolus. Il est donc préférable de laisser des experts écrire les parties de code sensible et d'utiliser les API fournies plutôt que de réécrire du code moins sécurisé.

Dans cet article, la fuite d'information était due à un effet de bord à caractère temporel. Optimiser le code pour gagner en performances n'est pas forcément une bonne idée. Dans un prochain article, nous étudierons une autre fuite d'information liée, cette fois, à la consommation électrique.

Georges Bart

georges.bart@free.fr

RÉFÉRENCES

[1] *Using the RDTSC Instruction for Performance Monitoring*, Intel, 1997, **http://cedar.intel.com/software/idap/media/pdf/rdtscpm1.pdf**

[2] *Cassage et durcissement des mots de passe Unix*, Denis Ducamp, MISC n°5, janvier 2003.

[3] *Java Card Technology*, SUN, 1995-2003, **http://java.sun.com/products/javacard/**

[4] Paul Kocher, "*Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS and Other Systems.*", Proc. Crypto96, Springer-Verlag, New York, 1996, pp. 104-113, **http://www.cryptography.com/timingattack/**

[5] *Fast exponentiation algorithms*, **http://dimacs.rutgers.edu/Workshops/Security/program2/quisquater/node3.html**

[6] *Optimal Asymmetric Encryption - How to encrypt with RSA*, Mihir Bellare, Phillip Rogaway, Eurocrypt 94, 1994, **http://www.cs.ucsd.edu/users/mihir/papers/oaep.html**

PEARL

6, rue de la Scheer - ZI Nord - 67603 SELESTAT

GRATUIT !

Demandez notre catalogue 96 pages, par téléphone, fax, internet ou minitel !

0,12 €/mn
N°Indigo 0 820 822 823

Nous vous proposons de nombreux autres produits pour LINUX !

Chargeur universel

Rechargez jusqu'à 8 piles en une seule fois ! Ce chargeur intelligent vous permet bien sûr de recharger vos accus vides mais également d'éviter l'effet mémoire grâce à une fonction de rafraîchissement. Chaque emplacement est géré séparément et vous pourrez visualiser l'état de chargement de vos accus grâce un LED (un LED par emplacement). ▶Compatible avec les accus Ni-MH et Ni-Cd ▶5 types d'accus supportés : AAA (maximum 6 à la fois), AA (maximum 6 à la fois), C (maximum 4 à la fois), D (maximum 4 à la fois), 9V (maximum 2 à la fois). **Réf.PE7080**

79,90 TTC € (524,11 F) Plus non incluses

Carte mémoire USB Easydisk

16 Mo	Réf. PE6071	Prix : 24,90€TTC/163,33F
32 Mo	Réf. PE6072	Prix : 29,90€TTC/196,13F
64 Mo	Réf. PE6073	Prix : 39,90€TTC/261,73F
128 Mo	Réf. PE6074	Prix : 69,90€TTC/458,51F
256 Mo	Réf. PE6075	Prix : 119,90€TTC/786,49F
512 Mo	Réf. PE6076	Prix : 239,90€TTC/1573,64F

à partir de **24,90 TTC €** (163,33 F)

Transmetteur TV/AV/télécommande

Avec ce système de transmission hautes fréquences vous allez pouvoir diffuser dans toute la maison, et même dans le jardin, les images et sons issus de vos téléviseurs, magnétoscopes, chaines hifi, micro-ordinateurs, etc... Vous pouvez choisir l'un des 4 canaux dans la plage des 2,4 GHz. Grâce à un capteur infrarouge situé sur le boîtier récepteur, vous pouvez piloter à distance de n'importe quelle autre pièce votre appareil avec sa télécommande d'origine. Si votre PC dispose d'une sortie TV, vous pourrez connecter l'ordinateur à votre TV sans tirer de câble. Vous pourrez ainsi faire des présentations ou visionner des DVD sans aucun problème.
Caractéristiques : ▶Norme vidéo: PAL ▶Connecteurs péritel et cinch ▶Livré avec câbles cinch et adaptateur péritel ▶Alimentation secteur incluse ▶Portée de 30 à 100 mètres **Réf. PE753**

79,90 TTC € (524,11 F)

