Le Grand Livre de

http ://www.securiteinfo.com
18 février 2002

Table des matières

Chapitre 1

Les attaques

1.1 Le hacking

1.1.1 Qu'est-ce que c'est ?

Le hacking est un ensemble de techniques informatiques, visant à attaquer un réseau, un site, etc. Ces attaques sont diverses. On y retrouve :

- L'envoi de "bombes" logicielles.
- L'envoi et la recherche de chevaux de Troie.
- La recherche de trous de sécurité. Le détournement d'identité.
- La surcharge provoquée d'un système d'information (Flooding de Yahoo, eBay...).
- Changement des droits utilisateur d'un ordinateur. La provocation d'erreurs non gérées.
- Etc.

Les attaques peuvent être locales (sur le même ordinateur, voir sur le même réseau) ou distantes (sur internet, par télécommunication).

1.1.2 Le but du hacking

Le but du hacking est divers. Selon les individus (les "hackers"), on y retrouve : Vérification de la sécurisation d'un système. Vol d'informations (fiches de paye...). Terrorisme. Espionnage "classique" ou industriel. Chantage. Manisfestation politique. Par simple "jeu", par défi. Pour apprendre. Etc.

1.1.3 Le hacking légal

Le site anglophone Cyberarmy comporte une idée originale. Cette armée virtuelle est composée de hackers de tout niveau. Lorsque vous vous inscrivez vous êtes un Trooper (soldat de 2e classe). Le site propose plusieurs niveaux de protection qu'il faut hacker, et ce, en toute légalité. Au fur et à mesure que vous passez les niveaux de protection de cyberarmy, vous montez en grade. Bien sûr, plus vous montez en grade, plus le niveau est difficile. Pour information le webmaster de securiteinfo.com est Colonel de la cyberarmy (Kernel scrap). L'inscription se fait sur Zebulun

1.2 Les types d'attaque

1.2.1 Introduction

Les hackers utilisent plusieurs techniques d'attaques. Ces attaques peuvent être regroupées en trois familles différentes :

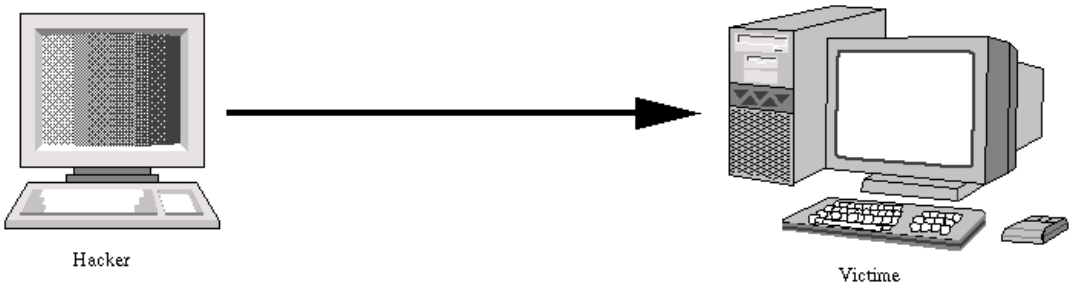
- Les attaques directes.
- Les attaques indirectes par rebond.

– Les attaques indirectes par réponses.

Nous allons voir en détail ces trois familles.

1.2.2 Les attaques directes

C'est la plus simple des attaques. Le hacker attaque directement sa victime à partir de son ordinateur. La plupart des "script kiddies" utilisent cette technique. En effet, les programmes de hack qu'ils utilisent ne sont que faiblement paramétrable, et un grand nombre de ces logiciels envoient directement les packets à la victime.



Attaque directe

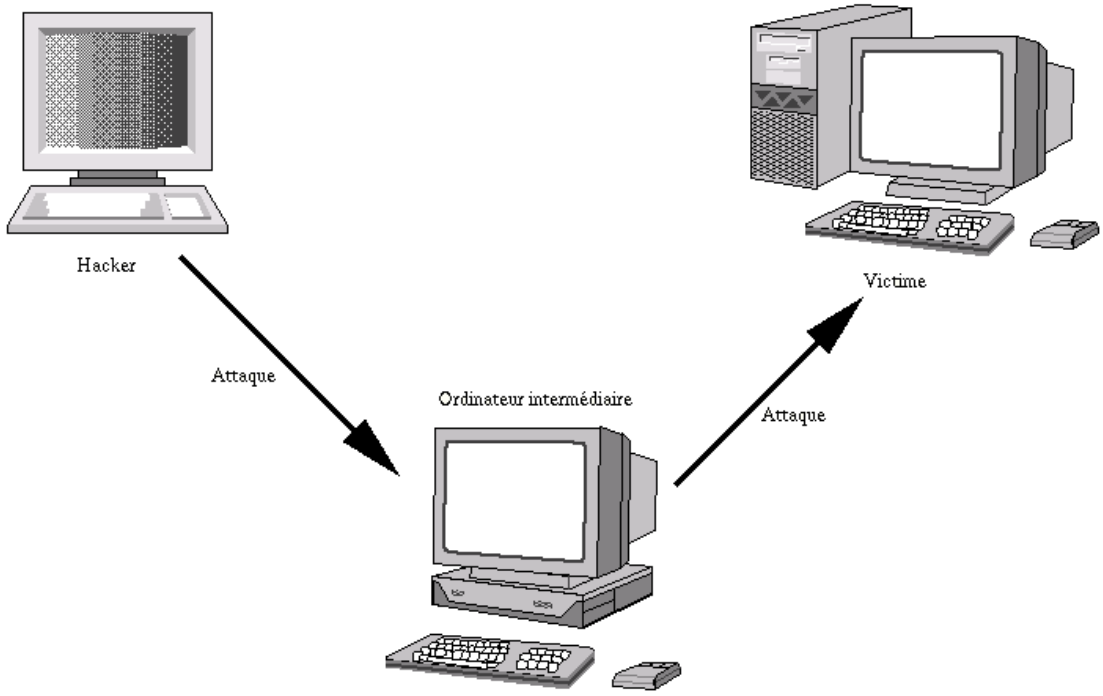
Si vous vous faites attaqués de la sorte, il y a de grandes chances pour que vous puissiez remonter à l'origine de l'attaque, identifiant par la même occasion l'identité de l'attaquant.

1.2.3 Les attaques indirectes par rebond

Cette attaque est très prisée des hackers. En effet, le rebond a deux avantages :

– Masquer l'identite (l'adresse IP) du hacker.
– Eventuellement, utiliser les ressources de l'ordinateur intermédiaire car il est plus puissant (CPU, bande passante...) pour attaquer.

Le principe en lui même, est simple : Les packets d'attaque sont envoyés à l'ordinateur intermédiaire, qui répercute l'attaque vers la victime. D'où le terme de rebond.

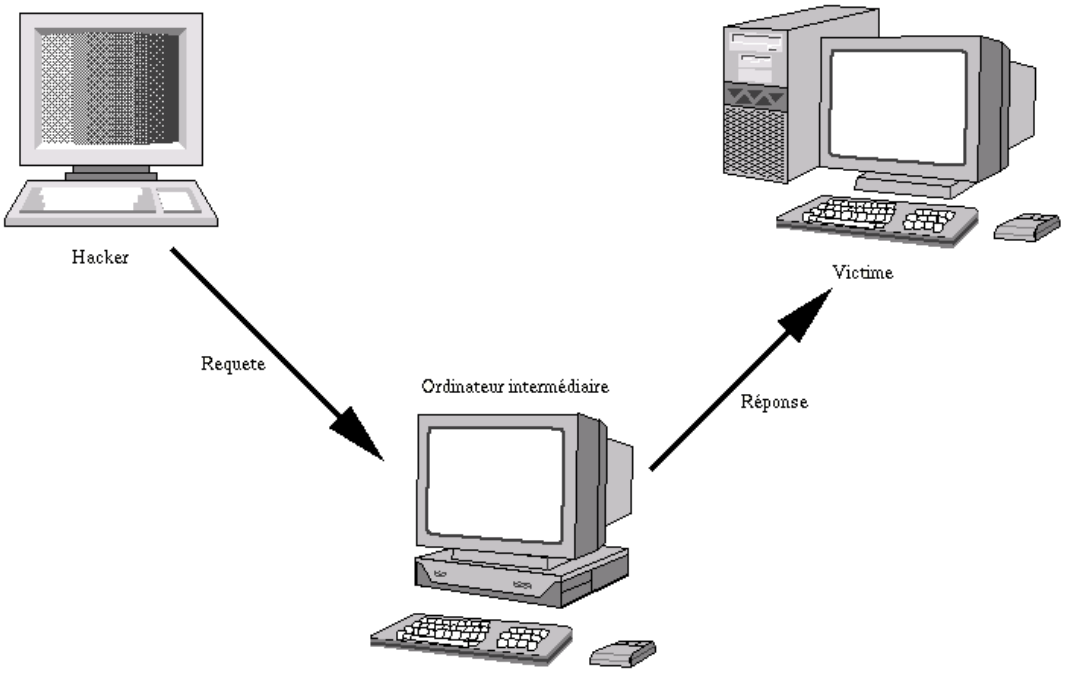


Attaque indirecte par rebond

L'attaque FTP Bounce fait partie de cette famille d'attaque. Si vous êtes victime de ce genre d'attaque, il n'est pas facile de remonter à la source. Au plus simple, vous remontrez à l'ordinateur intermédaire.

1.2.4 Les attaques indirectes par réponse

Cette attaque est un dérivé de l'attaque par rebond. Elle offre les même avantages, du point de vue du hacker. Mais au lieu d'envoyer une attaque à l'ordinateur intermédiaire pour qu'il la répercute, l'attaquant va lui envoyer une requête. Et c'est cette réponse à la requête qui va être envoyée à l'ordinateur victime.



Attaque indirecte par réponse

Là aussi, il n'est pas aisé de remonter à la source...

1.2.5 Conclusion

Lorsque vous vous faites attaquer, cela peut se faire en direct ou via un ou plusieurs ordinateurs intermédiaires. Le fait de comprendre l'attaque va vous permettre de savoir comment remonter au hacker.

1.3 L'attaque +++ATHZero

1.3.1 Qu'est-ce que c'est ?

L'attaque +++ATH0 vise certains modems compatibles Hayes. Lorsque ce type de modem reçoit la commande +++ATH0, il risque de se déconnecter. En effet, cette commande permet de positionner le modem en commande manuelle. En pratique, cela se traduit par l'envoi d'un "Ping" contenant la chaîne de caractères "+++ATH0".

1.3.2 Conséquences

- Déconnexion du modem

1.3.3 Comment s'en protéger ?

- Pour les systèmes Win32, vous devez rechercher dans la base de registres la clé : HKEY_LOCAL_MACHINE\Sy et créer la chaîne "UserInit", ayant pour valeur "s2=255".
- Pour tous les autres systèmes, ajouter la commande "S2=255" dans la chaine d'initialisation du modem. Cela donne "ATZ ATS2=255&W". Cette commande ajoutée permet de désactiver la commande de mode manuel.

Si des problèmes persistent, jetez un oeil dans le manuel de votre modem.

1.4 L'attaque Boink

1.4.1 Qu'est-ce que c'est ?

L'attaque Boink vise les systèmes Win32. Elle est semblable à l'attaque Bonk. Elle consiste à envoyer des packets UDP corrompus sur tous les ports ouverts. L'ordinateur victime ne gère pas ces paquets et provoque un plantage.

1.4.2 Conséquences

Blocage système Crash système

1.4.3 Comment s'en protéger ?

Mettre à jour l'OS. Utilisation d'un firewall pour refuser les packets UDP corrompus.

1.5 L'attaque Cisco ® 7161

1.5.1 Qu'est-ce que c'est ?

Cela consiste à se connecter au port 7161 d'un routeur Cisco ® et d'envoyer un retour chariot. Le routeur peut alors planter.

1.5.2 Conséquences

Plantage du routeur Cisco ®.

1.5.3 Comment s'en protéger ?

Contactez Cisco ® pour obtenir une solution.

1.6 L'attaque Click - WinNewk

1.6.1 Qu'est-ce que c'est ?

Cette attaque vise tous les systèmes. Elle consiste à envoyer un message d'erreur ICMP (typiquement, ICMP inaccessible) à l'ordinateur cible ou au serveur auquel la victime est connectée. La victime risque alors d'être déconnectée du réseau ou du serveur.

1.6.2 Conséquences

Déconnexion

1.6.3 Comment s'en protéger ?

Configurer les firewall/routeurs pour gérer ces messages

1.7 Le Mail Bombing

1.7.1 Qu'est-ce que c'est ?

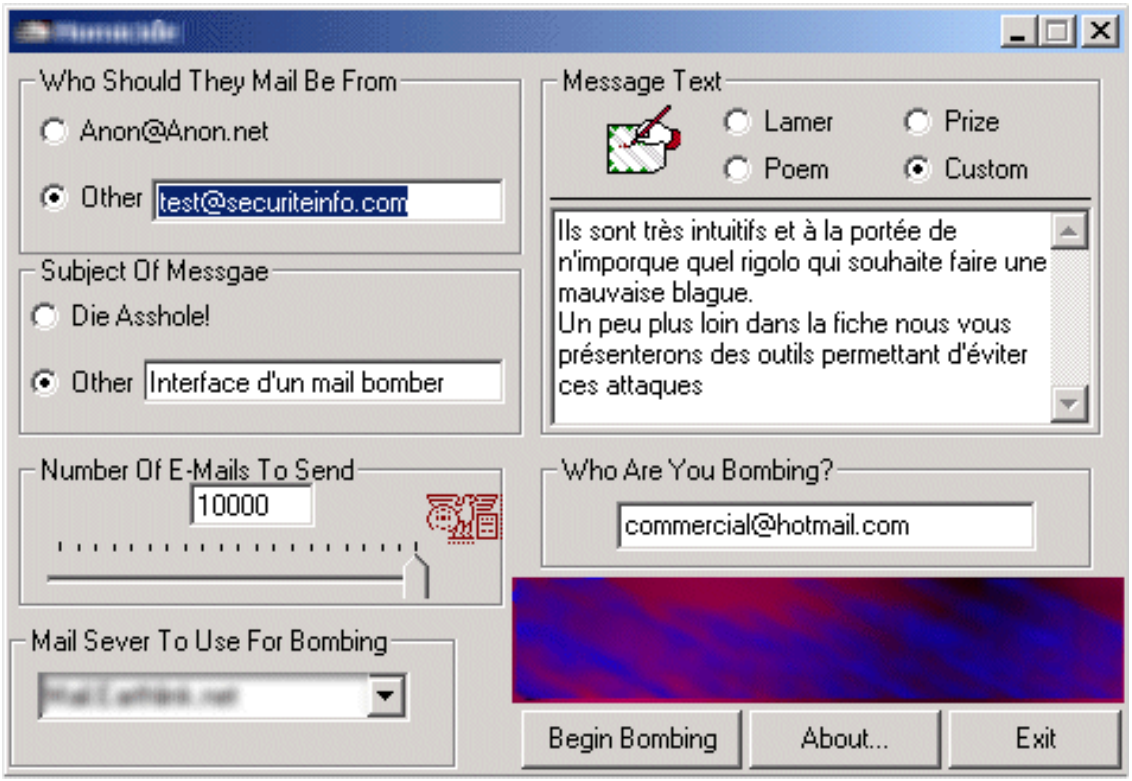
Le Mail Bonbing constite à envoyer un nombre faramineux d'emails (plusieurs milliers par exemple) à un ou des destinataires. L'objectif étant de :

- saturer le serveur de mails
- saturer la bande passante du serveur et du ou des destinataires,
- rendre impossible aux destinataires de continuer à utiliser l'adresse électronique.

1.7.2 L'attaque

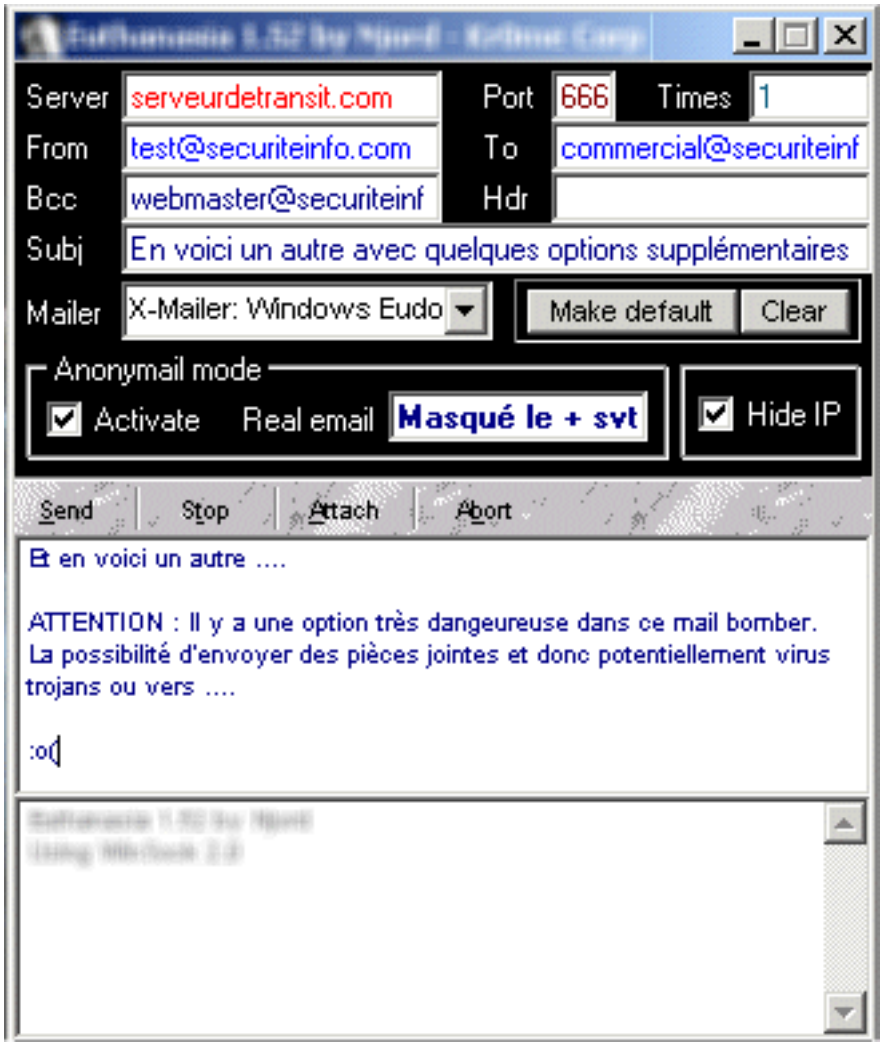
Il est nécessaire pour l'auteur de l'attaque de se procurer un logiciel permettant de réaliser le mail bombing. Voici comment cela fonctionne

Exemple 1



- L'attaquant ici choisi différents options :
- l'adresse qu'il veut faire apparaître en tant qu'émetteur du message ;
- le sujet du message,
- le nombre de messages à envoyer, le serveur de mail à partir duquel les messages seront émis, (bien souvent si les administrateurs de serveurs mails ne se protègent pas assez, des serveurs "innocents" servent de relais sans le savoir, et le danger pour leurs propriétaires est de se retrouver "black listés" c'est à dire voir son fournisseur d'accès internet lui couper sa connection),
- le corps du message,
- l'adresse email de la victime.

Exemple 2



Cet outil est aussi intuitif que le précédent. Cependant il semble nettement plus dangereux. En effet, la possibilité d'attacher une pièce jointe est une sérieuse menace, puisqu'elle permet à l'expéditeur d'insérer virus et troyens dans les messages. Une fois de plus, rappelons qu'il faut impérativement éviter d'ouvrir une pièce jointe ayant pour extension .com, .bat, .pif ou .exe...

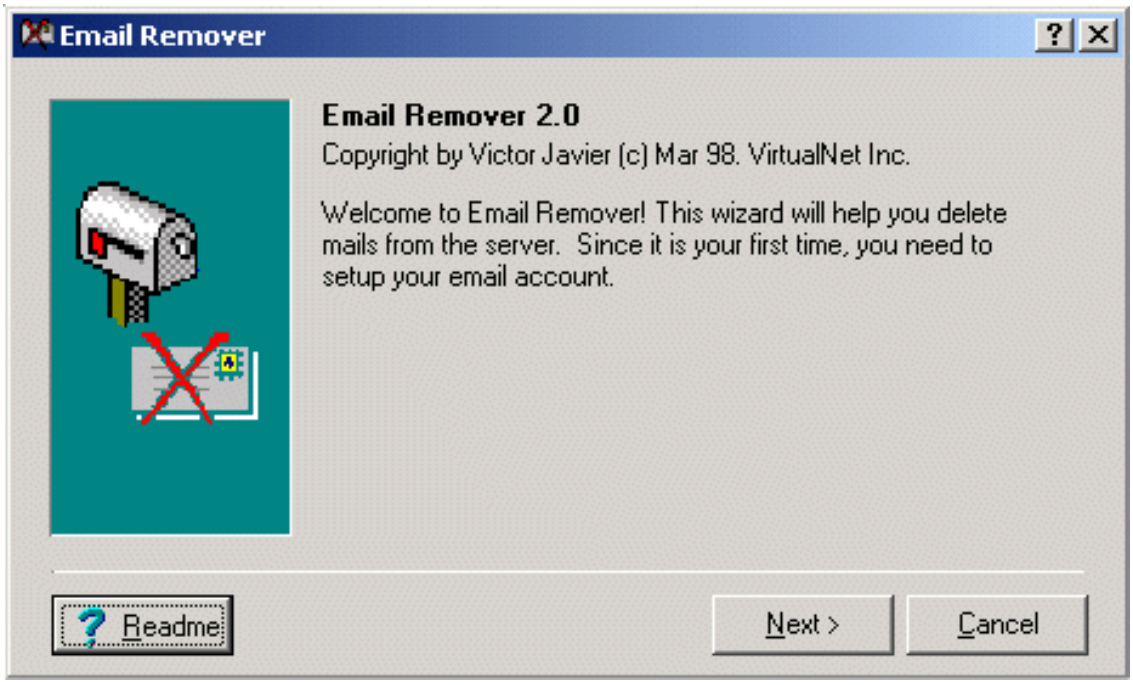
1.7.3 Comment réagir ?

Evitez cette attaque

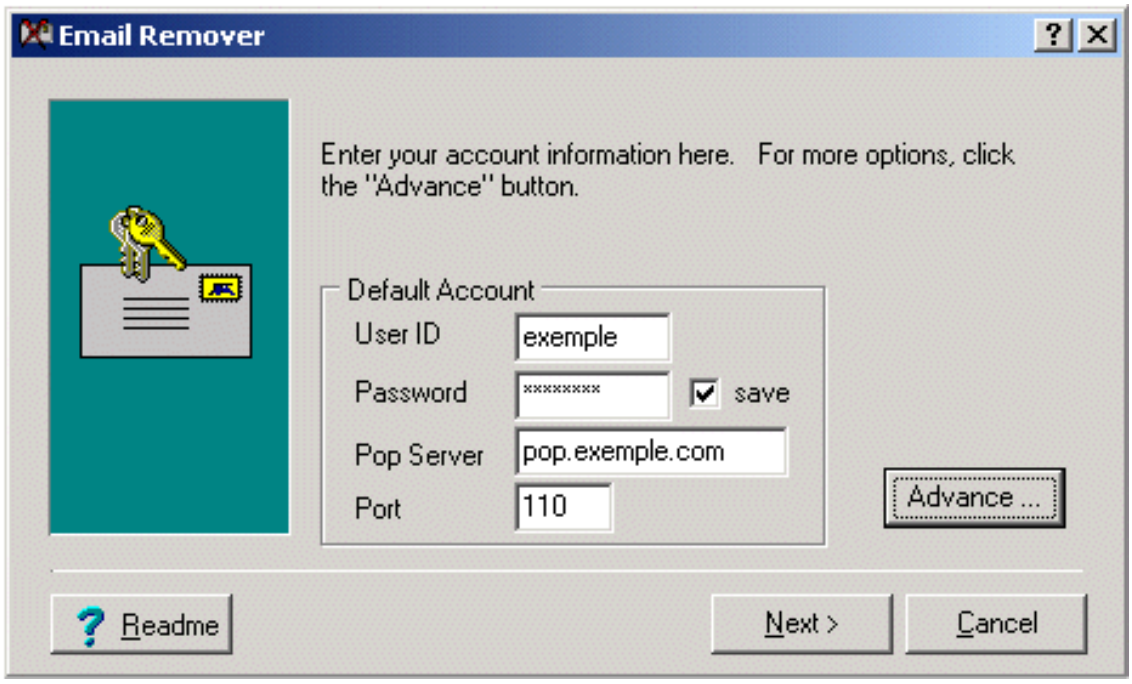
Avant de prendre le risque d'avoir une adresse électronique inutilisable mieux vaut prendre ses précautions :

- Si vous avez une adresse personnelle à laquelle vous tenez, ne la communiquez qu'aux personnes dignes de confiance,
- Créez vous un second compte de messagerie, pour tout ce qui est mailing list par exemple et groupe des discussion, ainsi, vous ne craingez pas de perdre d'infomations vitales. Si ce compte est attaqué vous pourrez sans dificulté reprendre une autre adresse et vous réabonner.
- Utilisez eremove pour éviter les mail bombers.

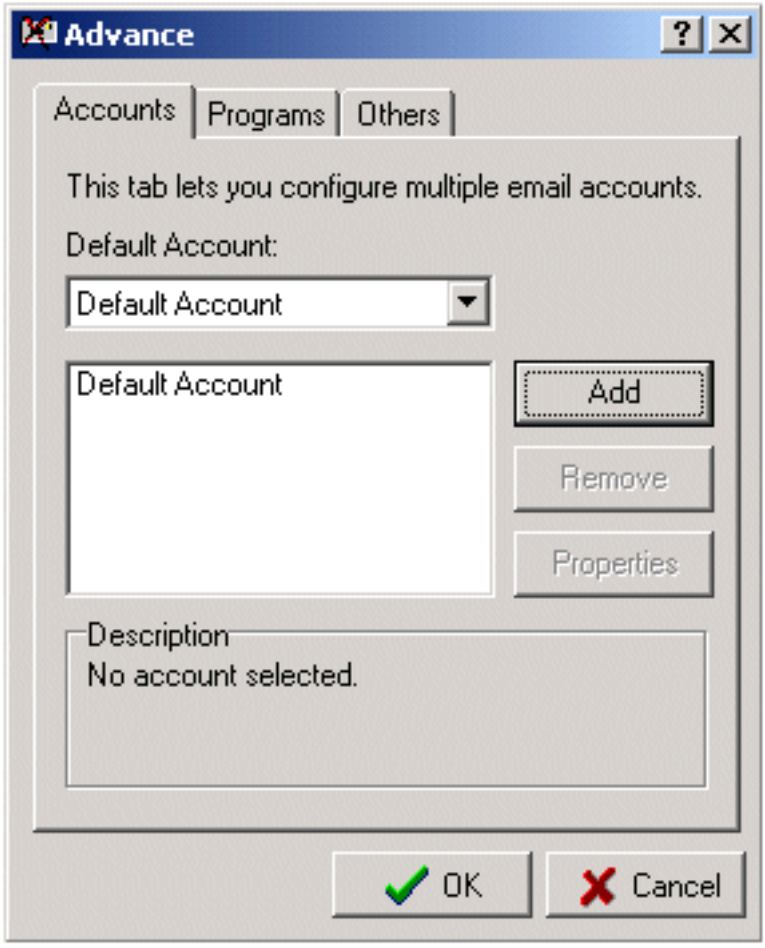
Lorsque vous lancez l'installation du programme vous retrouvez cet écran :

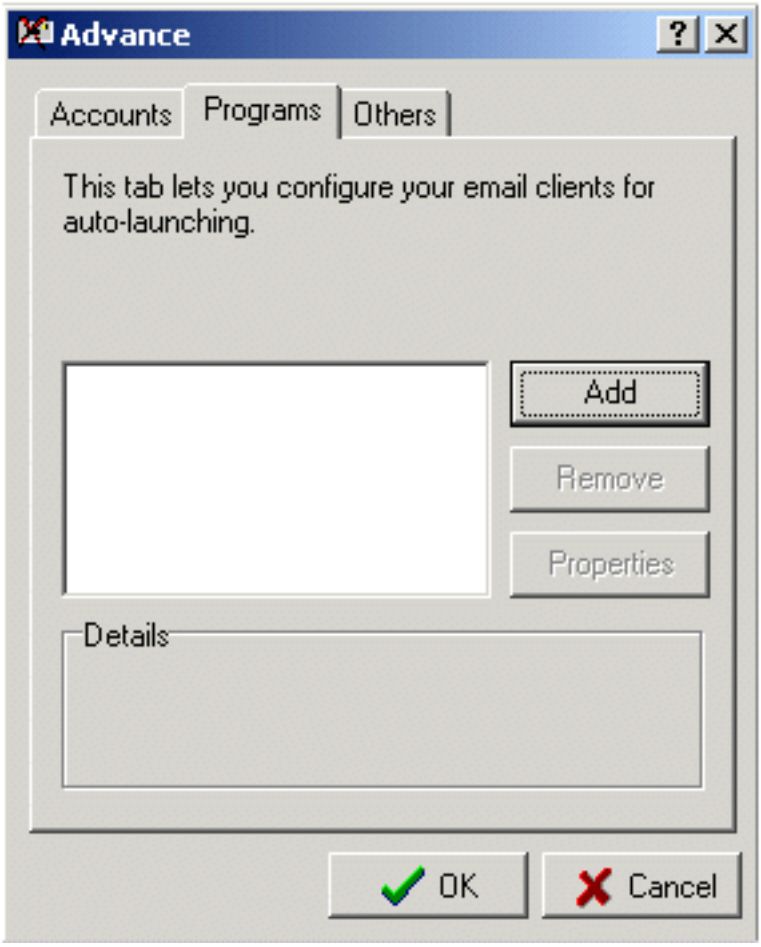


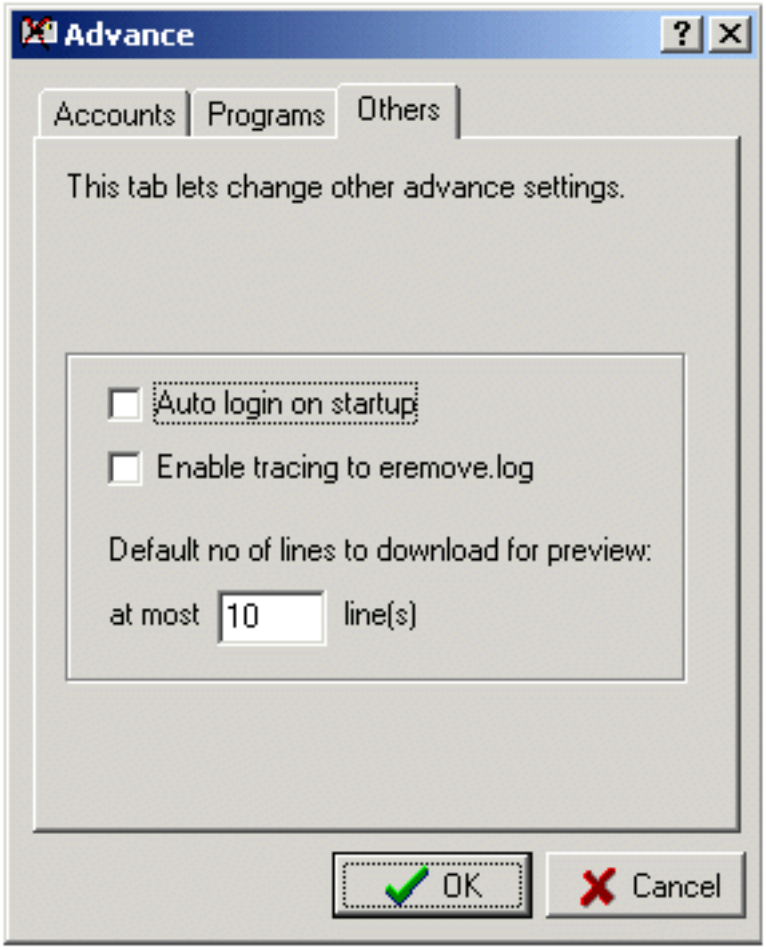
Vous allez maintenant pouvoir commencer la configuration en tapant sur next.



Ici, vous devez rentrer les identifiants de votre messagerie, votre mot de passe, le serveur de votre FAI ainsi que le port utilisé (par défaut c'est généralement le 110). Pour les personnes disposant de plusieurs comptes de messagerie différents, il est nécessaire de passer par le mode avancé de configuration. Cliquer sur "Advance" vous amène aux écrans suivants :


Advance
Accounts
Programs
Others
This tab lets you configure multiple email accounts.
Default Account:
Default Account
Default Account
Add
Remove
Properties
Description
No account selected.
OK
Cancel

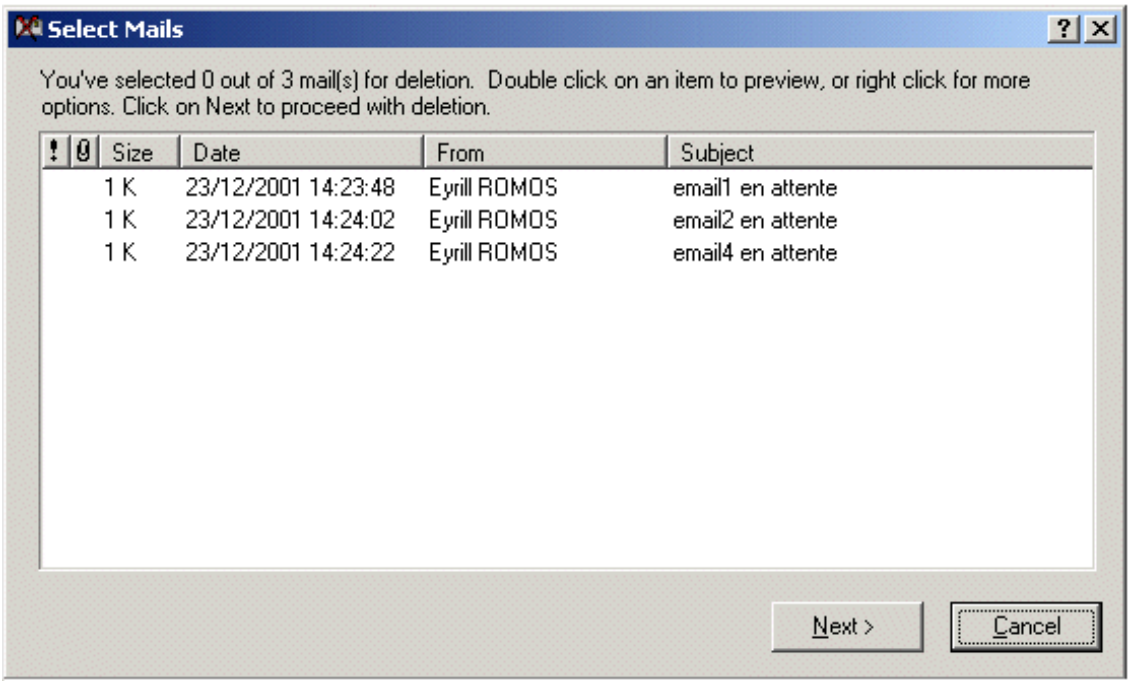

Advance
Accounts
Programs
Others
This tab lets you configure your email clients for auto-launching.
Add
Remove
Properties
Details
OK
Cancel



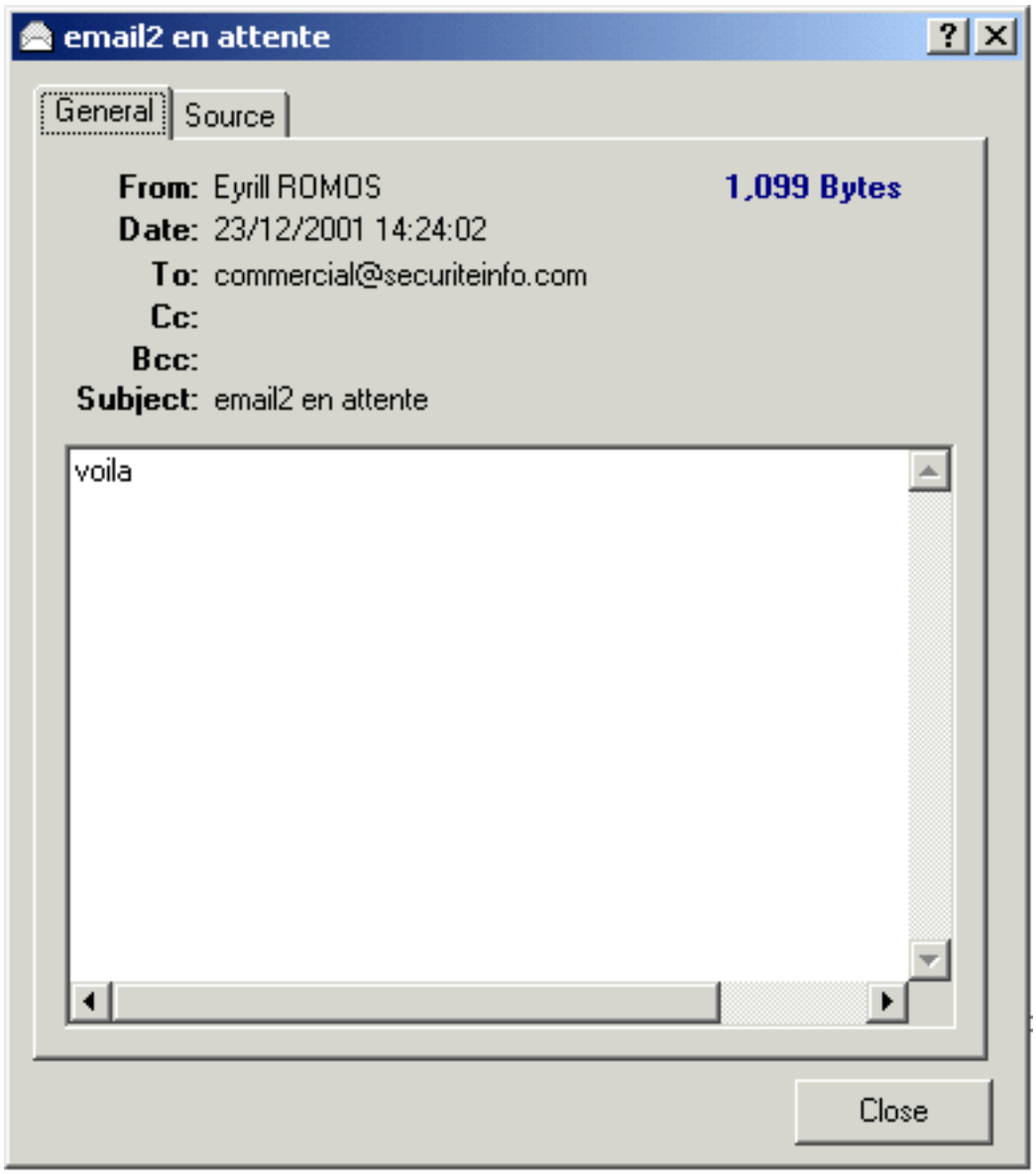
L'onglet Account permet de donner les indications sur tous les comptes de messagerie que vous souhaitez protéger. A chaque fois que vous cliquez sur Add vous trouverez un écran similaire à celui que vous avez vu pour votre compte principal. Vous pouvez entrer autant de comptes que vous le désirez.

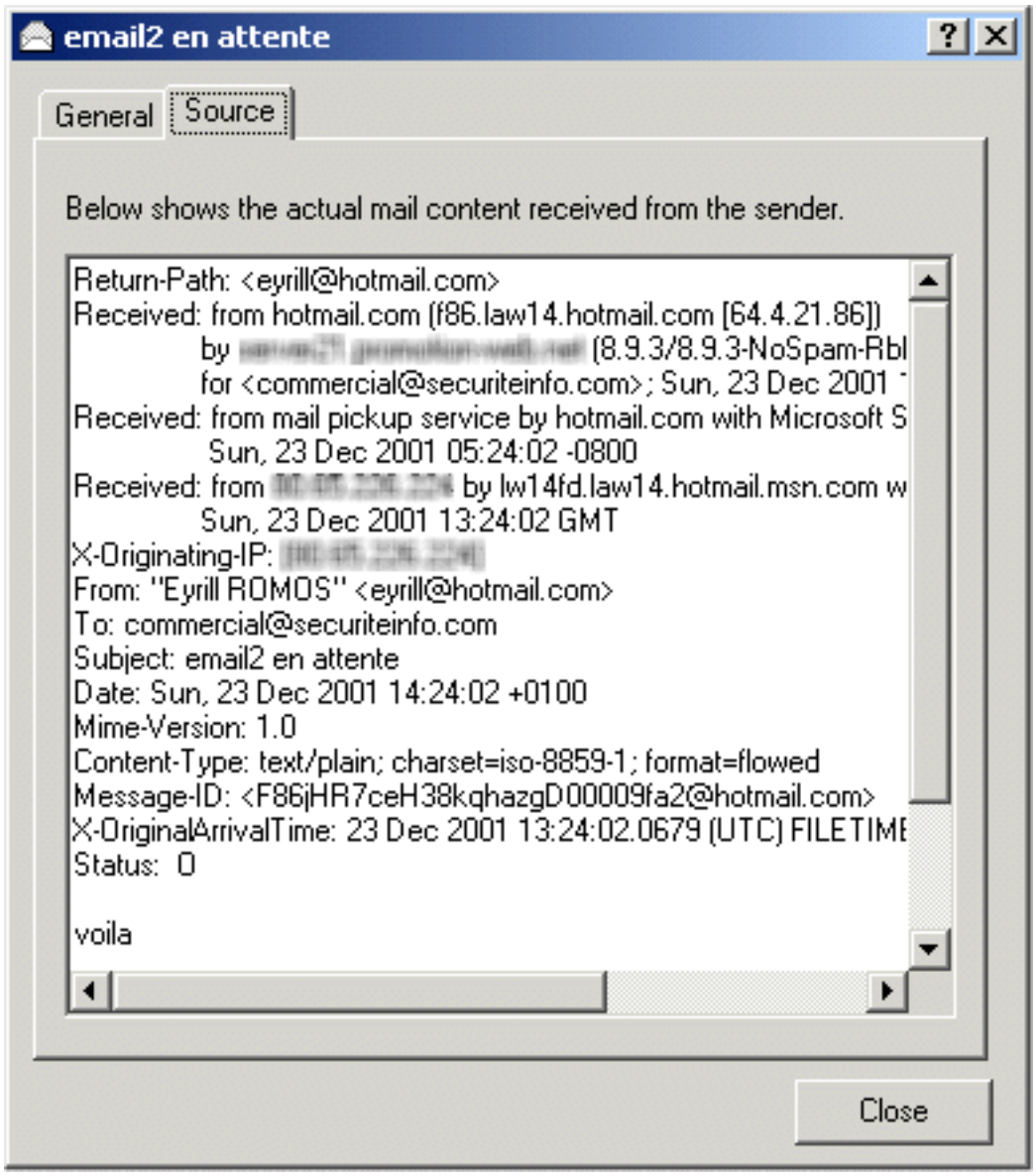
L'onglet Programs vous permet de déterminer quel est le type de Boîte aux Lettres que vous utilisez (Eudora, Outlook, etc ...).

L'onglet Others vous permet de déterminer si le programme se connecte directement à votre boite aux lettres à son lancement. Vous pouvez aussi spécifier un fichier de logs



Lorsque vous lancez eremove, le programme vous montre le nombre de messages, ainsi que la taille de chacun et l'émetteur. Il vous suffit ensuite de sélectionner le ou les messages que vous ne souhaitez pas recevoir et ils seront directement détruits sur le serveur de messagerie.


email2 en attente
General
Source
From: Eyrill ROMOS
1,099 Bytes
Date: 23/12/2001 14:24:02
To: commercial@securiteinfo.com
Cc:
Bcc:
Subject: email2 en attente
voila
Close



Vous pouvez au préalable vérifier le contenu du message en faisant un click gauche avec la souris sur le message. Cela vous donnera des éléments sur le corps du message et sur l'expéditeur.

Vous avez été attaqué

Si vous avez été victime d'un mail bombing, il est parfois possible de remonter jusqu'à l'émetteur. En effet, il existe des informations dans chaque message qui donnent des informations sur leur auteur. Voici un exemple de propriétés de message :

- Return-Path : <eyrill@hotmail.com> Ici vous trouvez l'email de l'émetteur
- Received : from hotmail.com (f88.law14.hotmail.com [64.4.21.88]) by servertoto.pourexemplejenevaispsvousmons (8.9.3/8.9.3-NoSpam-Rbl-ORBS) with ESMTP id PAA19370 Ici vous trouvez le serveur par lequel l'attaquant à envoyé les messages. Si la personne débute il se peut que ce serveur soit réel et il vous faut vous rapprocher de son propriétaire pour vous plaindre.
- For commercial@securiteinfo.com ; Sat, 22 Dec 2001 15 :45 :34 +0100 Ici vous trouvez normalement votre adresse de messagerie ainsi que des indications horaires
- Received : from mail pickup service by hotmail.com with Microsoft SMTPSVC ; Sat, 22 Dec

2001 06 :33 :00 -0800
- Received : from XXX.XXX.XXX.XXX by lw14fd.law14.hotmail.msn.com with HTTP ; Sat, 22 Dec 2001 14 :33 :00 GMT
- X-Originating-IP : [XXX.XXX.XXX.XXX] Ici vous trouvez les indications sur l'IP d'où sont partis les messages. Attention, il est possible mais assez rare que l'adresse IP soit modifiée
- From : "Eyrill ROMOS" <eyrill@hotmail.com> De nouveau l'émetteur
- To : &ltcommercial@securiteinfo.com> De nouveau le destinataire
- Subject : =?iso-8859-1?B?UHJvcHJp6XTpcyBkJ3VuIG1lc3NhZ2Ug?= Le sujet du message encodé
- Date : Sat, 22 Dec 2001 15 :33 :00 +0100 Date et heure
- Mime-Version : 1.0 Version Mime utilisée pour l'encodage du message
- Content-Type : text/plain ; charset=iso-8859-1 ; format=flowed Type de contenu
- Message-ID : <F88lyg35sQtgTIFprhM000098e1@hotmail.com> Identifiant interne du message
- X-OriginalArrivalTime : 22 Dec 2001 14 :33 :00.0561 (UTC) FILETIME=[8E825010 :01C18AF5] Heure et date d'arrivée du message

Si vous retrouvez des informations comme l'adresse email ou le serveur qui ont permis l'arrivée des messages, il est important de se plaindre auprès du fournisseur d'accès. En effet, dans la plupart des cas les fournisseurs d'accès n'apprécient pas ce type de procédés via leurs serveurs et prennent toutes les mesures nécessaires pour empêcher les auteurs de recommencer.

1.7.4 Conclusion

Le mail bombing n'est, à priori, pas illégal. Il n'existe pas de limite légale déterminant le nombre maximum de messages à envoyer à un internaute. Cependant, les fournisseurs d'accès à Internet n'apprécient pas ce type de procédés. En effet, cela leur cause des soucis de bande passante et la saturation de leurs serveurs de messagerie. En conséquence, n'hésitez surtout pas à les solliciter si vous êtes victime d'une telle attaque. Ils réagissent généralement rapidement pour éviter que leurs abonnés recommencent. Par ailleurs, prendre le temps d'intaller eremove est indispensable si l'on désire éviter tout soucis et ne pas se retrouver contraint à changer d'adresse électronique. Une fois installé vous pouvez en toute quiétude ne plus craindre les attaques par mail bombing :o) ! ! !