Kit Bluetooth Class I

Ces deux adaptateurs Bluetooth Class I (fonctionnement jusqu'à **100 mètres**) vous permettent de mettre en place un réseau sans fil très simplement et **facilement extensible**. Il vous permettrons également de faire communiquer vos ordinateurs avec les nouveaux périphérriques bluetooth (téléphone portable, assistant personnel,...) Compatible Windows 98 ou supérieur et MAC OS X. Livré avec de nombreux logiciels. **Réf.PE145**

99,90 TTC € (655,30 F)

Boîtier USB 2.0 Alu

Ce petit boîtier USB 2.0 en aluminiun et au design très soigné s'alimente via le port USB de votre machine. Idéal lorsque vous êtes en déplacement, il vous permet de transférer des données sur votre portable sans alimentation externe. Livré avec une sacoche de transport. Possibilité d'utiliser une alimentation externe (non fournie). **Réf. PE298**

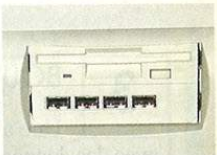
59,90 TTC € (392,92 F)

Lecteur de cartes mémoires TV

Ce boîtier externe autonome vous permet de lire vos photos (au format jpg), films (au format vcd) et vos fichiers MP3 enregistrés sur vos cartes mémoires sur votre télé. La télécommande fourni vous permet de faire un SlideShow entre vos différentes photos. Vous pouvez le brancher sur un vidéoprojecteur et faire ainsi des présentations sans avoir à transporter du matériel informatique lourd et encombrant. ▶Compatible avec les Smart Media, Compact Flash, Microdrive, MemoryStick, Secure Digital et Multimedia Card. ▶Connectique : 3 prises cinch (1video, 2 audio) ▶**Inclus** : adaptateur secteur, câble de connexion à la télé ou au vidéoprojecteur, télécommande. ▶**Dimensions** : 163x119x25mm(LxPxH) ▶**Poids** : 220g **Réf. PE660**

139,90 TTC € (917,68 F)

Baie HUB-USB prises frontales



Cette baie USB se positionne dans une baie 3,5" de votre ordinateur. Vous pourrez ainsi connecter 4 périphériques USB supplémentaires par l'avant de votre PC.**Réf.PE8257**

Version USB 2 **Réf.PE8264**
Prix : 39,90€TTC /261,73F

à partir de **14,90 TTC €** (97,74 F)

Bon de Commande à retourner à PEARL Diffusion à l'adresse ci-dessous

Quantité	Désignation	Prix Unitaire	Prix Total

MISC6

TOTAL :
Frais de port : 7 €
Si contre remboursement : +6,86 €
Option 24H : +6,86 €
TOTAL A PAYER :

6, rue de la Scheer - B.P. 121
ZI Nord - 67603 SELESTAT
Tél : 03 88 58 02 02
Fax : 03 88 58 02 07

Nom & Prénom _____

Adresse _____

Code postal / Ville _____

Signature

Mode de paiement :
__Chèque (Uniquement par courrier) __Mandat __Contre remboursement
__Carte bancaire : N° : ____ / ____ / ____ / ____
Date d'expiration : ____ / ____

Commandes et devis administratifs uniquement par courrier - Pas de livraison dans les DOM TOM et Hors Europe

Pocket cam 3M

Cet appareil numérique est équipé d'un capteur CMOS de **2Millions de pixels** interpolable à 3 millions. Il est de par sa taille et son ergonomie le partenaire idéal de vos «reportages photos». **Caractéristiques :** ▶Résolution 1600x1200 et jusqu'à **2048x1536** par interpolation ▶écran LCD de 1.5" ▶Mémoire : 16Mo intégré et extensible par cartes Compactflash ▶**Flash intégré** ▶Mise en veille automatique ▶Connexion USB ▶Alimentation par piles (2xAA) ▶Retardateur : 10 secondes ▶Taille : 98x57x45mm ▶Poids 100g ▶Inclus câble de liaison **Réf.PE3328**

159,90 € TTC (1048,88 F)

Souris optique 4D à fréquence

Cette souris intègre deux technologies de pointe. Un capteur optique qui évite l'encrassement de la souris et la technologie sans fil qui libère l'utilisateur de la contrainte des câbles. Le récepteur des informations provenant de la souris sert également de base de recharge pour la souris. ▶Connectique USB ou PS/2 ▶**Portée :** jusqu'à 3 mètres ▶**Inclus :** batteries (2 piles Ni-MH), adaptateur secteur, adaptateur USB/PS2. **Réf. PE9308**