1.8 L'attaque Out Of Band (OOB)

1.8.1 Qu'est-ce que c'est ?

L'attaque OOB est plus connue sous le nom de "Nuke". Elle est courante, car il y a de nombreux "utilitaires" qui permettent d'exploiter la faille. Les systèmes visés sont Win32. Le port visé est le 139 (Netbios Session Service port). Lorsqu'un packet est envoyé sur le port 139 avec le flag "Urgent", Win95/NT/3x attend des données qui doivent suivre le flag. S'il n'y a pas de données qui arrivent, le système ne sait pas gérer cette absence...

1.8.2 Conséquences

Ecran bleu de la mort Perte de la connexion internet Blocage système Crash système

1.8.3 Comment s'en protéger ?

Win95 : Mettre à jour Winsock avec la version 2. Win NT 3.51 : Installer le service pack 5 + le patch oob-fix. Win NT 4.0 : Installer le service pack 4. Utilisation d'un firewall et blocage du port 139. Utilisation d'un écouteur des ports du système.

1.9 L'attaque NT Inetinfo

1.9.1 Qu'est-ce que c'est ?

L'attaque NT Stop vise les systèmes WinNT 4.0. Elle consiste à se connecter au port 1031 (inetinfo) et à envoyer n'importe quoi. Quelques fois, il suffit juste de se connecter à ce port et de se déconnecter immédiatement. Le processus Inetinfo utilise alors énormément de ressources système et peut provoquer un plantage ou un reboot.

1.9.2 Conséquences

Blocage système Crash système Reboot système

1.9.3 Comment s'en protéger ?

Mettre à jour l'OS.

1.10 Ping of death

1.10.1 Qu'est-ce que c'est ?

Un ping a normalement une longueur maximale de 65535 ((2 exp 16) - 1) octets, incluant une entête de 20 octets. Un ping of death c'est un ping qui a une longueur de données supérieure à la taille maximale. Lors de sont envoi, le ping of death est fragmenté en packets plus petits. L'ordinateur victime qui reçoit ces packets doit alors les reconstruirent. Certains systèmes ne gèrent pas cette fragmentation, et se bloquent, ou crashent complètement. D'où le nom de cette attaque. Qui peut provoquer cette attaque? N'importe qui, du moment qu'il a un logiciel permettant de le faire.

1.10.2 Conséquences

Crash système. Blocage système.

1.10.3 Comment s'en protéger ?

Mettre à jour l'OS. Effectuer un test avant que quelqu'un d'autre le fasse à votre place. Si le système réagit correctement, il n'y a pas de problème.

1.11 L'attaque par requête HTTP incorrecte

1.11.1 Qu'est-ce que c'est ?

Cette attaque vise tous les systèmes, y compris IIS. Elle consiste à envoyer une requête HTTP déformée vers le site web cible. Le serveur peut alors se planter. Cette attaque à cette forme : GET / HTTP/1.0 hostname : aaaaaaaaaaaa... (256 octets) hostname : aaaaaaaaaaaa... (256 octets) ... 10,000 lignes ... hostname : aaaaaaaaaaaa... (256 octets)

1.11.2 Conséquences

Crash système, plantage du site web

1.11.3 Comment s'en protéger ?

Mettre à jour votre serveur web.

1.12 L'attaque Snork

1.12.1 Qu'est-ce que c'est ?

L'attaque Snork vise les systèmes WinNT. Elle consite à envoyer une trame UDP provenant du port 7 (Echo), 19 (Chargen) ou 135, et ayant pour destination le port 135 (Microsoft Location Service). Si les services sont lancés, cela a pour conséquence d'établir une communication de durée infinie, et génère des trames non nécessaires. Cela réduit considérablement la bande passante et la puissance CPU.

1.12.2 Conséquences

Ralentissement système Perte de bande passante

1.12.3 Comment s'en protéger ?

Configurer les routeurs et firewalls pour bloquer les packets UDP ayant une destination de port 135 et ayant un port source de 7,19 ou 135 et qui proviennent de l'extérieur de votre réseau. Microsoft a fourni un patch.

1.13 L'attaque SMTPd overflow

1.13.1 Qu'est-ce que c'est ?

Cela consiste à envoyer la commande "HELP" avec un argument trop long vers un serveur SMTP. Si le gestionnaire SMTP n'est pas patché pour prévenir de cette attaque, il plante.

1.13.2 Conséquences

Plantage du démon SMTP. Impossibilité d'envoyer ou recevoir un mail.

1.13.3 Comment s'en protéger ?

Pour les démons qui ne supportent pas une commande "HELP" trop longue, il existe certainement un patch. Mettre à jour votre démon SMTP.

1.14 L'ARP redirect

1.14.1 Qu'est-ce que c'est ?

L'attaque ARP redirect vise les réseaux locaux ethernet, qu'ils soient partitionnés ou non en sous-réseaux (switchés). C'est une technique de spoofing efficace bien que détectable dans les logs d'administration ; elle consiste à s'attribuer l'adresse IP de la machine cible, c'est-à-dire à faire correspondre son adresse IP à l'adresse MAC de la machine pirate dans les tables ARP des machines du réseau. Pour cela il suffit en fait d'envoyer régulièrement des paquets ARP_reply en broadcast, contenant l'adresse IP cible et la fausse adresse MAC. Cela a pour effet de modifier les tables dynamiques de toutes les machines du réseau. Celles-ci enverront donc leur trames ethernet à la machine pirate tout en croyant communiquer avec la cible, et ce de façon transparente pour les switches. De son côté, la machine pirate stocke le traffic et le renvoie à la vraie machine en forgeant des trames ethernet comportant la vraie adresse MAC (indépendament de l'adresse IP). Cette technique est très puissante puisqu'elle opère au niveau ethernet, permettant ainsi de spoofer le traffic IP et même TCP (cela dépend entre autres des délais engendrés par la machine pirate). D'autre part, elle permet de contourner les barrières que constituent habituellement les switches (partitionnement de réseaux).

1.14.2 Conséquences

- Compromissions
- DoS (cache poisoning), etc...

1.14.3 Comment s'en protéger ?

- Avoir des logs régulièrement mis à jour et étudiés.
- N'utiliser que des tables ARP statiques.
- Utiliser des logiciels spécialisés pour monitorer les paires IP/MAC.

1.15 L'attaque Bonk

1.15.1 Qu'est-ce que c'est ?

L'attaque Bonk vise les systèmes WinNT 3.51 et 4.0. Elle consiste à envoyer des packets UDP corrompus sur le port 53. Chaque packet UDP corrompu est consitué de deux fragments IP assemblés en un UDP. Les offsets qui se superposent ont pour conséquence de faire écraser la seconde moitié de l'entête UDP par le second packet IP. L'ordinateur victime ne gère pas ces paquets et provoque un plantage (message STOP 0x0000000A) dû à une allocation excessive de la mémoire du noyau.

1.15.2 Conséquences

Blocage système Crash système

1.15.3 Comment s'en protéger ?

Mettre à jour l'OS (le fichier concerné est tcpip.sys). Utilisation d'un firewall pour refuser les packets UDP corrompus.

1.16 L'attaque BrKill

1.16.1 Qu'est-ce que c'est ?

L'attaque BrKill vise les systèmes Win32. Elle consiste à générer des packets qui génèrent un reset, permettant à l'attaquant de couper la connexion de la victime, à distance. Les transferts dits connectés (FTP, IRC, telnet, ICQ, ...) sont alors les cibles potentielles de cette attaque.

1.16.2 Conséquences

Déconnexion du réseau.

1.16.3 Comment s'en protéger ?

Protection inconnue

1.17 L'attaque Coke

1.17.1 Qu'est-ce que c'est ?

L'attaque Coke vise les systèmes WinNT qui exécutent le service WINS (Windows Internet Name Service). Elle consiste à se connecter à la cible et à envoyer n'importe quoi. En fonction de la configuration de l'ordinateur cible, celui ci inscrira un message d'erreur dans le log pour chaque

packet invalide reçu. Ceci a pour but de ralentir le système, et d'utiliser de place disque. De façon extrème, le disque peut arriver à saturation, écroulant considérablement les performances, et pouvant bloquer le système.

1.17.2 Conséquences

- Ralentissement système
- Blocage système
- Crash système

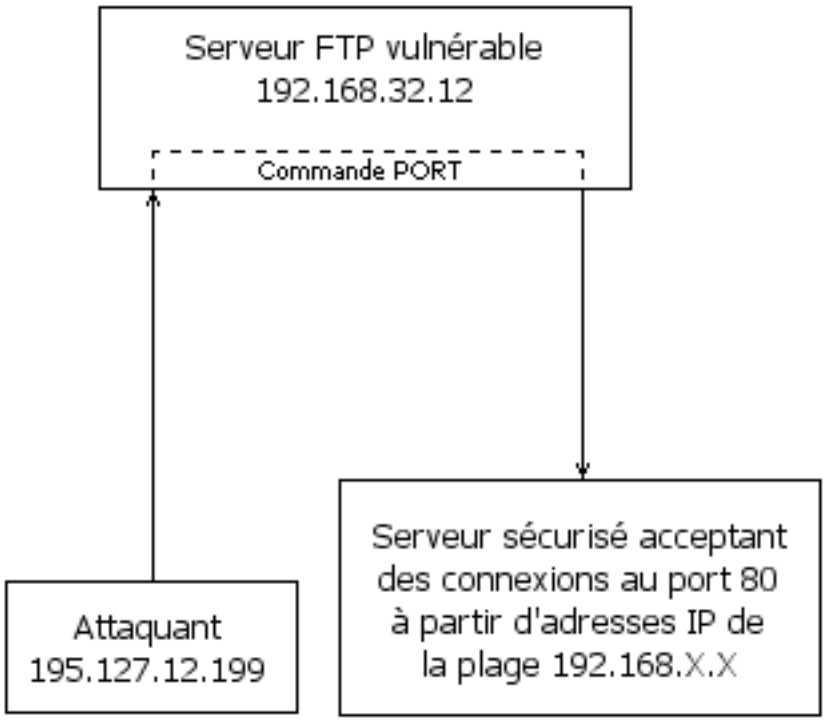
1.17.3 Comment s'en protéger ?

Configurer les routeurs et firewalls pour bloquer les packets dirigés vers le service WINS qui proviennent de l'extérieur de votre réseau.

1.18 Le FTP Bounce

1.18.1 Qu'est-ce que c'est ?

Le FTP Bounce signifie Rebond FTP. C'est un cas de spoofing d'adresse IP. Cette technique est ancienne et ne devrait plus être d'actualité. Cette technique est en accord avec les RFC, ce qui fait une cible potentielle de tous les serveurs FTP. Elle est basée sur une utilisation de la commande PORT du protocole FTP lorsque le serveur FTP est en mode actif. En effet, cette commande permet de se connecter à n'importe quel autre serveur distant, et à un port donné. Dans ce cas, il est possible que la sécurité du serveur cible soit compromis, dans le cas ou il effectue une vérification des adresses IP d'orinigine. En effet, l'adresse IP que le serveur cible verra sera l'adresse IP du serveur FTP, et non de l'adresse IP de l'attaquant. Ce petit schéma explique la technique utilisée :



1.18.2 Conséquences

Vol d'identité Permet d'accéder à des données confidentielles lorsque le serveur filtre les adresses IP entrantes.

1.18.3 Comment s'en protéger ?

Dû au fait que l'attaque est compatible avec le protocole (cf RFC), la politique de sécurité peut-être variable selon les implémentations. Nous suggérons de supprimer la commande PORT sauf dans le cas ou celle-ci est utilisée vers le client d'origine (celui qui a demandé la connexion FTP).

1.19 Land attack

1.19.1 Qu'est-ce que c'est ?

C'est une attaque axée réseaux, faisant partie de la grande famille des Refus De Service (DOS : Denial Of Service). Les systèmes visés sont Win32. Ce procédé est décomposé en trois étapes :

- La première est de récupérer l'adresse IP de la cible par spoofing.
- La seconde est de scanner tous les ports ouverts de l'ordinateur victime.
- La dernière est d'envoyer un packet à chaque port ouvert. Mais ces packets ont deux caractéristiques : Ils ont le flag SYN positionné, et les adresses source et destination du packet est la même : Celle de l'ordinateur cible.

1.19.2 Conséquences

Blocage système Crash système

1.19.3 Comment s'en protéger ?

Configurer le firewall ou le routeur pour filtrer les packets qui ont la même adresse IP en source et destination, sur tous les ports.

1.20 L'attaque NT Stop

1.20.1 Qu'est-ce que c'est ?

L'attaque NT Stop vise les systèmes WinNT 4.0. Elle consiste à envoyer une requête SMB logon avec une taille spécifiée incorrecte. L'ordinateur victime génère une corruption de mémoire causant une erreur "STOP 0x0000000A" ou "STOP 0x00000050" et provoque un plantage.

1.20.2 Conséquences

- Blocage système
- Crash système
- Reboot système

1.20.3 Comment s'en protéger ?

Mettre à jour l'OS.

1.21 L'attaque Oshare

1.21.1 Qu'est-ce que c'est ?

L'attaque Oshare vise tous les systèmes. Elle consiste à envoyer une entête IP invalide à la victime. Cette entête IP est rendue invalide en jouant sur la valeur des champs de l'entête qui spécifient la longueur du datagramme. Ce sont les champs "IHL" (mot de 32 bits indiquant la longueur du header) et "Total lenght" (mot de 16 bits indiquant la longueur du datagramme en octets, entête comprise), qui sont modifiés pour cette attaque. Les conséquences sont multiples : Elles dépendent du hardware de la carte réseau. Néanmoins, cette attaque ne peut porter que sur le même sous réseau. En effet, ces packets invalides (somme de contrôle IP incorrect dû à la mauvaise longueur du datagramme), ne peuvent passer les routeurs.

1.21.2 Conséquences

- Déconnexion du réseau
- Ralentissement système
- Blocage système
- Plantage système

1.21.3 Comment s'en protéger ?

Protection inconnue.

1.22 Ping flooding

1.22.1 Qu'est-ce que c'est ?

Ce procédé consiste à envoyer un flux maximal de ping vers une cible.

1.22.2 Qui peut provoquer cette attaque ?

N'importe qui, du moment qu'il a un logiciel permettant de le faire. Plus il a de personnes qui font un ping flooding vers une cible, et plus la situation devient critique pour cette cible..

1.22.3 Conséquences

Ralentissement système Blocage système Crash système

1.22.4 Comment s'en protéger ?

Utilisation d'un firewall.

1.23 L'attaque Pong

1.23.1 Qu'est-ce que c'est ?

L'attaque "Pong" est aussi connue sous le nom d'"Echo Reply Without Request" ou "ICMP echo reply attack". Elle consiste à envoyer à l'ordinateur victime le résultat d'un Ping (autrement dit, un Pong), alors que la victime n'a pas envoyé de Ping.

1.23.2 Conséquences

- Détermination de l'architecture réseau derrière un firewall. En effet, la plupart des firewall laissent passer les requêtes ping/pong. Si un routeur reçoit un pong à destination d'un ordinateur qui n'existe pas, il renverra un message "ordinateur inexistant (host unreachable)" à l'envoyeur, c'est à dire à l'attaquant. L'attaquant pourra donc déterminer le nombre de machines derrière le firewall, et plus encore, il aura les adresses IP de ces ordinateurs.
- Communication avec un cheval de Troie. Les requêtes ICMP passant sans problème par les firewalls, certains chevaux de Troie utilisent ces trames pour signaler leur présence.
- Attaques distribuées. Les requêtes ICMP passant sans problèmes par les firewalls, une attaque par abondance de requêtes Pong (type "flooding") permet de saturer un routeur ou un ordinateur derrière un firewall.
- Attaques "spoofed". Une attaque par ping flooding peut se produire contre un ordinateur en simulant que ces pings viennent de votre ordinateur (de votre adresse IP). Vous recevez donc uniquement les pongs provenant de l'ordinateur victime de l'attaque.

1.23.3 Comment s'en protéger

Utilisation d'un firewall pour enregistrer les pongs reçus alors qu'il n'y a pas eu de pings envoyés.

1.24 Les attaques Smack - Bloop

1.24.1 Qu'est-ce que c'est ?

Ces attaques visent tous les systèmes. Elles ressemblent à l'attaque Click : Elles consistent à envoyer des messages d'erreur ICMP (typiquement, ICMP inaccessible) à l'ordinateur cible. Mais, ces attaques n'ont pas pour but de déconnecter l'ordinateur victime. Elles provoquent un flood qui visent les transferts dits connectés (FTP, IRC, telnet, ICQ, ...).

1.24.2 Conséquences

Ralentissement des transferts connectés. Déconnexion des transferts connectés.

1.24.3 Comment s'en protéger ?

Configurer les firewalls/routeurs pour gérer ces messages.

1.25 L'attaque "Smurf"

1.25.1 Qu'est-ce que c'est ?

C'est un ping flooding un peu particulier. C'est une attaque axée réseaux, faisant partie de la grande famille des Refus De Service (DOS : Denial Of Service). Ce procédé est décomposé en deux étapes : La première est de récupérer l'adresse IP de la cible par spoofing. La seconde est d'envoyer un flux maximal de packets ICMP ECHO (ping) aux adresses de Broadcast. Chaque ping comportant l'adresse spoofée de l'ordinateur cible. Si le routeur permet cela, il va transmettre le broadcast à tous les ordinateurs du réseau, qui vont répondre à l'ordinateur cible. La cible recevra donc un maximum de réponses au ping, saturant totalement sa bande passante... Bien entendu, plus de réseau comporte de machines, plus c'est efficace...

1.25.2 Conséquences

- Perte de la bande passante
- Ralentissement système
- Perte du réseau
- Blocage système
- Crash système

1.25.3 Comment s'en protéger ?

Configurer le firewall pour filtrer les packets ICMP echo ou les limiter à un pourcentage de la bande passante. Configurer le routeur pour désactiver le broadcast.

1.26 L'attaque sPing - Jolt - IceNewk

1.26.1 Qu'est-ce que c'est ?

Cette attaque, qui comporte trois noms différents, visent les systèmes Win32. Elle consiste à envoyer un très grand nombre de packets ICMP très fragmentés à l'ordinateur victime. Les conséquences sont diverses : Les systèmes Win32 ont beaucoup de mal à s'y retrouver dans la défragmentation des packets.

1.26.2 Conséquences

- Blocage système.
- Etc...

1.26.3 Comment s'en protéger ?

- WinNT 4.0 : Installer le service Pack 3 + patch ICMP.
- WinNT 3.51 : Installer le service Pack 5 + patch ICMP.
- Win95 : Installer le patch ICMP.

1.27 L'attaque Tear Drop

1.27.1 Qu'est-ce que c'est ?

L'attaque Tear Drop vise les systèmes Win32 et Linux inférieur à 2.0.32. Elle consiste à envoyer des packets TCP qui se recouvrent. Lorsque l'ordinateur victime reçoit ces packets, il tente de les reconstruire. N'y arrivant pas, cela provoque un plantage.

1.27.2 Conséquences

Blocage système Crash système

1.27.3 Comment s'en protéger ?

Mettre à jour l'OS. Utilisation d'un firewall pour refuser les packets qui se recouvrent.

1.28 Les trous de sécurité applicatifs.

1.28.1 Qu'est-ce que c'est ?

Un trou de sécurité applicatif est le résultat d'un fonctionnement anormal d'une application. Il en résulte un plantage de l'application, ou bien un état non stable. Concrètement, il s'agit de trouver un fonctionnement que n'a pas prévu le programmeur. De ce fait, il est parfois possible d'en exploiter des failles. Cela devient très interessant lorsque c'est un programme réseaux (client/serveur, mail, www, architecture distribuée...). Ce type d'attaque peut être utilisée en local (pour obtenir l'accès root) ou à distance pour s'introduire dans un réseau, ou planter à distance un serveur.

1.28.2 Qui peut provoquer cette attaque ?

Un bon programmeur, qui a la connaissance d'un trou de sécurité, peut créer un programme qui va exploiter cette faille. Certains programmes existent déjà et exploitent les trous de sécurité les plus courants. Dans ce cas, n'importe qui est en mesure de pénétrer un réseau mal sécurisé.

1.28.3 Conséquences

- Obtention de l'accès root en local.
- Intrusion de réseaux ou d'ordinateur.
- Plantage à distance un serveur.

1.28.4 Comment s'en protéger ?

Se tenir au courant des logiciels qui comportent des failles. Quelques site web sont spécialisés dans ce domaine. Utiliser périodiquement un logiciel dédié à la surveillance du parc logiciel. Lorsqu'une version non fiable d'un logiciel est installée sur un poste, un message de mise en garde est affiché. Ce type de logiciel est disponible sur ce site web. Il s'appelle Securitor. Vous pouvez le trouver ici.

1.29 L'attaque UDP 0

1.29.1 Qu'est-ce que c'est ?

Cela consiste à envoyer une trame UDP vers le port zéro d'un ordinateur distant. L'ordinateur cible peut alors planter. Si celui-ci est derrière un firewall, le firewall peut éventuellement planter.

1.29.2 Conséquences

- Plantage système.
- Plantage du firewall.

1.29.3 Comment s'en protéger ?

Vérifiez que le firewall que vous utilisez gère correctement cette attaque. Bloquez les trames ayant un port de destination égal à zéro.

1.30 Les attaques WinArp - Poink

1.30.1 Qu'est-ce que c'est ?

Deux noms sont donnés pour une même attaque : WinArp et Poink. Elle vise les systèmes Win32. Elle consiste à envoyer plusieurs packets ARP à l'ordinateur victime. Pour chaque packet

ARP reçu, Windows affiche une boite de message avec un bouton "Ok". Imaginez si plusieurs milliers de packets ARP sont envoyés à la cible... L'origine de l'attaquant est connue car l'adresse MAC est à l'intérieur du packet ARP.

1.30.2 Conséquences

Ralentissement système Consommation des ressources systèmes

1.30.3 Comment s'en protéger ?

Pas de solutions connue.

1.31 L'attaque WINS 53 flood

1.31.1 Qu'est-ce que c'est ?

Cette attaque vise les systèmes WinNT qui exécutent le service WINS (Windows Internet Name Service). Elle consiste à envoyer un flot de packets aléatoires en taille et en contenu au port 53 (DNS) de l'ordinateur cible. Le serveur peut alors se planter.

1.31.2 Conséquences

Crash système

1.31.3 Comment s'en protéger ?

Installer le service pack 4

1.32 L'attaque WINS 137 flood

1.32.1 Qu'est-ce que c'est ?

Cette attaque vise les systèmes Win32 qui exécutent le service WINS (Windows Internet Name Service).. Elle consiste à envoyer un flot de packets UDP aléatoires en taille et en contenu au port 137 (UDP) de l'ordinateur cible. Après 5 secondes, le service WINS s'arrête. Le service doit être redémarré manuellement.

1.32.2 Conséquences

Perte du service WINS

1.32.3 Comment s'en protéger ?

Configurer le firewall et/ou les routeurs pour filtrer les packets UDP.

1.33 Le Cross Site Scripting

1.33.1 Introduction

Le Cross Site Scripting (CSS) est une attaque qui est rarement prise au sérieux par les non-initiés. En effet, à la différence de nombreuses techniques de piratage, celle-ci ne s'attaque pas à un serveur mais à l'internaute via une faille au niveau d'un serveur Web ou d'une application Web.

1.33.2 Le principe

Il est plus simple d'expliquer cette faille par l'exemple. Soit le site www.unsitecomplice.fr, le moyen de vérifier s'il est vulnérable à une attaque de type CSS est de demander l'affichage d'une page inexistante. La particularité du nom de cette page est le fait qu'il contient des balises HTML :

http ://www.unsitecomplice.fr/<B>nimportequoi</B>.html

Le site renvoie une page du type :

Erreur la page nimportequoi.html est introuvable.

Le site est bien vulnérable puisqu'il a renvoyé une page contenant le nom du fichier introuvable mais surtout parce que les balises HTML sont conservées. Le navigateur interprète donc le code HTML (ici les balises mettent simplement le texte en gras).

1.33.3 Les conséquences possibles

Les conséquences semblent anodines au premier abord. Mais le CSS est exploitable de la manière suivante. Un pirate va envoyer un mail en HTML à sa victime. Ce mail comporte un lien sur un site vulnérable à un CSS que la victime a pour habitude de visiter. Les conséquences deviennent graves à partir du moment où le code HTML passé dans l'URL permet l'exécution de Javascript (balise <SCRIPT>) sur la machine de l'utilisateur. Effectivement, le pirate connaissant bien le Javascript peut facilement récupérer le cookie de la victime utilisé sur ce même site. La gravité croît alors avec la sensibilité des informations contenues dans le cookie (authentification, identificateur de session, ...)

Le CSS est une technique déclinable pour les applications Web, cette attaque fonctionne dès que l'application restitue dans un message le nom d'un fichier ou d'un paramètre présent dans une URL sans prendre en compte l'éventuelle présence de balises HTML.

1.33.4 Comment éviter ce type de faille

Chaque élement de l'URL subissant un traitement doit obligatoirement être filtré afin d'ôter toutes les balises HTML. Une simple transformation peut suffir pour les rendre inexécutables (par exemple le caractère < est remplacé par &amplt ; que le navigateur affiche bien < mais sans l'interpréter).

1.34 Les buffers overflow

1.34.1 Qu'est-ce que c'est ?

Un buffer overflow est une attaque très efficace et assez compliquée à réaliser. Elle vise à exploiter une faille, une faiblesse dans une application (type browser, logiciel de mail, etc...) pour executer un code arbitraire qui compromettra la cible (acquisition des droits administrateur, etc...).

1.34.2 En bref

Le fonctionnement gébéral d'un buffer overflow est de faire crasher un programme en écrivant dans un buffer plus de données qu'il ne peut en contenir (un buffer est un zone mémoire temporaire utilisée par une application), dans le but d'écraser des parties du code de l'application et d'injecter des données utiles pour exploiter le crash de l'application.

Cela permet donc en résumé d'exécuter du code arbitraire sur la machine où tourne l'application vulnérable.

L'intérêt de ce type d'attaque est qu'il ne nécessite pas -le plus souvent- d'accès au système, ou dans le cas contraire, un accès restreint suffit. Il s'agit donc d'une attaque redoutable. D'un autre côté, il reste difficile à mettre en oeuvre car il requiert des connaissances avancées en programmation ; de plus, bien que les nouvelles failles soient largement publiées sur le web, les codes

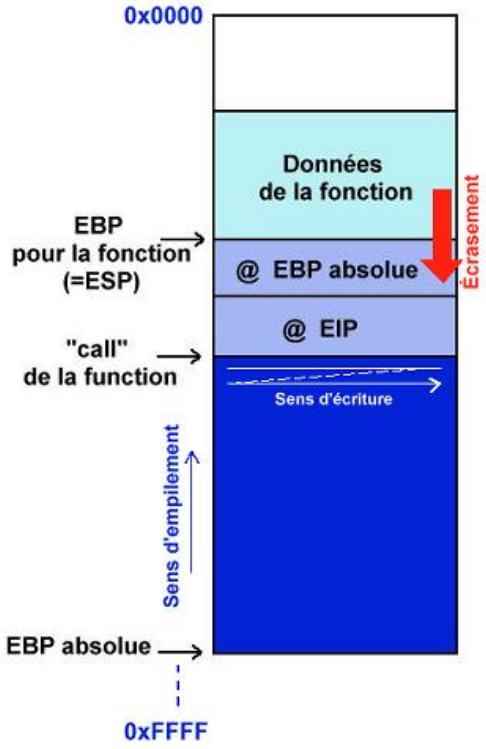
ne sont pas ou peu portables. Une attaque par buffer overflow signifie en général que l'on a affaire à des attaquants doués plutôt qu'à des "script kiddies".

1.34.3 Technique

Le problème réside dans le fait que l'application crashe plutôt que de gérer l'accès illégal à la mémoire qui a été fait. Elle essaye en fait d'accéder (lire, exécuter) à des données qui ne lui appartiennent pas puisque le buffer overflow a décalé la portion de mémoire utile à l'application, ce qui a pour effet (très rapidement) de la faire planter.

D'un point de vue plus technique, la pile (stack en anglais) est une partie de la mémoire utilisée par l'application pour stocker ses variables locales. Nous allons utiliser l'exemple d'une architecture intel (32 bits). Lors d'un appel à une sous-routine, le programme empile (push) le pointeur d'instruction (EIP) sur la stack et saute au code de la sous-routine pour l'exécuter. Après l'exécution, le programme dépile (pop) le pointer d'instruction et retourne juste après l'endroit où a été appelée la sous-routine, grâce à la valeur d'EIP. En effet, comme EIP pointe toujours vers l'instruction suivante, lors de l'appel de la sous-routine il pointait déjà vers l'instruction suivante, autrement dit l'instruction à exécuter après la sous-routine (= adresse de retour).

D'autre part, lors de l'appel de la sous-routine, celle-ci va dans la majorité des cas créer sa propre pile dans la pile (pour éviter de gérer des adresses compliquées). Pour cela elle va empiler la valeur de la base de la pile (EBP) et affecter la valeur du pointeur de pile (ESP) à celle de la base (EBP).



- ESP est le pointeur du sommet de la pile.
- EBP est le pointeur de la base de la pile.
- EIP est le pointeur de la prochaine instruction à exécuter. Il pointe donc toujours une exécution en avance.

En résumé, on sauvegarde la valeur originale de la base et on décale le tout ensuite. Lors du retour de la sous-routine, on dépile EBP et réaffecte sa valeur originale pour restaurer la pile initiale.

Voici pour le déroulement "normal" des opérations. Un point intéressant à citer est le fait que

dans notre architecture, les zones mémoires allouées dans la stack se remplissent dans le sens croissant des adresses (de 0..0H à F..FH) ce qui semble logique. Par contre, l'empilement sur la stack s'éffectue dans le sens décroissant ! C'est-à-dire que l'ESB originale est l'adresse la plus grande et que le sommet est 0..0H. De là naît la possibilité d'écraser des données vitales et d'avoir un buffer overflow. En effet, si notre buffer se trouve dans la pile d'une sous-routine et si nous le remplissons jusqu'à déborder sa taille allouée, nous allons écrire par-dessus les données qui se trouvent à la fin du buffer, c'est-à-dire les adresses qui ont été empilées précédement : EBP, EIP... Une fois la routine terminée, le programme va dépiler EIP et sauter à cette adresse pour poursuivre son exécution. Le but est donc d'écraser EIP avec une adresse différente que nous pourrons utiliser pour accéder à une partie de code qui nous appartient. (par exemple le contenu du buffer)

Un problème à ce stade est de connaitre l'adresse exacte de la stack (surtout sous Windows) pour pouvoir sauter dedans. On utilise généralement des astuces propres à chaque système (librairies, etc..) qui vont permettre -indirectement- d'atteindre notre stack et d'exécuter notre code. Cela nécessite un débogage intensif qui n'est pas à la portée de tout le monde...

1.34.4 Solutions

- lors du développement : propreté du source (utiliser malloc/free le plus possible, utiliser les fonctions n comme strncpy pour vérifier les limites...), utilisation de librairies de développement spécialisée contre les buffers overflow (Libsafe d'Avayalabs)
- utiliser un langage n'autorisant pas ce type d'attaques : Java, Cyclone (qui est issu du C).
- utiliser des logiciels spécialisés dans la vérification de code, comme par exemple le compilateur StackGuard d'Immunix.
- appliquer le plus rapidement possible les patches fournis par les développeurs.

1.35 Le Déni de Service (DoS)

1.35.1 Background

Le "Denial-of-service" ou déni de service est une attaque très évoluée visant à rendre muette une machine en la submergeant de traffic inutile. Il peut y avoir plusieurs machines à l'origine de cette attaque (c'est alors une attaque distribuée, voir fiche DDoS) qui vise à anéantir des serveurs, des sous-réseaux, etc. D'autre part, elle reste très difficile à contrer ou à éviter.

1.35.2 Définition

D'une manière générale, on parle de déni de service quand une personne ou une organisation est privée d'un service utilisant des ressources qu'elle est en droit d'avoir en temps normal. On trouvera par exemple des dénis de service touchant le service de courrier électronique, d'accès à Internet, de ressources partagées (pages Web), ou tout autre service à caractère commercial comme Yahoo ! ou EBay. Quoiqu'il en soit, le déni de service est un type d'attaque qui coûte très cher puisqu'il interrompt le cours normal des transactions pour une entreprise ; les sommes et les enjeux sont énormes et cela ne peut aller qu'en s'agravant tant que des parades réellement efficaces n'auront pas été trouvées.

1.35.3 Types d'attaques

Il existe de nombreuses façons pour faire planter une machine ; en fait, on peut s'appuyer sur presque toutes les attaques existantes (voir sur la page d'accueil de SecuriteInfo) car en faisant planter la machine cela résultera inévitablement en un déni de service sur cette machine. La différence se situe au niveau des intentions du hacker, c'est-à-dire savoir si le déni de service est intentionnel ou s'il n'est que la résultante d'une attaque plus aggressive visant à détruire une machine. Il ne faut plus aujourd'hui négliger cet aspect intentionnel, au vu des sommes qui entrent en jeu.

Parmi les attaques propres à créer un déni de service, nous pouvons rappeler entre autres :
- les buffers overflows (mails, ping of Death...)
- l'attaque SYN
- l'attaque Teardrop
- l'attaque SMURF
- les virus
- ...

1.35.4 Contre-mesures

Les contre-mesures sont très compliquées à mettre en place et très ciblées vis-à-vis du type de déni de service envisagé. En effet, d'un point de théorique, la plupart des attaques visant à créer des dénis de service sont basées sur des services ou protocoles normaux sur Internet. S'en protéger reviendrait à couper les voies de communications normales avec Internet, alors que c'est bien là la raison d'être principale des machines concernées (serveurs web, etc...). Il reste tout de même la possibilité de se protéger contre certains comportement anormaux (voir les attaques précédentes) comme une tentative de flooding, un trop grand nombre de paquets ou de requêtes de connexion provenant d'un petit nombre de machines. Mais cela implique beaucoup de choses en fait : il faut monitorer le traffic (ce qui est loin d'être simple, du fait de la quantité de données qui transitent), établir des profils types de comportement et des écarts tolérables au-delà desquels on considérera que l'on a affaire à une attaque ; il faut également définir les types d'attaques auxquelles on souhaite se protéger (analyses de risques à l'appui) car il est impossible de toutes les prévoir. On est donc loin de la protection absolue ; il s'agit de metter en place une protection intelligente et flexible.

C'est ce que l'on retrouve à l'heure actuelle dans la plupart des systèmes de protection contre les dénis de service. Ainsi Cisco propose des produits incluant à différents niveaux des services spécifiques :
- test de la taille des paquets
- test des adresses source et destination (ainsi que loop-back, unicast, multicast...)
- test de la fragmentation
- utilisation d'adresses IP virtuelles pour validation de sessions et ACK (contre attaques TCP)
- test du nombre de SYN (contre attaques TCP)
- NAT d'adresses locales vers IP virtuelles basées sur IP globales
- contrôles de flux
- contrôles de contenus (port, tag, url, extensions de fichiers)
- autres fonctions de firewall, le tout basé sur du load-balancing et de la redondance.

Comme on peut le remarquer, l'accent est mis sur la sécurité du système de protection en lui-même pour qu'il puisse faire face à des situations extrêmes (traffic énorme, etc...) D'autre part, il reste que la plupart des contre-mesures visent à protéger contre un type d'attaque particulier. L'efficacité d'un tel système se révèlera par sa capacité à détecter et prévenir de nouveaux types d'attaques.

1.36 Le Déni de Service Distribué (DDoS)

1.36.1 Background

Le "Distributed denial-of-service" ou déni de service distribué est un type d'attaque très évolué visant à faire planter ou à rendre muette une machine en la submergeant de traffic inutile (voir fiche DoS). Plusieurs machines à la fois sont à l'origine de cette attaque (c'est une attaque distribuée) qui vise à anéantir des serveurs, des sous-réseaux, etc. D'autre part, elle reste très difficile à contrer ou à éviter. C'est pour cela que cette attaque représente une menace que beaucoup craignent.

1.36.2 Les outils

Pour mieux comprendre le phénomène, il parait impossible de ne pas étudier les outils les plus importants dans ce domaine, qui doivent leur notoriété à des célèbres attaques ayant visé des grands sites sur le net.

Un réseau typique se compose donc d'un maitre (point central) et de nombreux hôtes distants, encore appelés démons. Pendant le déroulement de l'attaque, le hacker se connecte au maitre qui envoie alors un ordre à tout les hotes distants (via UDP, TCP ou ICMP). Ces communications peuvent également dans certains cas être chiffrées. Ensuite, les hôtes distants vont attaquer la cible finale suivant la technique choisie par le hacker. Ils vont par exemple se mettre à envoyer un maximum de paquets UDP sur des ports spécifiés de la machine cible. Cette masse de paquets va submerger la cible qui ne pourra plus répondre à aucune autre requête (d'où le terme de déni de service). D'autres attaques existent, tel que l'ICMP flood, le SYN flood (TCP), les attaques de type smurf, les attaques dites furtives, les attaques de déni de service dites aggressives (dont le but est bel et bien de faire crasher complètement la cible), ou encore des attaques de type "stream attack" (TCP ACK sur des ports au hasard)... Certains outils se sont même inspiré des chevaux de Troie (voir fiche sur les chevaux de Troie) qui installent de petits serveurs irc permettant au hacker de les commander via cette interface.

1.36.3 Mode opératoire

Les DDoS se sont démocratisées depuis 2-3 ans. En effet dans les premiers temps, cette attaque restait assez compliquée et nécessitait de bonnes connaissances de la part des attaquants ; mais ceux-ci ont alors développé des outils pour organiser et mettre en place l'attaque. Ainsi le processus de recherche des hôtes secondaires (ou zombies) a été automatisé. On cherche en général des failles courantes (buffer overflows sur wu-ftpd, les RPCs...) sur un grand nombre de machines sur Internet et l'attaquant finit par se rendre maître (accès administrateur) de centaines voir de milliers de machines non protégées. Il installe ensuite les clients pour l'attaque secondaire et essaye également d'effacer ses traces (corruption des fichiers logs, installation de rootkits). Une fois le réseau en place, il n'y a plus qu'à donner l'ordre pour inonder la victime finale de paquets inutiles.

Il est intéressant de noter que les victimes dans ce type d'attaques ne sont pas que celles qui subissent le déni de service ; tous les hôtes secondaires sont également des machines compromises jusqu'au plus haut niveau (accès root), tout comme l'hôte maître.

La menace provient du fait que les outils automatisant le processus ont été très largement diffusés sur Internet. Il n'y a plus besoin d'avoir des connaissances pointues pour la mettre en place, il suffit de "cliquer" sur le bouton.

1.36.4 Contre-mesures

Il n'est pas évident de se prémunir contre ces attaques par déni de service, car la mise en place du réseau offensif par l'attaquant repose sur le fait que beaucoup de machines sont peu ou pas sécurisée et présentent des failles. Ces failles sont tellement nombreuses et d'autre part il existe tellement de machines vulnérables sur Internet qu'il devient impossible d'empêcher de telles attaques. Ainsi, si un outil de DDoS est détecté sur un système, cela signifie surement que il a été installé sur de nombreux autres systèmes sans être décelé. D'autre part, la présence de cet outil signifie également que le système a été intégralement compromis, qu'il présente surement des bakdoors et qu'on y a peut-être installé un rootkit (type Adore). Il est donc urgent et nécessaire de retirer complètement cette machine du réseau et de l'inspecter pour éventuellement la réinstaller.

Pour détecter un tel outil, on pourra chercher des noms évocateurs parmi les processus système s'il n'y a pas de rootkit installé et si l'attaquant à laissé un nom par défaut. Ces noms peuvent être regroupés dans la liste suivante (non exhaustive) :

Trinoo maître : master

Broadcast : ns

TFN client : tfn

Démon : td

Stacheldraht Handler : mserv

Agent : td

Shaft Handler : shaftmaster

Agent : shaftnode mstream

Handler : master

Agent : server

Trinity Agent : /usr/lib/idle.so

Portshell : /var/spool/uucp/uucico

Alt. Portshell : /var/spool/uucp/fsflush

Il peut être également fort utile de connaitre les outils (au nombre de 4, principalement) utilisés par les hackers. Des analyses sont disponibles pour 3 d'entres eux :

http ://staff.washington.edu/dittrich/misc/trinoo.analysis

http ://staff.washington.edu/dittrich/misc/tfn.analysis

http ://staff.washington.edu/dittrich/misc/stacheldraht.analysis

Il existe également un compte-rendu de congrès scientifique sur le sujet qui apporte beaucoup d'idées pour se défendre et récupérer après de tels incidents :

http ://www.cert.org/reports/dsit_workshop.pdf

1.36.5 Le Pushback : une contre-mesure en développement

Face aux menaces grandissantes provoquées par ce type d'attaques, les scientifiques se penchent de plus en plus sur des techniques capables de les contrer ; une des plus récentes est la technique du Pushback. Nous ne rentrerons pas dans les détails ici, tous les papiers étant disponibles sur le site ACC and Pushback.

Très brièvement, cette technique a pour but d'identifier les attaques de DoS et surtout de DDoS grâce à des heuristiques, de les contrer en remontant à leur source, enfin de maintenir et de protéger le bon traffic qui souffre également la plupart du temps des congestions engendrées par de telles attaques. Cette méthode utilise un contrôle de congestion basé sur des aggrégats, un aggrégat étant défini comme un sous-ensemble du traffic présentant une propriété identifiable. Exemples de propriétés :

- Paquets TCP SYN
- Paquets à destination de X
- Paquets IP dont les checksums sont incorrects
- ...

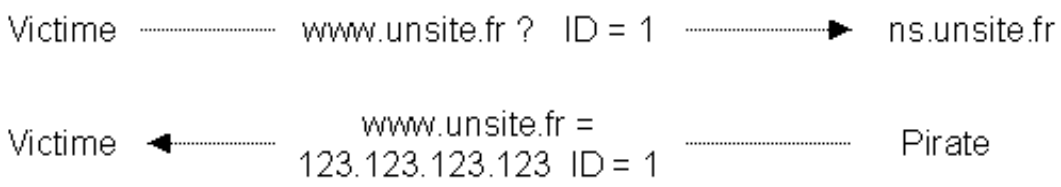
Le but est d'identifier les aggrégats responsables de la congestion et de les éliminer pour rétablir un traffic normal. Une fois la signature (c'est-à-dire la propriété identifiante, le trait caractéristique de l'attaque) établie, le flux est comparé en temps réel dans le routeur le plus proche de la cible du DDoS. Ce routeur commence à rejeter (drop) les paquets correspondants à la signature et envoie également un message d'alerte aux routeurs en amont sur les brins d'où lui parvient le traffic incriminé. Ce message d'alerte contient entre autres choses la signature qui va permettre à ces routeurs d'éliminer à leur tour les paquets correspondants à l'attaque. Et ces routeurs vont également envoyer des messages d'alerte aux routeurs situés en amont. Cette technique récursive a pour avantage de pouvoir remonter jusqu'aux sources de l'attaque ; elle permet également de décongestionner le coeur même du réseau, ce qui était impossible avec les techniques centrées sur la protection pure de la cible. Enfin, même si une partie du traffic légitime est tout de même perdue, les résultats finaux sont plutôt positifs.

1.37 Le DNS Spoofing

1.37.1 Qu'est-ce que c'est ?

L'objectif de cette attaque est de rédiriger, à leur insu, des Internautes vers des sites pirates. Pour la mener à bien, le pirate utilise des faiblesses du protocole DNS (Domain Name System) et/ou de son implémentation au travers des serveurs de nom de domaine. A titre de rappel, le protocole DNS met en oeuvre les mécanismes permettant de faire la correspondance entre une adresse IP et un nom de machine (ex. : www.truc.com). Il existe deux principales attaques de type DNS Spoofing : le DNS ID Spoofing et le DNS Cache Poisoning. Concrètement, le but du pirate est de faire correspondre l'adresse IP d'une machine qu'il contrôle à un nom réel et valide d'une machine publique. Description de l'attaque DNS ID Spoofing Si une machine A veut communiquer avec une machine B, la machine A a obligatoirement besoin de l'adresse IP de la machine B. Cependant, il se peut que A possède uniquement le nom de B. Dans ce cas, A va utiliser le protocole DNS pour obtenir l'adresse IP de B à partir de son nom.

Une requête DNS est alors envoyée à un serveur DNS, déclaré au niveau de A, demandant la résolution du nom de B en son adresse IP. Pour identifier cette requête une numéro d'identification (en fait un champs de l'en-tête du protocole DNS) lui est assigné. Ainsi, le serveur DNS enverra la réponse à cette requête avec le même numéro d'identification. L'attaque va donc consister à récupérer ce numéro d'identification (en sniffant, quand l'attaque est effectuée sur le même réseau physique, ou en utilisant une faille des systèmes d'exploitation ou des serveurs DNS qui rendent prédictibles ces numéros) pour pouvoir envoyer une réponse falsifiée avant le serveur DNS. Ainsi, la machine A utilisera, sans le savoir, l'adresse IP du pirate et non celle de la machine B initialement destinatrice. Le schéma ci-dessous illustre simplement le principe du DNS ID Spoofing.



1.37.2 DNS Cache Poisoning

Les serveurs DNS possèdent un cache permettant de garder pendant un certain temps la correspondance entre un nom de machine et son adresse IP. En effet, un serveur DNS n'a les correspondances que pour les machines du domaine sur lequel il a autorité. Pour les autres machines, il contacte le serveur DNS ayant autorité sur le domaine auquel appartiennent ces machines. Ces réponses, pour éviter de sans cesse les redemander aux différents serveurs DNS, seront gardées dans ce cache. Le DNS Cache Poisoning consiste à corrompre ce cache avec de fausses informations. Pour cela le pirate doit avoir sous son contrôle un nom de domaine (par exemple fourbe.com) et le serveur DNS ayant autorité sur celui-ci ns.fourbe.com.

L'attaque se déroule en plusieurs étapes :

- Le pirate envoie une requête vers le serveur DNS cible demandant la résolution du nom d'une machine du domaine fourbe.com (ex. : www.fourbe.com)
- Le serveur DNS cible relaie cette requête à ns.fourbe.com (puisque c'est lui qui a autorité sur le domaine fourbe.com)
- Le serveur DNS du pirate (modifié pour l'occasion) enverra alors, en plus de la réponse, des enregistrements additionnels (dans lesquels se trouvent les informations falsifiées à savoir un nom de machine publique associé à une adresse IP du pirate)
- Les enregistrements additionnels sont alors mis dans le cache du serveur DNS cible

- Une machine faisant une requête sur le serveur DNS cible demandant la résolution d'un des noms corrompus aura pour réponse une adresse IP autre que l'adresse IP réelle associée à cette machine.

1.37.3 Comment s'en protéger ?

- Mettre à jour les serveurs DNS (pour éviter la prédictibilité des numéros d'identification et les failles permettant de prendre le contrôle du serveur)
- Configurer le serveur DNS pour qu'il ne résolve directement que les noms des machines du domaine sur lequel il a autorité
- Limiter le cache et vérifier qu'il ne garde pas les enregistrements additionnels.
- Ne pas baser de systèmes d'authentifications par le nom de domaine : Cela n'est pas fiable du tout.