34,90 € TTC (228,93 F)

Système audio 5.1 Q-Sonic

Profitez pleinement de vos DVD grâce au son surround ! Très simple à installer, ce système complet se compose d'un caisson de basse, de 4 satellites ainsi que d'une voie centrale. Le son venant de toutes les directions, vous aurez ainsi l'impression d'être au milieu de l'action ! Vous pourrez également brancher le système à votre PC muni d'une carte son 5.1 en utilisant 3 câbles jack – cinch (référence PE8214). **Caractéristiques techniques :** ▶Subwoofer actif amplifié. Puissance : 35 W RMS ▶Réglages du volume (subwoofer,centre, master, arrière) ▶Plage de fréquences : Subwoofer : 30 Hz – 200 Hz, satellites : 180 Hz – 20 kHz ▶**Dimensions (lxHxP) :** Subwoofer : 180x243x320mm Satellites : 116x318x116mm Voie centrale : 264x80x116mm ▶Connectique : 3 x audio in stéréo (cinch) ▶Inclus : câbles pour satellites : 2 x 450 cm pour les arrières, 2 x 170 cm (avant), 1 x 170 cm (voie centrale).) **Réf. PE2954**

89,90 € TTC (589,71 F)

Boîtier 3,5" Cristal USB

Connectez un périphérique IDE au format 3,5" (disque dur, lecteur ZIP, lecteur LS120,...) à chaud grâce à ce boîtier extra plat. Nécessite Windows 98, Millenium, 2000, MAC OS 8.6 ou 9.X

Version USB **Réf. PE8987**

79,90 € TTC (524,11 F)

Lecteur de cartes multifonction

Ce boîtier a 2 avantages : il vous permet de lire 6 types de cartes mémoires (Compact Flash, Smart Media, Multimédia Card, Memorystick, Microdrive et Secure Digital), vous apporte des prises en façade (2 prises USB, 1 prise Firewire, 1 entrée audio, 1 sortie audio). Il se branche dans une baie 5"1/4 et est alimenté par le PC. **Réf. PE7108**

livré sans cartes

49,90 € TTC (327,32 F)

Hub Firewire + USB

Avec un seul boîtier, vous aurez 4 ports USB et 3 ports Firewire supplémentaires. Un système de LED vous permet de contrôler le statut de l'appareil en un coup d'œil. Fourni avec un câble USB-Firewire en Y de 1.8m. **Réf. PI59**

89,95 € TTC (590,03F)

Kit Spécial portable

Ce kit sera le partenaire idéal de votre portable lors de vos déplacements. Il se compose d'un **HUB USB extra fin 4 ports, 1 pavé numérique USB, 1 mini souris** optique USB ainsi qu'un câble de modem avec enrouleur, le tout livré dans une sacoche de transport. **Réf. PE8757**

59,90 € TTC (392,92 F)

Adaptateur IDE USB externe

Grâce à ce kit composé d'un adaptateur IDE / USB et d'une alimentation, vous pourrez brancher votre disque dur ou n'importe quel périphérique IDE sur un port USB à chaud et ce, sans boîtier externe ! Vous pouvez ainsi disposer d'un moyen de sauvegarde très pratique et peu encombrant. **Réf. PE8191**

69,90 € TTC (458,51 F)

Adaptateur IDE USB 2.0 externe

Réf.PE8193 Prix : 79,90€TTC/524,11F

Kit graveur YAMAHA CRW F1 SX Externe SCSI 44x 24x 44x

Ce graveur externe ultra rapide de Yamaha bénéficie de caractéristiques technniques exceptionnelles et d'une bonne ergonomie. Non seulement il atteint des vitesses de gravures impressionnantes mais en plus il intégre une nouvelle fonctionnalité de personnalisation des CD : la technologie T@2 vous permet d'intégrer des images sur la surface libre de la face gravée ▶**Buffer 8Mo** ▶technologie Safeburn ▶type des CD lus : CD-DA, CD-ROM, CD-ROM XA, CD-I, CD-Digital, CD-Bridge, CD-Extra et Vidéo CD ▶fonction audio master : **améliore la qualité sonore du CD d'origine** ▶Boîtier muni d'une prise mi ni SubD50 (Interface SCSI - 3 Ultra SCSI) ▶Garantie 2 ans **Réf. PE1525**