1.38 L'IP Spoofing

1.38.1 Qu'est-ce que c'est ?

L'IP Spoofing signifie usurpation d'adresse IP. Bien que cette attaque soit ancienne, certaines formes d'IP Spoofing sont encore d'actualité. Effectivement, cette attaque peut être utilisée de deux manières différentes :

- La première utilité de l'IP Spoofing va être de falsifier la source d'une attaque. Par exemple, lors d'une attaque de type déni de service, l'adresse IP source des paquets envoyés sera falsifiée pour éviter de localiser la provenance de l'attaque.
- L'autre utilisation de l'IP Spoofing va permettre de profiter d'une relation de confiance entre deux machines pour prendre la main sur l'une des deux. Il s'agit de cette attaque dont il va être question ici.

1.38.2 Description de l'attaque

Il existe plusieurs types d'IP Spoofing. La première est dite Blind Spoofing, c'est une attaque "en aveugle". Les paquets étant forgés avec une adresse IP usurpée, les paquets réponses iront vers cette adresse. Il sera donc impossible à l'attaquant de récupérer ces paquets. Il sera obligé de les "deviner". Cependant, il existe une autre technique que le Blind Spoofing. Il s'agit d'utiliser l'option IP Source Routing qui permet d'imposer une liste d'adresses IP des routeurs que doit emprunter le paquet IP. Il suffit que l'attaquant route le paquet réponse vers un routeur qu'il contrôle pour le récupérer. Néanmoins, la grande majorité des routeurs d'aujourd'hui ne prennent pas en compte cette option IP et jettent tous paquets IP l'utilisant.

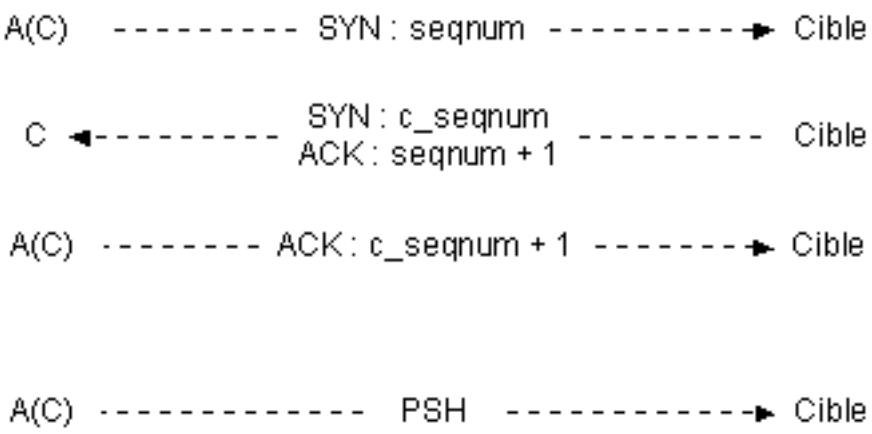
La cible de cette attaque sera un service de type rlogin ou rsh (qui ne sont plus très utilisés de nos jours). Ce sont des services avec une authentification basée sur l'adresse IP de la machine cliente. En outre, les protocoles utilisés par ces services sont simples, les réponses attendues n'en sont que plus facilement prédictibles quand il y en a. A noter : L'IP Spoofing peut être utilisé pour faire du DNS Spoofing...

Cette attaque va se dérouler en plusieurs étapes :

- Trouver la machine de confiance (son adresse IP) qu'accepte le service du serveur cible.
- Mettre hors service cette machine de confiance (avec un SYN Flooding par exemple) pour éviter qu'elle ne réponde aux paquets éventuellement envoyés par le serveur cible.
- Prédire les numéros de séquence TCP du serveur cible. Ce numéro caractérise une connexion TCP (un numéro de séquence initial est généré à chaque nouvelle connexion TCP). C'est un champ de l'en-tête du protocole TCP. De nos jours ce numéro est difficilement prédictible voir impossible sur des systèmes type Linux. Ce qui n'était pas le cas il y a quelques années.
- Lancer l'attaque. Elle va consister à créer une connexion TCP sur le serveur cible. Pour cela, l'attaquant va forger un paquet TCP avec le flag SYN et l'adresse IP source de la machine de confiance. Le serveur cible va répondre par un paquet TCP avec les flags SYN-ACK.

L'attaquant, qui aura prédis le numéro de séquence TCP, pourra forger un paquet TCP avec le flag ACK et le bon numéro d'acquittement (numéro de séquence envoyé par le serveur cible incrémenté de 1). Une connexion TCP est alors établie au niveau du serveur cible. L'attaquant n'a plus qu'à envoyer un paquet TCP avec le flag PSH (permettant de remonter directement à l'application les données du paquet) pour envoyer une commande au service (par exemple echo ++ >> /.rhosts). L'attaquant peut accéder librement au serveur cible.

Le schéma suivant illustre cette attaque. La machine A est celle de l'attaquant, la C celle de confiance et enfin Cible qui est le serveur cible. A(C) signifie que la machine A va envoyer un paquet en spoofant l'adresse IP de la machine C :



1.38.3 Conséquences

- Usurpation d'identité.
- Prise de contrôle du serveur cible.

1.38.4 Comment s'en protéger ?

- Supprimer tous les services de type rsh et rlogin.
- Ne pas utiliser uniquement l'adresse IP comme méthode d'authentification (ajouter au moins un login et un password).
- Vérifier que son système n'a pas des numéros de séquence TCP facilement prédictible.
- Vérifier que la fonction anti-spoofing est bien présente sur le firewall.

1.39 Dictionary Cracking

1.39.1 Qu'est-ce que c'est ?

Généralement les mots de passe de la plupart des logiciels sont stockés cryptés dans un fichier. Pour obtenir un mot de passe, il suffit de récupérer ce fichier et de lancer un logiciel de dictionary cracking. Ce procédé consiste à utiliser un logiciel qui va tester un nombre limité de mots, de manière à trouver un mot de passe. Chaque mot est en fait un mot d'usage courant, un prénom, un nom, ou une abbréviation. Tous les mots utilisés sont regroupés dans un seul fichier : un dictionnaire. D'où le nom de cette attaque.

1.39.2 Qui peut provoquer cette attaque ?

N'importe qui, du moment qu'il a un logiciel permettant de le faire et qu'il a récupéré le fichier de mots de passe.

1.39.3 Conséquences

Un mot de passe sera rapidement trouvé s'il est emprunté du langage courant.

1.39.4 Comment s'en protéger ?

Pour prévenir une telle attaque, il y a quelques règles de base à respecter :
- N'utilisez pas de mot de passes signifiant quelquechose.
- Utilisez plutôt une combinaison de chiffres et de lettres. Changer vos mots de passe régulièrement.

L'utilisation d'autres techniques pour la protection par mots de passe est recommandée.

1.40 Brute Force Cracking

1.40.1 Qu'est-ce que c'est ?

Généralement les mots de passe de la plupart des logiciels sont stockés cryptés dans un fichier. Pour obtenir un mot de passe, il suffit de récupérer ce fichier et de lancer un logiciel de brute force cracking. Ce procédé consiste à tester de façon exhaustive toutes les combinaisons possibles de caractères (alphanumériques + symboles), de manière à trouver au moins un mot de passe valide. Cette attaque se base sur le fait que n'importe quel mot de passe est crackable. Ce n'est qu'une histoire de temps. Mais la puissance des machines double tous les deux ans. On parle de plus en plus de processeurs 1,2 GHz... De plus, les crackers n'hésitent pas à fabriquer des cartes électroniques de cracking, ce qui améliore en conséquence la rapidité de la machine, et donc les chances de trouver un mot de passe valide. En général, cette méthode est empruntée lorsque la méthode du dictionary cracking a échoué.

1.40.2 Qui peut provoquer cette attaque ?

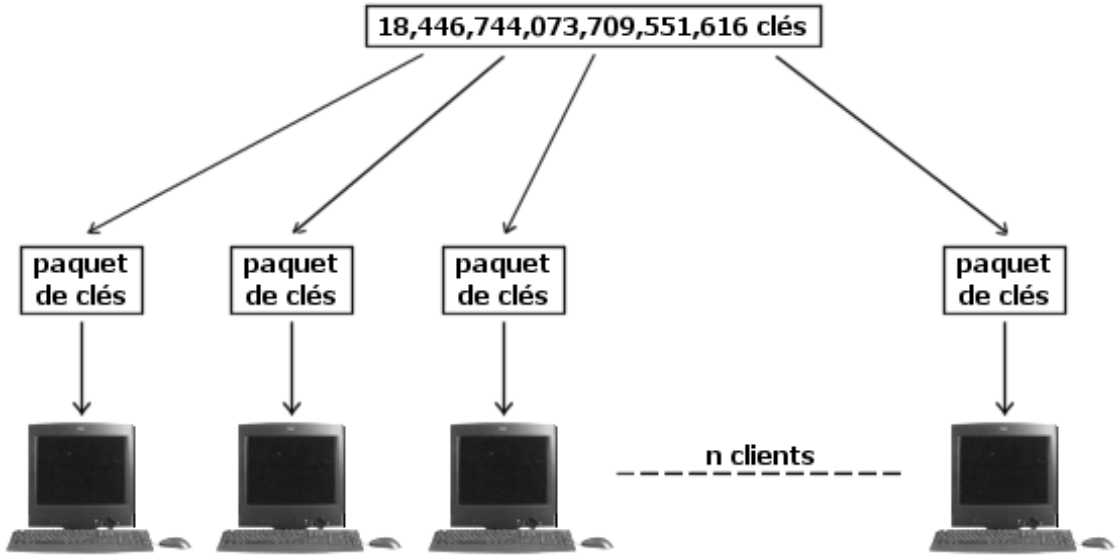
N'importe qui, du moment qu'il a un logiciel permettant de le faire et qu'il a récupéré le fichier de mots de passe.

1.40.3 Conséquences

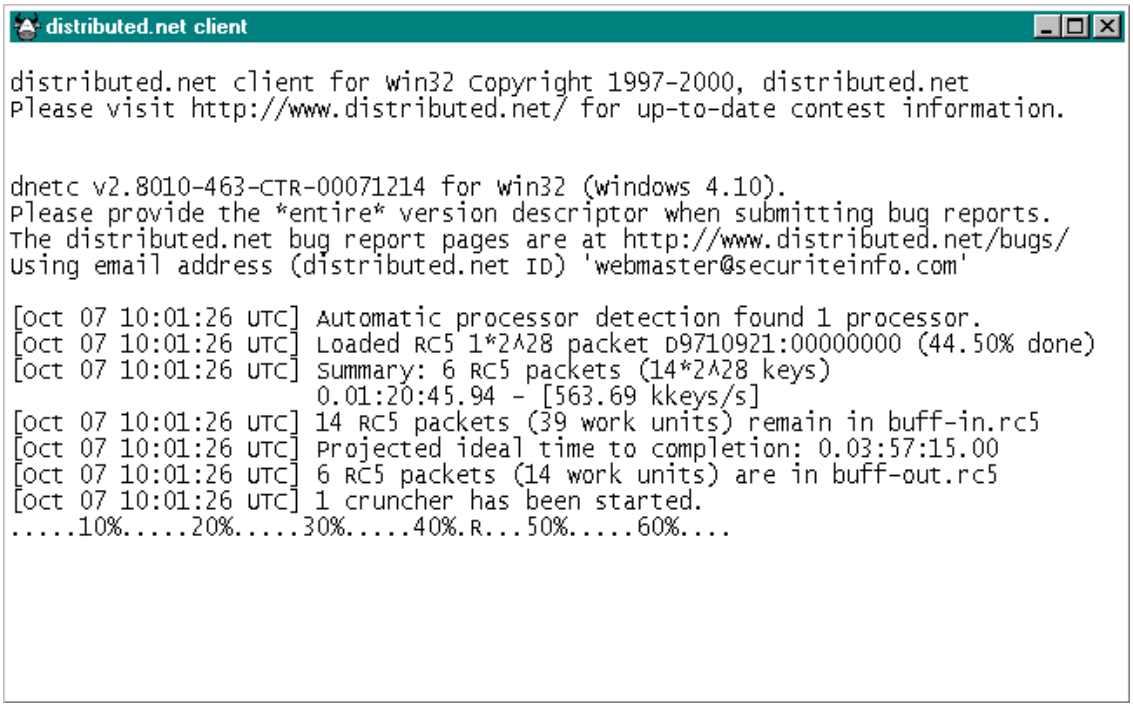
Les protections par mot de passe classique (de type login Unix) sont voués à disparaitre dans un futur proche...

1.40.4 Un exemple : distributed.net

Le RSA, organisme américain chargé entre autres de tester les systèmes de sécurité, a lancé un défi à la planète entière : L'organisation donnera 10000 dollars à quiconque décrypterait un message codé avec une clé RC5 de 64 bits. Le RC5 est un algorithme de cryptage. Le codage se faisant sur 64 bits, il y a donc au total 264 clés (1 bit=0 ou 1), ce qui représente 18,446,744,073,709,551,616 possibilités ! Une autre organisation, distributed.net, a décidé de relever le défi, en se basant sur une méthode de brute force cracking distribué. Cela veut dire que toutes les clés seront testées. Pour effectuer cela, le nombre total de clés est divisé en de multiples petits paquets, qui sont ensuite envoyés à des ordinateurs clients. Ces clients calculeront chacun les paquets de clés reçus, en utilisant un logiciel nommé DNETC fourni par distributed.net.



Une fois qu'un client a fini de calculer un paquet, il renvoie son résultat et reprend un autre paquet à calculer. Et ceci jusqu'à ce que la clé unique qui permet de décrypter le message soit trouvée.



Vous pouvez, si vous le désirez, participer à ce défi. Pour plus d'informations, n'hésitez pas à consulter :

- Une excellente page d'informations en français sur ce défi (rubrique RC5).
- Le site officiel de distributed.net dont une bonne partie est en français.

1.40.5 Comment s'en protéger ?

Utilisation d'autres techniques pour la protection par mots de passe : Les méthodes d'authentification forte.

1.41 Tempest

1.41.1 Qu'est-ce que c'est ?

Tempest est un dispositif électronique permettant de capter les émissions électromagnétiques générées par un appareil électrique. Ce type de matériel est considéré comme faisant partie des outils d'espionnage.

1.41.2 A quoi cela sert-il ?

A l'origine, Tempest permet de capter les émissions électromagnétiques d'un écran d'ordinateur, afin de les restituer sous la forme d'une image. En clair, c'est de la duplication d'image à distance. Tout ce que l'ordinateur espionné affiche sur son écran, Tempest le capte et reconstruit l'image en temps réel. Ce n'est plus la peine d'espionner les paquets TCP/IP pour comprendre ce que fait la victime. L'image est bien plus parlante que des bits !

1.41.3 Comment créer un système Tempest ?

Pour créer un système Tempest, il faut se procurer un poste radio classique, que l'on va modifier quelque peu pour capter la bande de fréquences des ondes générées par les écrans. Ces fréquences ainsi captées doivent ensuite être filtrées, mises en forme et enfin transformées en une image. Penser à capter les signaux de synchronisation de l'image (VBL et HBL). Nous n'allons pas détailler plus cet aspect qui sort du cadre de cette documentation.

1.41.4 Les avantages

L'intérêt de Tempest est qu'il peut espionner à distance. Il est indétectable.

1.41.5 Les limitations

La portée d'un tel dispositif est approximativement de 100 mètres. Les écrans à cristaux liquides génèrent beaucoup moins de signaux. C'est à l'heure actuelle la solution la plus économique de se prévenir de Tempest. Il est à noter que certains constructeurs certifient leurs ordinateurs "Anti-Tempest".

1.41.6 Le futur

Ce système peut être étendu à toute émission électromagnétique, générée par un CPU, un clavier, un disque dur... Mais la transformation de ces informations en données compréhensibles par l'homme est beaucoup moins évidente. Paraît-il que des résultats concluants ont été réalisés avec un système de ce type qui permettait de capter des signaux électromagnétiques sur les canalisations d'eau, mais aussi sur les fils électriques domestiques...

1.42 Les cartes magnétiques

1.42.1 Introduction

Issus d'une technologie du début du 20ème siècle, l'enregistrement et la lecture magnétique sont encore très utilisés de nos jours. En effet, il existe de nombreuses utilisations des supports magnétiques. A titre d'exemple, peuvent être cités les tickets avec une bande magnétique (métro, bus, train, parking,...), les bandes permettant le stockage de données (cassettes audio et vidéo,...) et surtout les cartes avec une piste magnétique (carte bleue, carte de fidélité, carte d'abonnement, carte de contrôle d'accès,...).

1.42.2 Enregistrement et lecture magnétique

Principe général

Le principe de l'enregistrement magnétique repose sur la magnétisation de très petites zones de la bande magnétique constituée de pigments magnétiques (oxyde de fer, oxyde de chrome ou ferrite de baryum). Cette opération de magnétisation est effectuée par une tête magnétique d'écriture. En fait, il s'agit d'un genre d'électro-aimant. En passant sur la bande magnétique, pour une opération d'écriture, la tête va plonger les pigments dans un champ magnétique proportionnel au courant la traversant. Cette magnétisation va subsister et correspondra alors à un enregistrement. Une caractéristique importante des supports magnétiques est leur champ coercitif ou coercitivité. C'est tout simplement leur résistance à la désaimantation. Une distinction est donc faite entre les supports HiCo (haute coercitivité) et LoCo (basse coercitivité). Par conséquent, un support dit HiCo pourra être désaimanté plus difficilement qu'un support LoCo. Pour l'opération de lecture, le passage de la tête sur la bande donnera naissance à un flux magnétique dans son noyau, lequel induira une tension électrique proportionnelle aux variations du flux. Le signal électrique (c'est à dire les informations) préalablement enregistré sur la bande magnétique est alors restitué.

Les données numériques

Le principe général est parfaitement adapté à l'enregistrement et la lecture de données analogiques (comme le son par exemple). Concernant l'enregistrement de données numériques (c'est à dire un signal avec seulement deux états à savoir le 1 et le 0) cette technique ne paraît poser aucun souci. Néanmoins, un problème se pose lors de la relecture puisque qu'il est alors impossible de séparer précisément une suite de 1 ou de 0. Effectivement, seule la transition entre l'état 1 et l'état 0 est marquée par une tension électrique contrairement à une succession de 1 (ou de 0) où n'apparaît aucun changement d'état donc de tension. Pour palier à cela un codage spécial pour l'enregistrement a été adopté : le codage F/2F.

Ce codage est basé sur l'enregistrement par inversion de flux. Cette technique consiste à faire circuler le courant, dans la tête, dans un sens puis dans l'autre. Il y aura donc uniquement deux orientations diamétralement opposées des pigments constituant le support magnétique. Le codage F/2F est en fait une évolution de cette technique. Dans ce codage le 0 sera alors représenté par une inversion de flux en début et en fin de bit tandis que le 1 aura une inversion supplémentaire en milieu de bit. Néanmoins, la "durée" (la longueur du support magnétique occupé) sera identique pour le 1 et le 0. Le bit 1 aura donc une fréquence double par rapport au 0 (d'où F/2F). Les cartes magnétiques normalisées Il s'agit des cartes ayant une piste magnétique les plus couramment utilisées (cartes bleues, cartes de fidélité,...). Elles respectent la norme ISO 7811 (composée de 5 parties). Celle-ci définit, entre autres, les trois pistes constituant la bande magnétique de la carte nomalisée : les piste ISO 1, ISO 2 et ISO 3. Ces 3 pistes sont caractérisées par leur positionnement, leur densité d'enregistrement et l'encodage des données utilisé. Le schéma de la figure 1 résume ces caractéristiques.

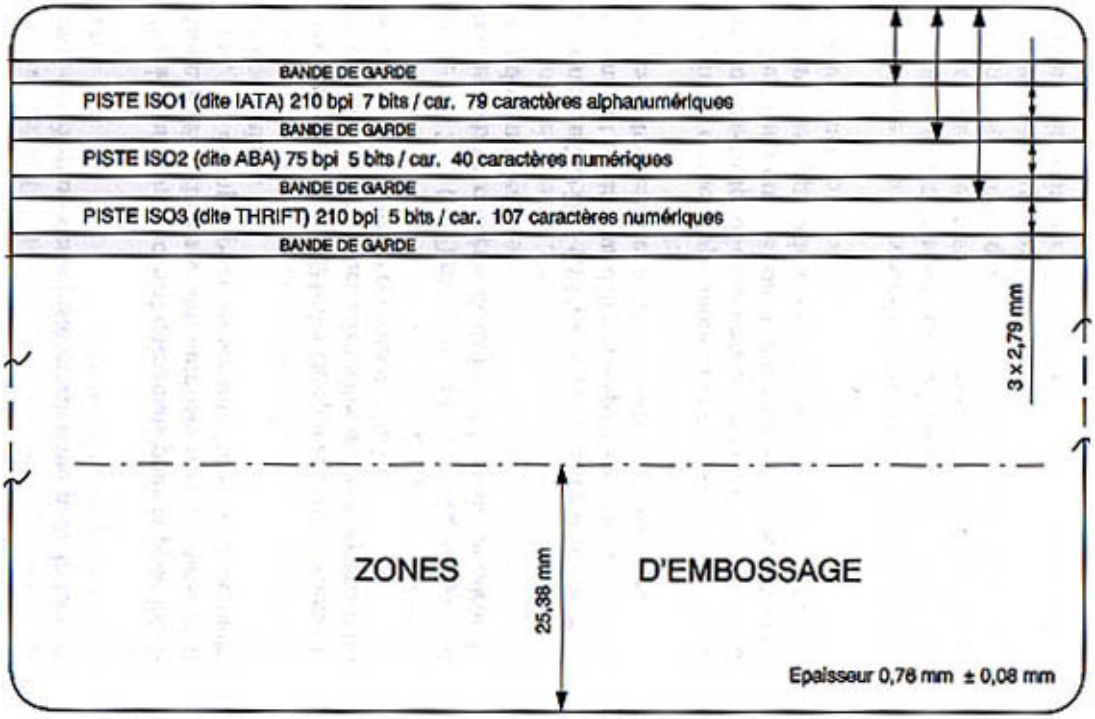


Fig. 1 : Carte ISO normalisée

La densité d'enregistrement est mesurée en bpi (bits per inch ou bits par pouce). Par exemple, la piste ISO 1 a une densité d'enregistrement de 210 bpi ce qui correspond donc à une capacité de 210 bits de données pouvant être enregistrées sur un pouce (27,07 mm) du support magnétique.

Les données enregistrées sur les pistes sont toujours encadrées par un caractère start et un caractère end. Ils vont simplement permettre la reconnaissance du type d'encodage du flot de données utilisé (sur 5 ou 7 bits). En outre, un caractère sep peut être utilisé pour séparer les différents champs de données.

La possibilité d'erreurs lors d'une opération de lecture implique l'introduction de mécanismes de contrôle d'erreurs. Tout d'abord un contrôle de parité est utilisé, cela consiste à ajouter un bit supplémentaire à chaque groupe de bits constituant un caractère (5 ou 7 bits suivant l'encodage) pour que le nombre total de bits à 1 soit toujours impair. Le deuxième mécanisme employé est le contrôle par LRC (Longitudinal Redundancy Check). Le LRC est le résultat d'un XOR (ou exclusif), glissant, sur les données. Soit l'exemple suivant :

Calcul du LRC des données suivantes : 0000 1000 0100 1100

0000 XOR 1000 = 1000

1000 XOR 0100 = 1100

1100 XOR 1100 = 0000 (LRC)

Les données protégées seront donc : 0000 1000 0100 1100 0000

Le LRC est ainsi ajouté sous la forme d'un caractère à la suite du caractère end dans le cas des cartes magnétiques normalisées.

1.42.3 La sécurité

En ce qui concerne la sécurité, il n'y a aucun moyen de sûr pour protéger physiquement les données enregistrées sur un support magnétique. En effet, la lecture et l'écriture sont entièrement libres (contrairement aux cartes à puce qui possèdent des zones où l'écriture voire la lecture est interdite). Il est donc indispensable de sécurisé l'application utilisant ce support. Concrètement, il faut crypter les données sensibles enregistrées sur le support et couplé l'utilisation de la carte avec un code secret. En outre, pour une sécurité accrue un contrôle de l'authenticité de la carte

en temps réel peut être utilisé (pour les applications fonctionnant on-line, c'est à dire avec une centralisation des traitements).

Note : cette fiche est inspirée de l'excellent ouvrage de Patrick GUEULLE "Cartes Magnétiques et PC". L'image de la figure 1 est extraite de ce même livre.

1.43 Les chevaux de Troie

1.43.1 Qu'est-ce que c'est ?

Les chevaux de Troie ("Trojan horses" ou "Trojans" en anglais) tirent leur nom de la célèbre légende mythologique. Comme dans cette dernière, ils utilisent une ruse pour agir de façon invisible, le plus souvent en se greffant sur un programme anodin.

Ils font partie des grandes menaces que l'on peut rencontrer sur le web, parmi les virus et autres vers. Pourtant, contrairement à ceux-ci, les chevaux de Troie de ne reproduisent pas (en tout cas, ce n'est pas leur objectif premier). Ce sont à la base de simples programmes destinés à être executés à l'insu de l'utilisateur.

1.43.2 Objectifs

Leur objectif est le plus souvent d'ouvrir une porte dérobée ("backdoor") sur le système cible, permettant par la suite à l'attaquant de revenir à loisir épier, collecter des données, les corrompre, contrôler voire même détruire le système. Certains chevaux de Troie sont d'ailleurs tellement évolués qu'ils sont devenus de véritables outils de prise en main et d'administration à distance.

1.43.3 Mode d'action

Leur mode opératoire est souvent le même ; ils doivent tout d'abord être introduits dans le système cible le plus discrètement possible. Les moyens sont variés et exploitent le vaste éventail des failles de sécurité, du simple économiseur d'écran piégé (envoyé par mail ou autre, du type cadeau.exe, snow.exe, etc, etc...) jusqu'à l'exploitation plus complexe d'un buffer overflow. Après leur introduction dans le système, ils se cachent dans des répertoires système ou se lient à des exécutables. Ils modifient le système d'exploitation cible (sous Windows, la base des registres) pour pouvoir démarrer en même temps que la machine. De plus, ils sont actifs en permanence (car un cheval de Troie est un véritable serveur, il reste à l'écoute des connections provenant de l'attaquant pour recevoir des instructions) mais ils restent furtifs et sont rarement détectables par l'utilisateur. Ainsi, un listing des tâches courantes ne fournira pas d'indication suffisante : soit le cheval de Troie y sera invisible, soit son nom sera tout ce qu'il y a de plus banal ("Patch.exe", ".exe", "winamp34.exe", "winrar.exe", "setup.exe", "rundlls").

1.43.4 Contre-mesures

Du fait qu'ils ne se répliquent pas (contrairement aux virus), ils ne possèdent pas de signature de réplication et ne sont donc pas détectables par les anti-virus, en tout cas à ce niveau là. De plus, les chevaux de Troie n'altèrent en général pas les données vitales de la cible (MBR...) qui sont protégées.

Par contre, comme ils restent des programmes assez répandus sur internet et qu'ils sont rarement modifiés par les apprentis hackers, il est assez facile de les détecter avec les anti-virus actuels qui connaissent très précisément leur empreinte ou leur code. Le problème est un peu plus compliqué lorsqu'il s'agit de programmes dont les sources sont disponibles librement sur internet. Il devient alors aisé de modifier le code et de le recompiler afin d'obtenir un cheval de Troie dont l'empreinte sera unique et donc inconnue des anti-virus.

Si l'on ne peut pas détecter leur présence, on peut essayer de détecter leur activité : un cheval de Troie est obligé d'ouvrir des voies d'accès pour pouvoir communiquer avec l'extérieur. Ainsi,

plusieurs ports de la machine risquent de le trahir (par exemple 12345, 31337, etc...) surtout s'ils sont habituellement inutilisés. D'autres chevaux de Troie ont détourné cette faiblesse en utilisant des ports plus communs (relatifs aux services ftp, irc...). Là encore, un utilisateur capable de voir ces ports ouverts doit se poser la question de savoir pourquoi tel service est actif.

=> Rappelons que la commande netstat permet d'obtenir de telles informations sous Linux et Windows.

Du point de vue réseau, il est également possible de détecter ce traffic (services/ports inhabituels) ou l'activité secondaire du cheval de Troie. En effet, il arrive que la cible infectée serve de point d'entrée à l'attaquant pour se propager dans tout le réseau. Pour cela, il devra effectuer différentes tâches dont certaines sont aisément détectables (scan de machines et de ports...).

Dans la majorité des cas, de telles données trahissent non seulement la présence du cheval de Troie mais fournissent également des informations sur son identité, permettant ainsi de mieux l'éradiquer. Il est même possible d'installer par la suite des leures qui garderont des traces des tentatives de connections externes (trahissant l'attaquant).

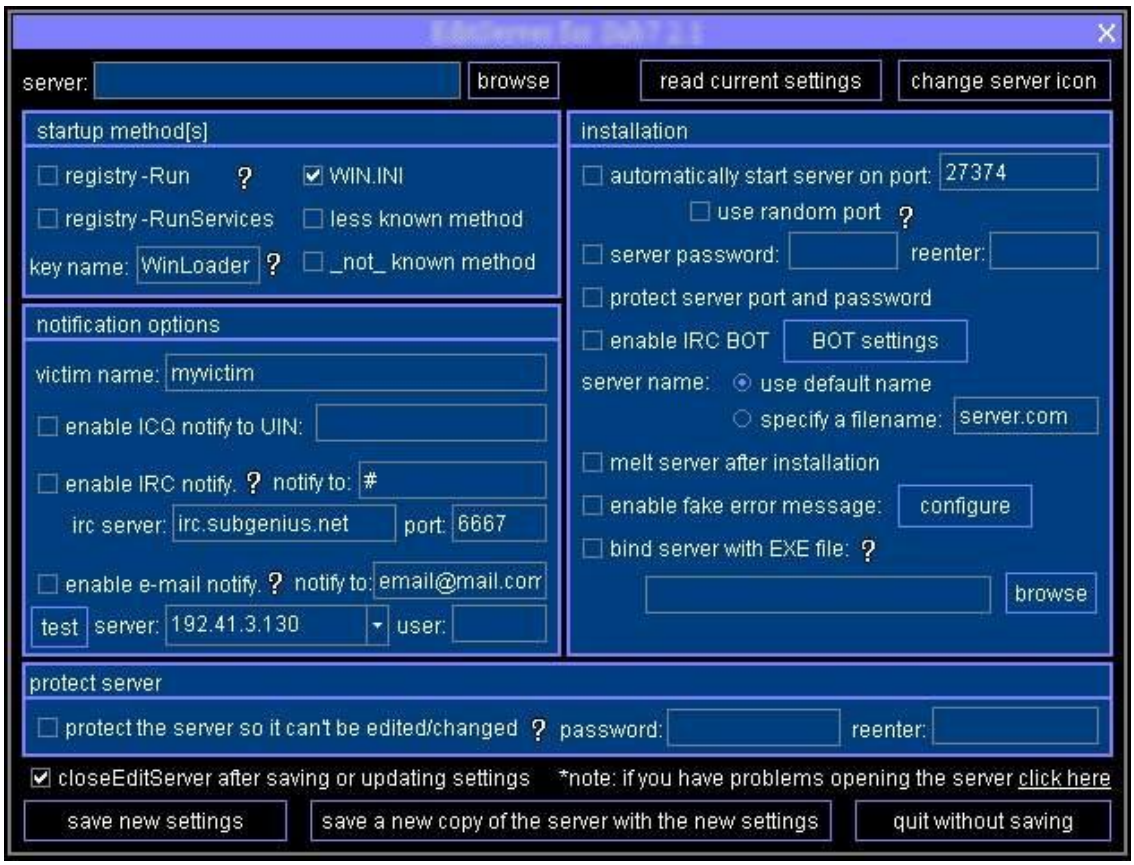
1.43.5 Cas concrets

Voici les fonctionnalités d'un des chevaux de Troie les plus répandus :

- Accès Telnet permet de lancer une application en mode texte type "Ms-Dos" ou "Invite de commande" de façon invisible et de rediriger l'entrée/sortie standard vers un port particulier. L'attaquant n'a plus qu'à s'y connecter (via telnet) pour communiquer directement avec l'application.
- Accès HTTP avec un navigateur, supporte le téléchargement et l'envoi de fichiers permet de créer un serveur web basique dont la racine est celle du disque dur (défaut). Ainsi, un simple navigateur web permet de naviguer dans l'arborescence des fichiers, d'en télécharger et même d'en rajouter.
- Information sur le système distant
- Récupère tous les mots de passe permet d'accéder aux fichiers mots de passe Windows (pwl et autres) et d'en afficher le contenu. A noter que les mots de passe utilisés pour des connections distantes, partages de documents, etc, sont également récupérés.
- Envoi de boite de dialogue (version Windows) avec réponse de l'utilisateur permet de communiquer avec l'utilisateur.
- Télécharger/Envoyer/Supprimer/Créer des fichiers permet d'accéder au système de fichiers dans sa totalité.
- Ouverture/Fermeture des fenêtres actives permet d'interagir avec le système cible.
- Accés a la base de registre
- Augmenter/Diminuer le volume sonore
- Ajouter des plugins
- Démarrage d'application
- Jouer des fichiers .wav
- Afficher des images
- Ouvrir des documents
- Imprimer
- Fonction keylogger permet d'enregistrer toute frappe au clavier pour récupération et traitement ultérieur (mots de passe sur le web, mails, etc..). Cette fonctionnalité existe également en version temps-réel : affichage des frappes clavier en direct chez l'attaquant.
- Capture d'écran permet de visualiser le poste de travail et les actions de l'utilisateur tout en économisant la bande-passante (par rapport au streaming video)
- Capture d'image si l'ordinateur est équipé d'une webcam opération basée sur l'utilisation détournée des librairies système (COM) qui supportent les webcams. Le résultat est complètement indétectable pour l'utilisateur.
- Capture du son si l'ordinateur/Serveur est équipé d'un microphone
- Eteindre l'ordinateur
- Redémarrer l'ordinateur

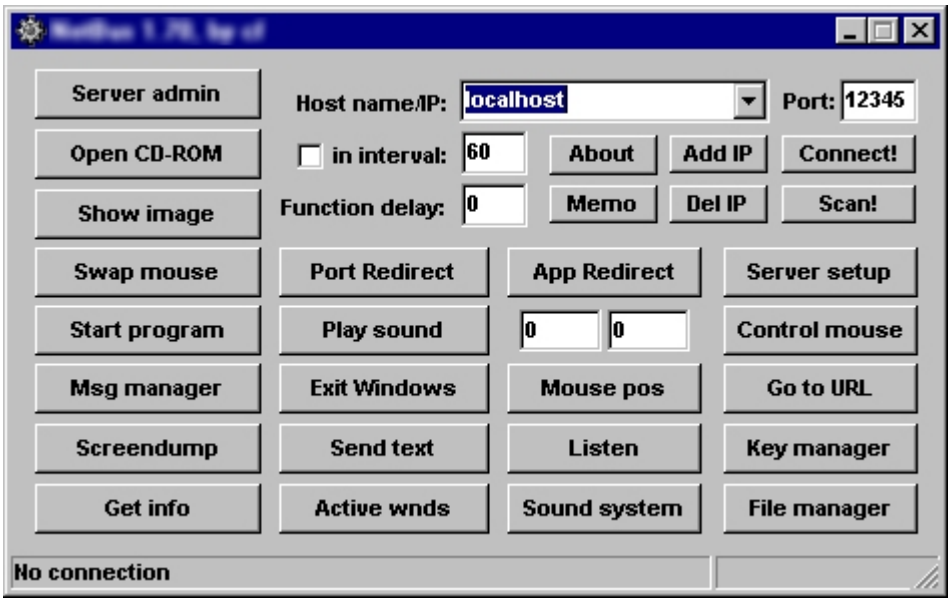
- Déconnecter l'ordinateur du réseau
- Dialogue avec l'utilisateur
- Ouverture/Fermeture du CD-ROM
- Inversion des bouton de la souris
- Envoyer l'utilisateur a une URL choisie
- Blocage du clavier

Une capture de l'interface de configuration d'un cheval de Troie :



Cette interface permet de modifier le cheval de Troie avant de l'envoyer à la cible : quel port doit-il écouter, quelle méthode de démarrage utiliser... Il est même possible de protéger l'accès au futur cheval de Troie par login/password ce qui évitera que d'autres attaquants ne s'y connectent.

Une capture du client d'un cheval de Troie :

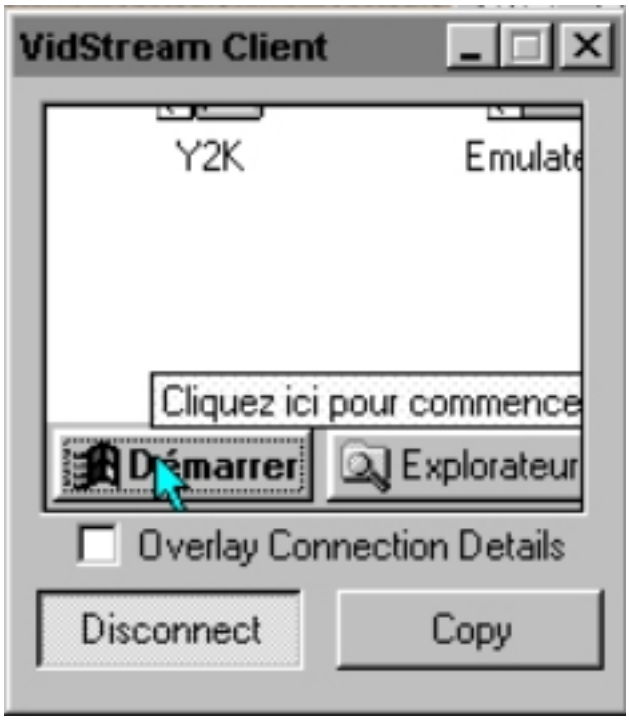


La partie cliente d'un cheval de Troie est l'application utilisée par l'attaquant pour se connecter au serveur, c'est-à-dire au programme installé sur la cible. Ce client permet d'automatiser et de simplifier nombre de tâches, et même de gérer plusieurs serveurs ! On y retrouve les fonctionnalités citées précédement (telnet, capture d'écran, etc...) et quelques autres comme la redirection de ports qui permet de récupérer tout le traffic que reçoit la cible sur des ports donnés, et donc de l'utiliser pour rebondir (le but étant de ne pas compromettre l'adresse IP de l'attaquant).

1.43.6 Back Orifice 2000

Back Orifice ("BO") est sans doute le cheval de Troie le plus connu. Il a été créé par The Cult Of The Dead Cow (cDc), un groupe de hackers formé en 1984. Sa version actuelle est la version 2000 (BO2k), et ses sources sont maintenant disponibles sur internet en license GPL. Cela a sensiblement changé son statut puisqu'il autorise tout personne à vérifier le contenu de l'application pour en être sûr (en effet, nombre de logiciels commerciaux sont accusés de receler une porte dérobée sous prétexte que leur code source n'est pas libre). Cela assure également son évolution et sa pérennité futures. Un autre point concernant BO2k est son extrême efficacité et ingéniosité. De nombreux bugs ont été corrigés par rapport aux versions précédentes et il possède un grand nombre d'extensions ou "plugins" qui lui donnent une modularité sans limites.

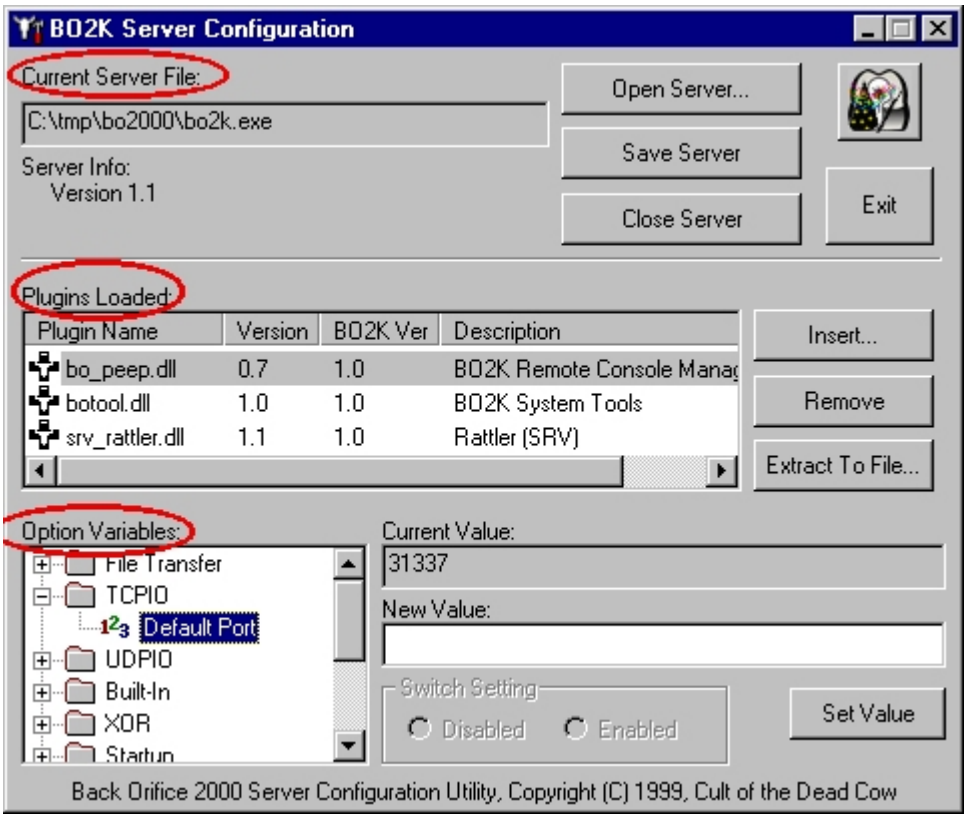
Ainsi, il est possible de visualiser en temps réel les déplacements de la souris sur la machine cible :



Tout comme il est possible de diriger cette souris et de contrôler le clavier :

Il existe également des plugins supportant le cryptage de manière à protéger les communications client-serveur ; les algorithmes supportés sont nombreux : RC6 384, IDEA 128, CAST 256 ; Back Orifice 2000 supporte même le tunneling SSH !

Regardons de plus près l'interface de configuration de BO2k :



Il y a 3 zones principales : la première où l'on spécifie le fichier du serveur que nous allons configurer (l'executable sera modifié), la seconde où nous définissons les extensions que nous allons utiliser plus tard (qui seront rajoutées à l'executable). La dernière zone concerne les paramètres de chaque fonctionnalité, y compris les extensions que nous venons d'ajouter. Ici nous pouvons voir que l'option de connexion par TCP a été choisie et que le port à utiliser est le 31337.

C'est à partir de cette interface de configuration que l'on peut modifier si l'on veut le nom du cheval de Troie. Une fois lancé, BO2k s'installe dans \Windows\System\ ou \WinNT\System32\ sous ce nom là (par défaut UMGR32.EXE). Après il modifie la base des registre.

- Sous Windows 95/98, la commande d'éxécution du serveur est écrite dans : HKEY_LOCAL_MACHINE\Softwa
- Sous Windows NT, la commande d'éxécution du serveur est écrite dans : HKEY_LOCAL_MACHINE\SOFTW

Le fichier initial peut ensuite être effacé (ou s'auto-effacer si spécifié). BO2k devient ensuite actif à chaque démarrage du système et reste en mémoire. Sous NT, le cheval de Troie utilise une astuce pour éviter d'être tué par le Gestionnaire de Tâches. Il change son PID constamment et créé des processus fils qui lui permettent de rester actif si l'un d'entre eux est tué. De plus, son nom comporte un grand nombre d'espaces et de 'e', ce qui a pour effet de renvoyer une erreur lorsqu'on tente de le tuer à partir de Windows (tout en n'affectant en rien son fonctionnement). Seule solution : le tuer à partir du DOS !

Sous Windows 9x, le fichier se renomme ".exe" (c'est-à-dire sans nom), ce qui le rend invisible dans le gestionnaire de tâches.

1.43.7 Back Orifice 2000 : fiche technique

Toutes les versions comportent au minimum :

- un client
- un serveur
- un application graphique de configuration du serveur

Le serveur ne marche que sous Windows (les dernières versions supportent Windows NT).

Le client était disponible pour Windows ou Unix dans ses versions précédentes. La version 2000 est réservée aux plateformes Win32.

Le serveur est totalement configurable (numero de port, type de liaison TCP/UDP...)

Les fonctionnalités disponibles (de base) comprennent :

- Liste de serveurs (style Address Book)
- Extensibilité via plugins
- Connections serveurs multiples (concurrentes possibles)
- Connections de plusieurs clients possible Journalisation de sessions
- Journalisation de frappes clavier
- Supporte HTTP pour navigation dans le système de fichiers (chargements possibles)
- Gestion du partage de fichiers Microsoft (ajout/suppression de partages, monitoring)
- Gestion directe de la Base de Registres
- Navigation directe dans le système de fichiers (transferts TCP, gestion...)
- Mises à jour à distance, ainsi que installation/désinstallation
- Redirection de connections TCP/IP
- Redirection d'applications texte pour accès via Telnet
- Support multimedia, capture audio/video, lecture audio
- Récupération de mots de passe stockés dans la SAM (NT registry) et économiseurs d'écran sous Win9x
- Contrôle/arrêt/lancement/listing des processus
- Affichage de message à l'écran
- Compression de fichiers propriétaire
- Redémarrage de la machine à distance
- Locking de la machine à distance
- Récupération d'informations système
- Résolution de noms DNS

1.43.8 Conclusion

Les chevaux de Troie représentent aujourd'hui un phénomène inquiétant car grandissant. Ils ont changé les règles du jeu, ouvert de nouvelles voies dans lesquelles se retrouvent de plus en plus d'apprentis hackers. Car contrairement aux virus, ils sont faciles à utiliser, accessibles à tous (sans pré-requis en programmation) et très efficaces. En témoigne leur utilisation croisante à des fins professionnelles : prise en main à distance (help desk, etc...), administration centralisée, gestion de parcs informatiques. Bien sûr, la majorité des produits utilisés dans ce secteur restent des produits commerciaux, mais la récente percée de Back Orifice 2000 démontre -s'il en était encore besoin- que les choses changent. Loin d'en promouvoir l'utilisation, rappelons enfin l'énorme menace que les chevaux de Troie représentent : ces outils sont conçus pour espionner et infiltrer les systèmes. Ils sont furtifs et très difficiles à détecter, surtout tant que l'attaquant ne cherche pas à se manisfester. Et cette efficacité ne se limite pas qu'à leur action ; il ne faut pas négliger l'impact médiatique entrainé par la découverte d'un tel programme dans le système d'une entreprise : compromission, espionnage industriel, remise en cause de la politique de sécurité, etc.

1.44 Les Key Loggers

1.44.1 Qu'est ce que c'est ?

Les Key Loggers sont des enregistreurs de touches et par extension des enregistreurs d'activités informatiques permettant d'enregistrer les touches utilisées par un utilisateur sur son clavier et tous les évènements déclenchés.