Garantie sur site

Inclus : Nero 5.5 +1 CD-RW + 1 CD-R

A SAISIR

179,90 € TTC (1180,07F)

LINUX Mandrake 9

Cette distribution basée sur le kernel 2.4.19 comprend la glibc 2.2.5, GCC 3.2, KDE 3.0.3, Gnome 2.0.1, OpenOffice 1.0.1, Mozilla 1.1, Gimp 1.3.2, XFree 4.2.1 et bien d'autres logiciels. Le support d'un grand nombre de cartes graphiques 3D et de l'USB en font un outil puissant et complet. Mandrake 9.0 est optimisée pour les processeurs Pentium (tm) et supérieurs.

Linux Mandrake Powerpack Ce pack très complet inclus les meilleures applications Open Source et commerciales disponibles. **Réf. LI809**

Prix : 69,90€/458,51 F

Débutant

Linux Mandrake Pro suite La solution Linux pour l'entreprise. Offrant un support technique étendu. **Réf.LI810**

Prix : 189,90€/1245,66F

à partir de **69,90 € TTC** (458,51 F)

MYTH II Soulblighter

Vous voilà commandant des troupes de magiciens, combattants et autre nains. Votre mission est de défendre le peuple Madrigal et Westens des maléfiques Soulblighter. Vous menerez vos troupes au combat en contrôlant chacun de leurs déplacements grâce à une interface 3D pilotée entièrement à la souris. Une carte d'accélération 3D (3dfx) est fortement conseillée. **Réf. LI15**

VALABLE JUSQU'A EPUISEMENT DE STOCK

19,90 € TTC (130,54 F)

DEBIAN

GNU/LINUX 2.1 (PC Intel)

Il s'agit d'une distribution Linux 100% libre. Elle se compose de 4 CDROMs (2 CD binaires + 2 CD sources) soit plus de 2200 packages. La mise à jour vers de prochaines versions se fait en tout simplicité et gratuitement via le serveur FTP officiel Debian **Réf. LI12**

5,95 € TTC (39 F)

LINUX Nirvana

Nous avons testé, trié et sélectionné pour vous les meilleurs logiciels LINUX disponibles sur Internet (plus de 600 Mo). Beaucoup sont livrés avec leurs sources et couvrent des domaines aussi divers que la P.A.O., le dessin, la C.A.O., les éditeurs, les langages, les utilitaires, etc...avec de nombreuses documentations. **Réf. CS25**

4,42 € TTC (29 F)

GNU Collection

What is GNU? "Gnu is Not Unix"! Nous avons récupéré pour vous le maximum de logiciels sous licence GPL comme Emacs, Tex, GCC, Gzip sont donc gratuits et LIVRÉS avec leurs SOURCES. Ces applications sont disponibles pour de nombreux systèmes d'exploitation, vous trouverez forcément votre bonheur. **Réf. CS24**

4,42 € TTC (29 F)

RedHat 8.0

Un système d'exploitation complet mis à jour avec les technologies les plus récentes. Ce système possède le nouveau bureau intuitif Bluecurve et des centaines d'applications (Evolution 1.0.8, Gimp 1.2.3, GNOME 2.0, KDE 3.0.3, Kernel 2.4.18, etc) **Réf. LI32**

79,95 € TTC (524,44 F)

PEARL
0,12 €/mn
N°Indigo 0 820 822 823
Tél. 03 88 58 02 02
Fax 03 88 58 02 07

EXPOSITION-CONFERENCES

NET SEC

SÉCURITÉ SAUVEGARDE STOCKAGE

Toutes les solutions pour l'entreprise

- 30 conférences
- 80 exposants
- 5.000 visiteurs

1•2•3 AVRIL 2003

Paris Expo - Porte de Versailles

INFOPROMOTIONS
LES SALONS SOLUTIONS

97, rue du Cherche-Midi - 75006 PARIS - FRANCE
Tél. : 33 (0)1 44 39 85 00 - Fax : 33 (0)1 45 44 30 40
seti@infopromotions.fr - www.groupesolutions.com

seti Semaine Européenne des Technologies de l'Information

hall4