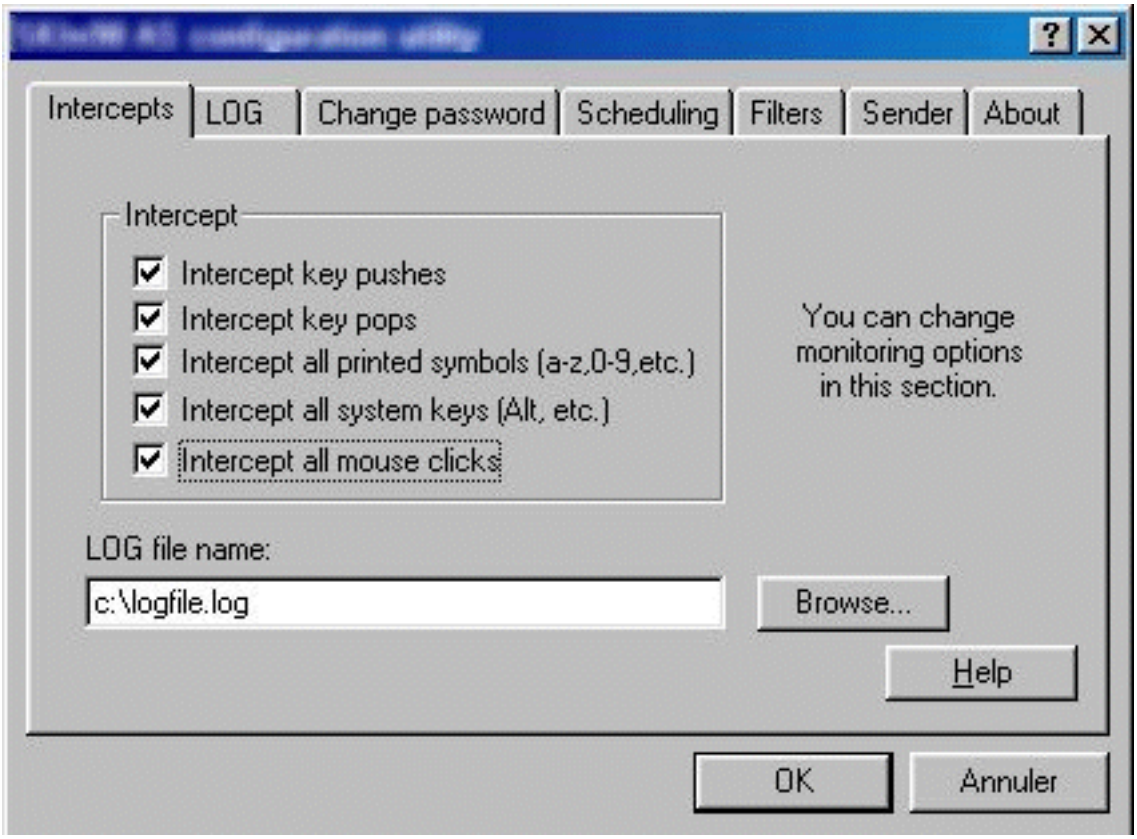
Dangereux, ils ne sont pourtant pas répertoriés parmis les virus, vers, ou chevaux de Troie car n'ont pas pour objectif de modifier quoi que se soit dans la machine cible et permettent simplement l'enregistrement d'informations.

1.44.2 Objectif

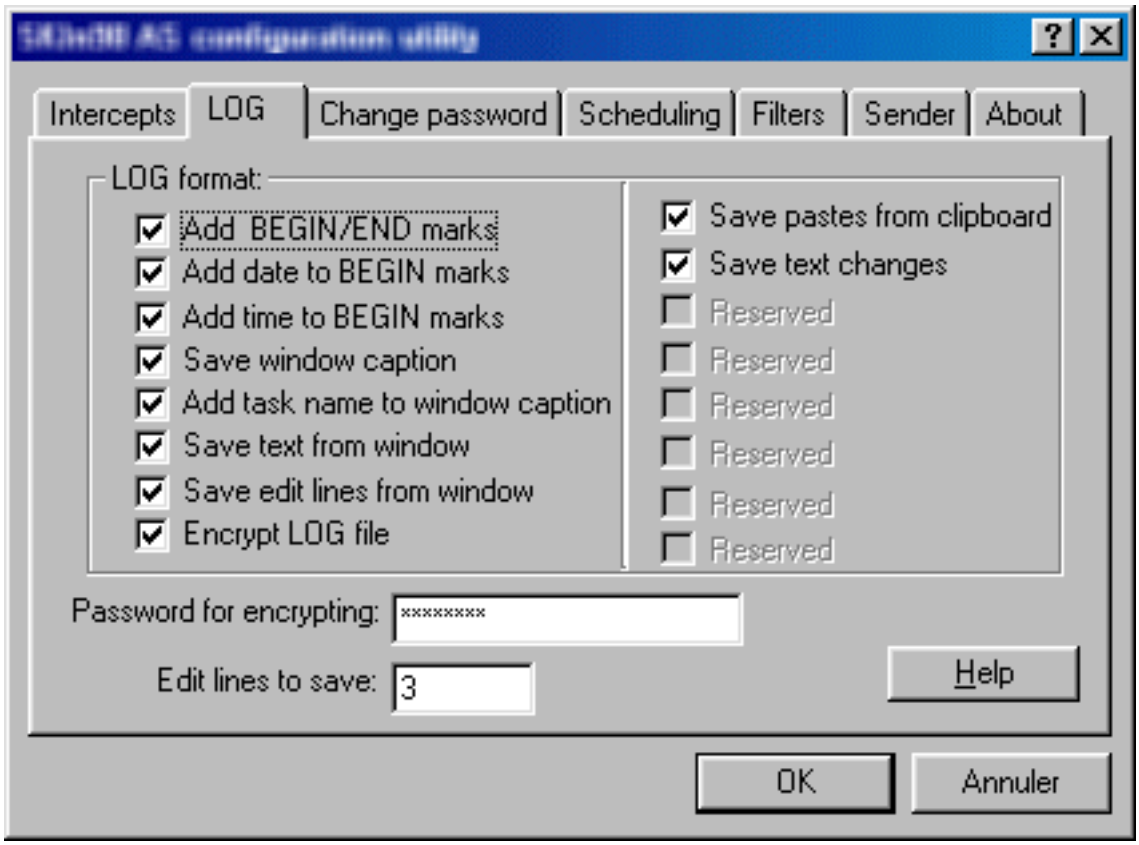
L'objectif des key loggers est d'enregistrer et de restituer tout le travail qui a été réalisé par un utilisateur. Les touches enregistrées permettent effectivement de retracer non seulement le travail courant, mais aussi de récupérer tous les identifiants et mots de passes.

1.44.3 Mode d'action

Le mode opératoire des Key Loggers est identique, même s'il existe une multitude de Key Loggers différents. Ils sont installés directement par le pirate sur la machine visée, si l'ordinateur n'a pas de connexion internet permettant une installation à distance via un cheval de Troie. En général, les Key Loggers se lancent directement au démarrage de la machine hôte. Une fois le key loggers lancé, il enregistre au fur et à mesure tout ce qui est réalisé. Dans la plupart des cas, si la machine cible est pourvue d'une connexion internet, le key logger enverra discrètement, à une adresse mail ou à un serveur internet, un fichier, généralement crypté, contenant tous les renseignements collectés. En fonction du Key Logger sélectionné différents types d'écran de configuration existent.



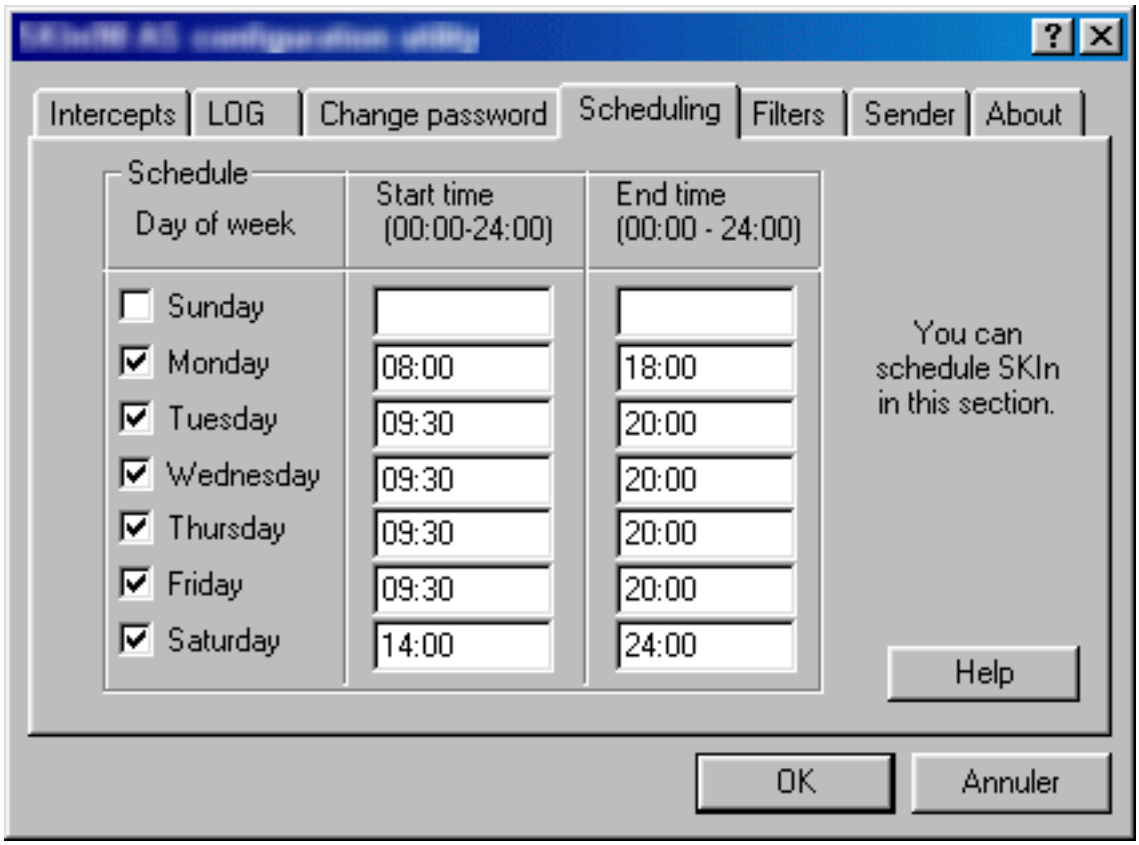
Dans cet écran, vous pouvez constater qu'en fonction des informations qui intéressent le pirate, il est possible de choisir quel type de touches seront enregistrées dans le fichier log. Ici, nous avons choisi d'enregistrer toutes les touches du clavier pour vous donner une copie écran plus explicite de ce qu'enregistrera le fichier de traces.



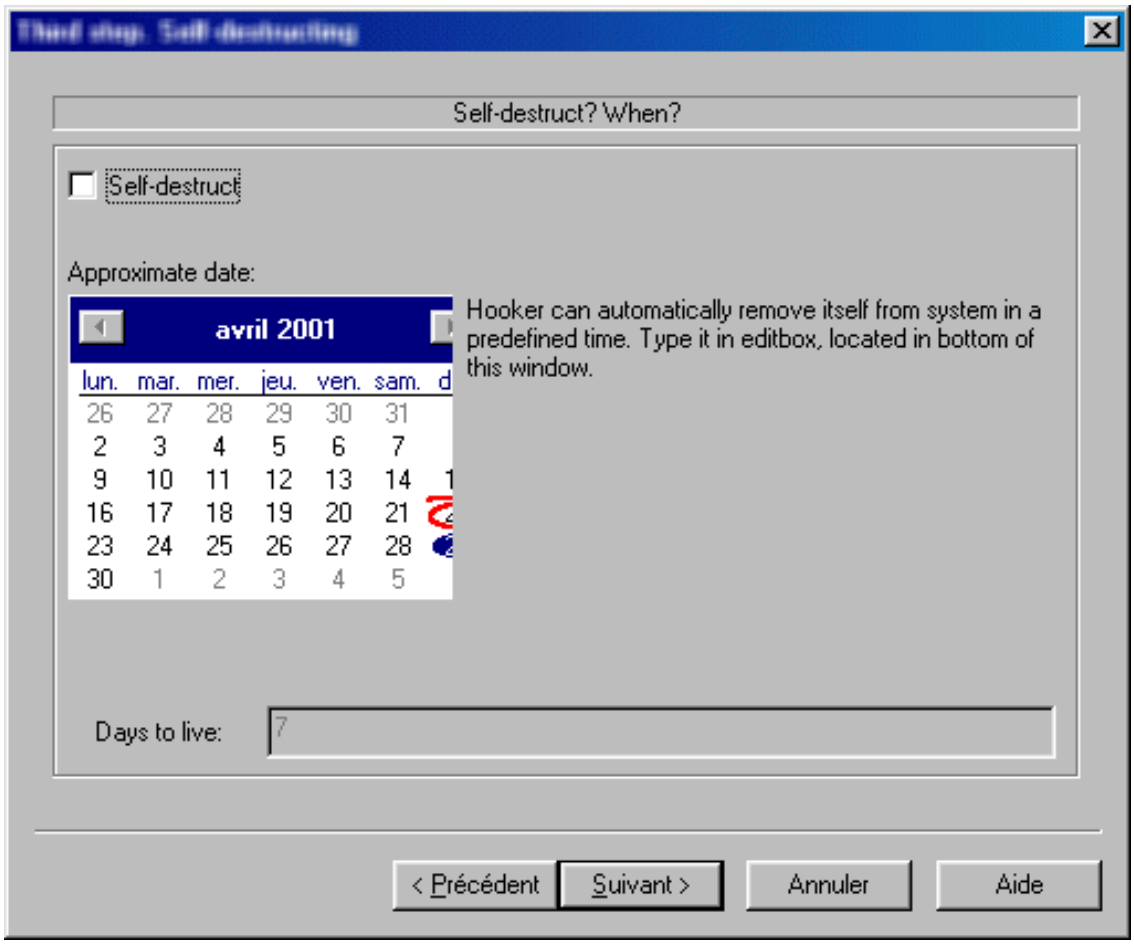
En complément des touches enregistrées vous pouvez constater ici que des éléments importants comme la date, les applications ouvertes et le choix ou non du cryptage du fichier de trace sont possibles. Le password réclamé par le key logger permet d'assurer au pirate que lui seul pourra décrypter le fichier. Même si un utilisateur découvrait un fichier crypté il ne saurait reconnaître les éléments contenus à l'intérieur. En effet pas évident de comprendre l'extrait crypté d'un fichier log comme :

#qblo"(qgm|m5qsgo[[bss@@bb-|sbo6)iqjbap3vgl''¿±ß±jt@@ß@æ6687

L'option de planification de l'activité du Key Logger peut être très utile au pirate. En effet, il peut planifier les jours et moments auxquels le Key Logger doit se mettre en fonction. Cela permet de n'avoir que les informations désirées et favorise une plus grande discrétion puisque la mémoire dans laquelle le Key Logger s'intalle généralement n'est sollicitée qu'à des moments bien précis.

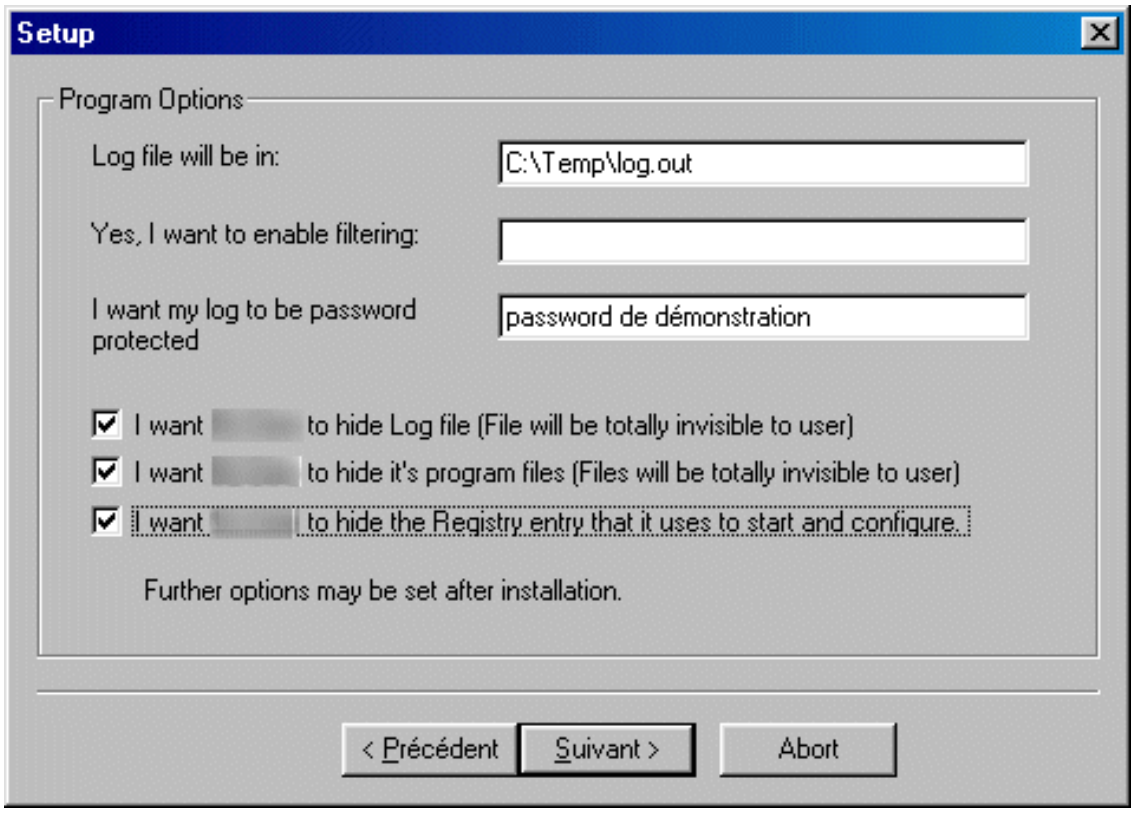


Selon le Key Logger choisi, il est possible de paramétrer l'option "auto-destruction". Dès lors impossible à l'utilisateur ou à l'administrateur du parc informatique de remonter au programme. Il suffit de déterminer la plage en nombre de jours pendant laquelle le Key Logger doit rester actif sur la machine cible.



1.44.4 Contre-mesures

Les Key Loggers ne sont pas toujours identifés par les anti-virus. Il n'est donc pas évident de les remarquer. En outre, dans la plupart des cas des options permettant l'invisibilité du programme exécuté existent.



Par contre, les Key Loggers s'éxécutent au démarrage de la machine. Tout ralentissement du système au lancement doit sembler suspect. Cependant, avec les nouvelles générations d'ordinateurs il est de moins en moins simple de noter ces ralentissements machines.

En général les fichiers de récupération, cryptés ou non sont stockés avec des noms très peu parlant dans c :/windows/temp. Il est intéressant d'aller tenter d'ouvrir les fichiers contenus dans ce répertoire. Pour ouvrir ces fichiers, cliquer en même temps sur "Shift" et le bouton droit de la souris. Parmi le menu qui s'offre vous verrez l'option "ouvrir avec". Le plus simple est alors de choisir "Note Pad" qui affichera les éléments en mode texte seulement. Vous pouvez, si le fichier n'est pas ou mal crypté retrouver des éléments qui doivent immédiatement vous alerter.

Dans le cas ou vous trouvriez un fichier suspect le plus simple est de commencer par faire travailler votre machine uniquement en local. Déconnectez vous de votre réseau et stoppez toute connexion internet. Cela empèchera aux fichiers de parvenir au pirate. Prévenez votre adminstrateur qui recherchera via le serveur les échanges de mail et tentera de retrouver l'adresse du destinaire.

Une inspection des tâches qui sont en train d'être exécutées par votre ordinateur s'impose. En effet un simple "Ctrl" "Alt" "Suppr" n'affichera pas les Key Loggers alors qu'un programme comme Procdump vous les signalera. En parallèle, recherchez sur votre pc tous les fichiers créés le jour ou vous décrouvrez ce soucis. Dans le pire des cas, il sera peut être nécessaire que vous sauvegardiez tous vos fichiers de données pour ensuite re formater votre disque dur.

1.44.5 Conclusion

Les Key Loggers sont des outils particulièrement utiles pour les pirates, puisqu'ils permettent en autre de récupérer comme nous l'avons dit les mots de passe et les loggins des utilsateurs. Cependant, ils peuvent aussi être un outil de "surveillance" pour les entreprises. En interne, cela pourrait permettre de vérifier les activités réalisées par les salariés pendant les heures de bureau. Cependant, l'utilisation d'un Key Logger ne saurait être utile sans consentement préalable du salarié. Sur un poste non connecté à internet, l'installation d'un tel outil implique le passage du

pirate sur la machine cible. Le meilleur moyen de prévention reste donc la vigilance, ne pas quitter son poste sans avoir au minimum vérouiller son écran, et bien ne pas diffuser son mot de passe de session à quiconque.

1.45 Les espiogiciels

A chaque connexion internet, un utilisateur laisse derrière lui très grand nombre d'informations. Ces traces sont généralement intéressantes mais non suffisantes à un public de professionnels ou d'espions cherchant à obtenir d'autres éléments que ceux techniques laissés en standard. Les professionnels d'un secteur déterminé cherchent à connaître les habitudes de téléchargement de leurs clients, leurs modes de consommations, leurs centres d'intérêts, ou la périodicité de leurs achats par exemple. Les pirates ou espions seront, eux, plus intéressés par le contenu des machines connectées, la réception de ces informations etc ....

1.45.1 Qu'est-ce qu'un espiogiciel ?

Pour faciliter la récolte de ce type de renseignements, il existe des "espiogiciels", en anglais Spywares. Ils se trouvent généralement dans le code d'un programme que l'utilisateur téléchargera innocemment sur internet. Dans la plupart des cas, ces espogiciels sont des "petits morceaux de codes parasite" (routines) intégrés dans le code principal du programme. Dans un même programme, il peut y avoir plusieurs routines parasites différentes, ayant chacune une fonction déterminée. Dans le cas d'un logiciel de messagerie par exemple, il est possible de trouver une routine faisant qu'une copie de chaque email sera envoyée à une adresse déterminée sans laisser de trace dans la boîte éléments envoyés de l'email dupliqué.

La détection de ces routines est très difficile. En effet, plus le logiciel initialement téléchargé est volumineux, plus les chances de trouver les routines éventuelles seront faibles. Il est impossible par exemple à un développeur seul, ou à une équipe d'analyser le code source d'un navigateur internet. Sans vouloir sembler paranoïaque, il est donc important de garder en mémoire que tout exécutable est potentiellement infecté d'un espiogiciel. En outre, dans certains pays, il n'est pas toujours légal de désassembler un logiciel, (rendre le code source du programme lisible et donc modifiable). Impossible donc en respectant la loi de valider l'intégrité des programmes utilisés.

Dans tous les cas, l'espiogiciel aura besoin d'une connexion internet pour la transmission des données. C'est pourquoi ces routines se trouvent majoritairement dans des exécutables prévus pour fonctionner avec internet (logiciels de téléchargement de MP3, films, traducteurs, browsers etc...). Généralement les logiciels libres (freewares) et logiciels d'évaluation (sharewares) sont les principaux vecteurs d'espigociels.

Un outil infecté par un spyware peut représenter une très grande menace pour la sécurité du système d'information infecté. En effet plusieurs routines successives peuvent permettre la détection de mots de passe encrypté et le crackage de ces informations. Il suffit pour cela d'indiquer dans une routine à l'ordinateur de mettre à profit le temps CPU disponible pour cracker le mot de passe à l'insu de l'utilisateur.

1.45.2 Comment s'en protéger ?

Se protéger des spywares n'est pas chose facile. En effet, un anti virus ne les détectera pas puisqu'il ne détaille pas l'ensemble du code des programmes mais reconnaît des signatures au préalable identifiées. De plus, un espiogiciel n'est pas un virus. Les éditeurs d'antivirus ne travaille donc pas sur ce "marché".

En outre, l'utilisation d'un firewall ne permettra généralement pas non plus la détection des espiogiciels. En effet, même si la routine provoque l'envoi d'un fichier par email à un destinataire non désiré la configuration du firewall, sauf exception, n'a pas pour but d'analyser ce qui sort du PC mais à l'inverse ce qu'il rentre. Le firewall n'a donc pas de moyen de savoir qu'un email est émis volontairement ou à l'insu de l'utilisateur. De plus, un firewall ne s'intéresse pas à la nature

des fichiers qui transitent mais aux paquets qui voyagent sur le réseau. Il n'y a donc pas de moyen simple pour le firewall d'identifier comme des menaces l'exécution des routines et la passation d'informations. Cependant, l'un des moyens existants pour suspecter un spyware sur une machine est de voir un flux de paquets nettement supérieur aux flux habituel passer via le firewall ou le modem. Mais là encore, c'est très difficile à détecter.

Il existe sur le net de nombreux sites référençant des spywares. Cependant aucun ne peut prétendre avoir une liste exhaustive des espiogiciels existants. De même certains outils permettent la détection de logiciels identifiés comme ayant des spywares mais les utiliser ne garantit pas une sécurisation à 100% du PC.

En conclusion, il est impossible à l'heure actuelle de surfer en étant certain que nos informations ne sont pas transmises. Il n'existe aucun moyen de s'assurer qu'un ordinateur connecté à internet ne soit pas à même d'envoyer à notre insu des éléments non désirés. C'est pourquoi il est parfois préférable d'utiliser un P.C. public, (cybercafé, université etc ...) si l'on veut surfer en paix sans donner d'informations. Une autre solution peut être dans le cadre d'un réseau d'avoir une machine par service internet utilisé. C'est à dire par exemple un PC dédié à la messagerie, l'autre pour le surf et un troisième pour l'utilisation des services FTP par exemple. Les risques sont toujours présents mais limités à chaque fois aux informations disponibles sur une machine seulement.

1.46 La stéganographie

1.46.1 Qu'est-ce que c'est ?

La stéganographie, c'est l'art de dissimuler des données dans d'autres données. Il existe plusieurs techniques différentes qui permettent ce "tour de magie". La stéganographie existe depuis longtemps, bien avant l'invention de l'ordinateur. En voilà un exemple frappant : Ceci est une lettre envoyée par George Sand à Alfred de Musset :

> Je suis très émue de vous dire que j'ai
> bien compris, l'autre jour, que vous avez
> toujours une envie folle de me faire
> danser. Je garde un souvenir de votre
> baiser et je voudrais que ce soit
> là une preuve que je puisse être aimée
> par vous. Je suis prête à vous montrer mon
> Affection toute désintéressée et sans cal-
> cul. Si vous voulez me voir ainsi
> dévoiler, sans aucun artifice mon âme
> toute nue, daignez donc me faire une visite
> Et nous causerons en amis et en chemin.
> Je vous prouverai que je suis la femme
> sincère capable de vous offrir l'affection
> la plus profonde et la plus étroite
> Amitié, en un mot, la meilleure amie
> que vous puissiez rêver. Puisque votre
> âme est libre, alors que l'abandon où je
> vis est bien long, bien dur et bien souvent
> pénible, ami très cher, j'ai le coeur
> gros, accourez vite et venez me le
> fait oublier. À l'amour, je veux me sou-
> mettre.

A priori, si vous n'avez pas compris que cette lettre en cache une autre, c'est beau, plein de poésie... Maintenant, lisez la première ligne et ensuite une ligne sur deux...

> Alfred de Musset a répondu ceci :
> Quand je vous jure, hélàs, un éternel hommage
> Voulez-vous qu'un instant je change de language
> Que ne puis-je, avec vous, goûter le vrai bonheur
> Je vous aime, ô ma belle, et ma plume en délire
> Couche sur le papier ce que je n'ose dire
> Avec soin, de mes vers, lisez le premier mot
> Vous saurez quel remède apporter à mes maux.

De la même manière George Sand a répondu ceci :

> Cette grande faveur que votre ardeur réclame
> Nuit peut-être à l'honneur mais répond à ma flamme.

Maintenant, je pense que vous avez compris ce qu'est la stéganographie...

Avec l'avènement de l'ordinateur et de son règne du "tout numérique", des techniques existent pour cacher n'importe quel document dans un autre document.

1.46.2 Comment est-ce possible ?

Les documents "porteurs" sont généralement des images (BMP,GIF...) ou des sons (WAV...). Nous allons voir comment il est possible de cacher un document dans une image RVB. Pour cela nous allons faire un bref rappel de ce qu'est une structure d'image.

Le pixel :

Un pixel est constinué de 3 octets : un octet pour la composante rouge, un octet pour la composante verte et un octet pour la composante bleue. C'est pour cela que l'on parle de RVB (Rouge Vert Bleu). A partir de ces trois octets, on peut donc avoir 256*256*256 = 16777216 couleurs différentes, ce qui est largement plus que ne peut distinguer l'oeil humain.

L'image :

L'image n'est ni plus ni moins que le stockage dans un fichier de tous les pixels RVB composant l'image finale.

Cacher le document :

L'astuce est de retirer un bit à chaque octet RVB qui compose chaque pixel de l'image. En effet, en retirant 1 bit, on dégrade l'image, mais ce n'est pas visible à l'oeil nu... De ce fait, on peut récupérer ce bit à chaque fois et l'utiliser pour stocker les données que l'on souhaite. Nous récupérons donc 1/8e de la taille de l'image pour cacher un document, quel qu'il soit.

1.47 Social Engineering

1.47.1 Qu'est-ce que c'est ?

> "Illusion is everything" - Bernz

C'est une technique qui a pour but d'extirper des informations à des personnes. Contrairement aux autres attaques, elle ne nécessite pas de logiciel. La seule force de persuasion est la clé de voûte de cette attaque. Il y a quatre grandes méthodes de social engineering : par téléphone, par lettre, par internet et par contact direct.

1.47.2 Par téléphone

Le hacker vous contactera par téléphone. C'est la technique la plus facile. Son but est d'avoir le renseignement le plus rapidement possible. Un bon hacker aura préparé son personnage et son discours. Il sera sûr de lui. Il sera très persuasif dans le timbre de sa voix. Certains hackers ont quelques techniques pour parfaire leur crédibilité, comme jouer sur un magnétophone une cassette préalablement enregistrée de bruits de bureau, ou encore utiliser un matériel qui change le timbre de la voix pour imiter celle d'une secrétaire.

1.47.3 Comment parer cette méthode ?

Si vous recevez un coup de fil d'une personne que vous ne connaissez pas : Ne donnez aucun renseignement. Restez dans le vague, et débarrassez vous de lui : soit vous mettez un terme à cet appel, soit demandez une confirmation par écrit (par fax) de la demande. Par fax, on obtient le numéro appelant, et il est donc facile de l'identifier. Et ainsi, on garde une trace écrite, cela pouvant être d'une grande importance pour déposer une plainte. Malheureusement, un bon hacker vous aura étudié avant, et se fera passer pour un client, un fournisseur, ou un de vos collègue situé dans un autre bureau dans le cas d'une grande entreprise. Pire encore, il attaquera au moment le plus propice pour lui : par exemple, lorsque le responsable d'un client est en vacances. Il devient très dur alors, de se douter d'une mauvaise intention... Attention aux entreprises qui externalisent leur sécurité informatique...

1.47.4 Par lettre

Le hacker vous fera une lettre très professionnelle. Au besoin, il n'hésitera pas à voir un imprimeur pour avoir du papier à lettre comporant un logo, un filigramme, téléphone, fax, email... Il utilisera très certainement une boîte postale pour l'adresse de sa société fictive.

1.47.5 Comment parer cette méthode ?

L'idéal serait de filtrer tout le courrier entrant de l'entreprise. Pour chaque source inconnue de l'entreprise, il faudrait faire une vérification de l'existence réelle de celle-ci.

1.47.6 Par internet

Le social engineering par internet est semblable à celui par téléphone. Le hacker se fera facilement passer pour un opérateur système, un responsable informatique ou un ingénieur système.

1.47.7 Comment parer cette méthode ?

Comme pour le téléphone, ne donnez pas de renseignements à quelqu'un que vous ne connaissez pas. Mais par internet, c'est plus facile de donner de la crédibilité, tant il y a de noms de domaines et d'adresses emails farfelus. Il n'est donc pas facile de faire la part des choses. Une bonne étude de la gestion de l'extranet et de la mise en place d'une structure matérielle et personnelle adéquate est la meilleure solution.

1.47.8 Par contact direct

C'est le social engineering le plus dur de la part du hacker. Il sera équipé pour que vous n'y voyez que du feu : costard, cravate, très classe, très propre, attaché-case , agenda rempli, documents divers, carte de visite, badge... Si le hacker prend de tels risques, c'est qu'il est déterminé à obtenir les renseignements souhaités. Il sera donc très persuasif.

1.47.9 Comment parer cette méthode ?

Cela est très difficile, car vous avez été directement confronté au charisme du hacker. S'il a réussi, vous êtes persuadé de son honnêteté. Cependant, lors d'une discussion, n'hésitez pas à demander un maximum de renseignements "concrets" (nom de votre interlocuteur, nom et adresse de la société, etc), pour, par la suite, vérifier auprès des organismes compétants l'existenceréelle de votre interlocuteur. N'hésitez pas à téléphoner à la société pour savoir si la personne existe, et si elle est au courant qu'elle vous a vu ces dernieres heures...

1.47.10 Conclusion

Il ne faut pas perdre de vue que le hacker ne se limitera pas à une seule de ces techniques, mais au contraire, utilisera une combinaison de ces quatre méthodes. Par exemple : téléphoner pour obtenir un rendez-vous, confirmer par écrit, et passer une heure en votre compagnie... Il est important, de la part d'une entreprise, de former le personnel à ce problème. Un bon hacker s'attaquera à la personne la plus faible de l'entreprise, à savoir le personnel non technique (secrétaires, comptables...) et les personnes récemment recrutées. La paranoïa reste l'arme ultime de ce genre d'attaque.

1.48 Les vers informatiques

1.48.1 Qu'est-ce que c'est ?

Un vers est un programme parasite. Il n'est pas forcément autopropageable. Son but est de grignoter des ressources système : CPU, mémoire, espace disque, bande passante... Ces petits bouts de programme sont dépendants du système d'exploitation ou d'un logiciel. Ils se progagent, comme toutes données binaires, par disquettes, CD ROM, réseaux (LAN ou WAN)... Depuis la démocratisation des virus (due notamment à la prolifération des générateurs de virus), le nombre de nouveaux vers est en net recul. Cependant, il en existe toujours. Pour les prévenir, on utilise très généralement la même stratégie que celle qui est adoptée contre les virus.

1.48.2 Qui peut provoquer cette attaque ?

N'importe qui, volontairement ou involontairement. Volontairement si la personne a au moins un ver en réserve, ou un logiciel permettant de générer des vers. Involontairement si la personne rapatrie d'internet des archives non sûres ou non certifiées par l'entreprise (99% des cas).

1.48.3 Conséquences

Les conséquences sont multiples :
- Ralentissement système.
- Blocage système.
- Crash système.
- Pertes de données.
- Etc.

1.48.4 Comment s'en protéger ?

Utilisation d'un firewall pour filtrer ce qui vient d'extranet. Utilisation d'au moins un antivirus, remis à jour très régulièrement et exécuté régulièrement.

1.49 Les virus informatiques

1.49.1 Qu'est-ce que c'est ?

Un virus est un programme parasite autopropageable. Ces petits bouts de programme sont dépendants du système d'exploitation ou d'un logiciel. Ils se progagent, comme toutes données binaires, par disquettes, CD ROM, réseaux (LAN ou WAN)... Leur nombre est sans cesse croissant depuis leur création. Dans le cadre de la sécurité informatique, les virus sont des logiciels à part. Cependant ils sont rarement propagés volontairement. Il s'agit souvent d'une négligence de la part de l'utilisateur.

1.49.2 Qui peut provoquer cette attaque ?

N'importe qui, volontairement ou non. Volontairement si la personne a au moins un virus en réserve, ou un logiciel permettant de générer des virus. Involontairement si la personne rapatrie d'internet des archives non sûres ou non certifiées par l'entreprise (99% des cas).

1.49.3 Conséquences

Les conséquences sont multiples :
- Ralentissement système.
- Blocage système.
- Crash système.
- Pertes de données.
- Etc.

1.49.4 Comment s'en protéger ?

Utilisation d'un firewall pour filtrer ce qui vient d'extranet. Utilisation d'au moins un antivirus, remis à jour très régulièrement et exécuté régulièrement.

Chapitre 2

Cryptographie

2.1 La cryptographie

2.1.1 Qu'est-ce que c'est ?

La cryptographie est une science permettant de convertir des informations "en clair" en informations codées, c'est à dire non compréhensible, puis, à partir de ces informations codées, de restituer les informations originales.

2.1.2 La cryptographie symétrique et la cryptographie asymétrique

La cryptographie symétrique

On parle de cryptographie symétrique lorsque plusieurs personnes utilisent une même clé pour crypter et décrypter des messages. Le principal inconvénient de ce système est le partage de cette clé unique entre les différentes personnes : Comment envoyer à tout le monde et de façon sécurisée cette clé unique qui permet de crypter et décrypter ?

La cryptographie asymétrique

Dans ce type de cryptographie, chaque utilisateur comporte deux clés :
- Une clé privée qui doit être gardée secrète.
- Une clé publique qui est disponible pour tous les autres utilisateurs.

Ces deux clés sont mathématiquement liées. Dans la pratique, la clé publique sert à crypter les messages, et la clé privée sert à les décrypter. Une fois le message crypté, seul le destinataire est en mesure de le décrypter. L'utilitaire PGP (Prety Good Privacy) fonctionne de cette manière.

2.1.3 L'intégrité des informations

Une bonne cryptographie doit pouvoir offrir une garantie de l'intégrité des informations. En effet, il ne doit pas être possible de pouvoir modifier des informations cryptées de façon totalement transparente. Un processus de vérification de l'intégrité du message (crypté et en clair) doit être mis en place. Ce processus est réalisé par une fonction de hachage. Le résultat d'un hachage (hash en anglais) est une sorte de condensé du message original.

2.1.4 L'authentification des correspondants

Un aspect à ne pas négliger lorsque l'on désire faire des transactions sécurisées est l'authentification des correspondants : La personne à qui j'envoie un message crypté est-elle bien celle à laquelle je pense ? La personne qui m'envoie un message crypté est-elle bien celle à qui je pense ?

Le principe de l'authentification met en oeuvre un prouveur (celui qui prétend être, qui s'est identifié) et un vérifieur (le fournisseur du service) : le vérifieur soumet un challenge au prouveur que ce dernier doit réaliser. Cela suppose qu'au préalable prouveur et vérifieur se sont entendus sur le partage d'un secret.

La signature digitale

C'est un code électronique unique qui permet de signer un message codé. Cette signature permet d'identifier l'origine du message : elle a la même fonction qu'une signature "à la main". C'est la clé privée qui permet de signer, et la clé publique qui permet de vérifier cette signature.

Le certificat digital

C'est un document électronique qui fait correspondre une clé avec une entité (personne, entreprise, ordinateur...). Cette correspondance est validée par une autorité de certification (Certificate Authority : CA). Ces certificats sont utilisés pour identifier une entité. Ce certificat est normalisé (norme X.509v3). Concrètement, les données utilisateur (identité du propriétaire de la clé, la clé publique et l'usage de la clé) sont elles même signées par l'autorité de certification, en y incluant certaines données propres (période de validité du certificat, l'algorithme de cryptage utilisé, numéro de série, etc...).

L'autorité d'enregistrement

C'est un organisme qui génère les demandes de certification d'un utilisateur. L'enregistrement de cet utilisateur n'est validé qu'après vérification des informations concernant cet utilisateur. La demande est ensuite envoyée à l'autorité de certification.

L'autorité de certification

C'est un organisme qui génère les certificats des différents utilisateurs. C'est un passage obligé pour la mise en place d'un système sécurisé (e-commerce...).

2.1.5 PKI

PKI signifie "Public Key Infrastructure", c'est à dire "Infrastructure de Gestion de Clés" (IGC). C'est un ensemble d'outils (logiciels et matériels) qui gèrent les clés cryptographiques et les certificats. L'IGC permet les transactions sécurisées et les échanges d'informations entre deux parties en garantissant le secret, l'intégrité et l'authentification. On y retrouve : La gestion des clés (création, distribution, stockage...). Association de la clé publique et de l'entité (certificat). Recouvrement de clé.

2.1.6 SPKI

SPKI signifie "Simple Public Key Infrastructure", c'est à dire "Infrastructure à Clés Publiques Simplifiée" (ICPS). Cette infrastructure permet une utilisation plus directe de l'autorisation. En effet, sous IGC, une autorisation se déroule de la manière suivante : De la clé, on obtient une identification via un certificat au format X.509. De cette identité, on obtient, ou non, l'autorisation. Sous ICPS, l'autorisation est donnée, ou non, à partir de la clé elle même.

2.1.7 L'aspect légal

En France, depuis les décrets du 19 mars 1999, il est possible d'utiliser : Une clé de 40 bits, en totale liberté quelque soit l'usage. Une clé de 128 bits en totale liberté pour un usage privé, et soumise à déclaration dans les autres cas.

2.2 La cryptographie à algorithmes symétriques

2.2.1 Qu'est-ce que c'est ?

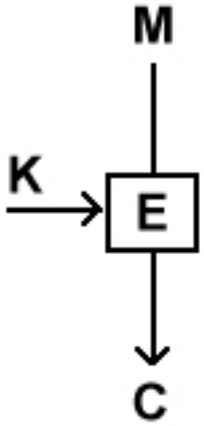
La cryptographie à algorithmes symétriques utilise la même clé pour les processus de codage et de décodage ; cette clé est le plus souvent appelée "secrète" (en opposition à "privée") car toute la sécurité de l'ensemble est directement liée au fait que cette clé n'est connue que par l'expéditeur et le destinataire. La cryptographie symétrique est très utilisée et se caractérise par une grande rapidité (cryptage à la volée, "on-the-fly"), des implémentations aussi bien software (Krypto Zone, firewalls logiciels type Firewall-1 et VPN-1 de Checkpoint) que hardware (cartes dédiées, processeurs cryptos 8 à 32 bits, algorithmes cablés...) ce qui accélère nettement les débits et autorise son utilisation massive. Ce type de cryptographie fonctionne habituellement suivant deux procédés différents, le cryptage par blocs et le cryptage de "stream" (en continu).

2.2.2 Le chiffrement par flot

Pour comprendre le cryptage en continu, il suffit de connaître par exemple les videos au format RealVideo très répandues sur internet : on visualise l'image au fur et à mesure que les données sont reçues. Le principe est le même dans le cas de nos "stream-ciphers" : le cryptage est effectué bit-à-bit sans attendre la réception complète des données à crypter. Une technique de chiffrement, du nom de "One-Time Pad" est utilisé pour chiffrer les flux. C'est le chiffrement inconditionnel le plus sûr. Pour cela, on a besoin d'une chaîne aléatoire de la même longueur que le message d'origine, ce qui n'est pas partique. Le but d'un stream cipher est de générer une chaîne aléatoire à partir d'une clé de longueur courte. Une autre technique consiste à "xorer", c'est-à-dire à appliquer un OU exclusif (XOR) au message avec un autre message prédéfini. Bien entendu, cela nécessite que le destinataire (la personne qui décrypte) connaisse le message prédéfini et donc cela rajoute de la complexité au schéma général. Les stream-ciphers sont utilisés aujourd'hui par différentes applications. Pour chiffrer les flux, l'algorithme RC4 est très utilisé.

2.2.3 Le chiffrement par bloc

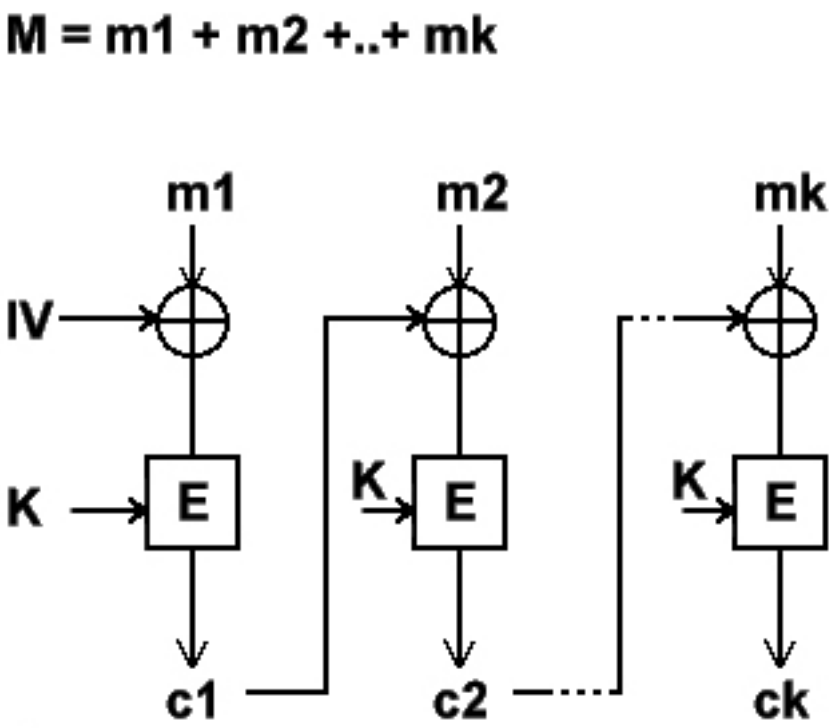
Quatre modes de chiffrement par bloc sont utilisés : Electronic CodeBook (ECB), Cipher Block Chaining (CBC), Cipher FeedBack (CFB) ou Output FeedBack (OFB). Le cryptage en blocs (block-cipher) est au contraire beaucoup plus utilisé et permet une meilleure sécurité. Les algorithmes concernés sont également plus connus (DES, AES, Skipjack...) ; leur nom leur vient du fait qu'ils s'appliquent à des blocs de données et non à des flux de bits (cf. stream-ciphers). Ces blocs sont habituellement de 64 bits mais cela dépend entièrement de l'algorithme utilisé et de son implémentation. De même, la taille de la clé varie suivant l'algorithme et suivant le niveau de sécurité requis ; ainsi, un cryptage de 40 bits (c'est-à-dire utilisant une clé longue de 40 bits) pourra être déclaré faible puisque aisément cassable. Un cryptage de 56 bits (qui est le standard dans le cas du DES) sera qualifié de moyen puisque cassable mais nécessitant pas mal de moyens pour être exploitable (vis-à-vis du temps requis et de la valeur des données). Enfin, un cryptage de 128 bits (valeur standard utilisée par Rijndael alias AES) est plutôt fort à l'heure actuelle. Rappelons à cette occasion que la Loi de Moore prévoit le doublement de la puissance de calcul des processeurs tous les 18 mois (Loi toujours vérifiée de la fin des années 70 à nos jours). Sans entrer dans les détails, il faut savoir que le cassage de cryptés nécessite essentiellement des ressources processeur, RAM et éventuellement ROM ou disque dur si le cassage se fait par précalcul. L'évolution générale est donc extrêmement rapide, sans parler des ordinateurs plus perfectionnés (scientifiques ou autres), à architectures parallèles, ou distribuées... Il reste donc relatif de parler de sécurité absolue, en tout cas en ce qui concerne la cryptographie symétrique. Les quatres modes cités précédement sont plus ou moins indépendants de l'algorithme choisi. Toutefois, tout les algorithmes ne permettent pas d'utiliser tout les modes possibles. Pour mieux comprendre, voyons ces modes plus en détails. Pour désigner le processus de cryptage simple (tel que décrit précédement), on utilisera la notation suivante :



où K désigne la clé utilisée par l'algorithme, E désigne le cryptage en lui-même, M (ou m, mi) désigne le message en clair (c'est-à-dire un bloc) et C (ou c, ci) le chiffré résultant.

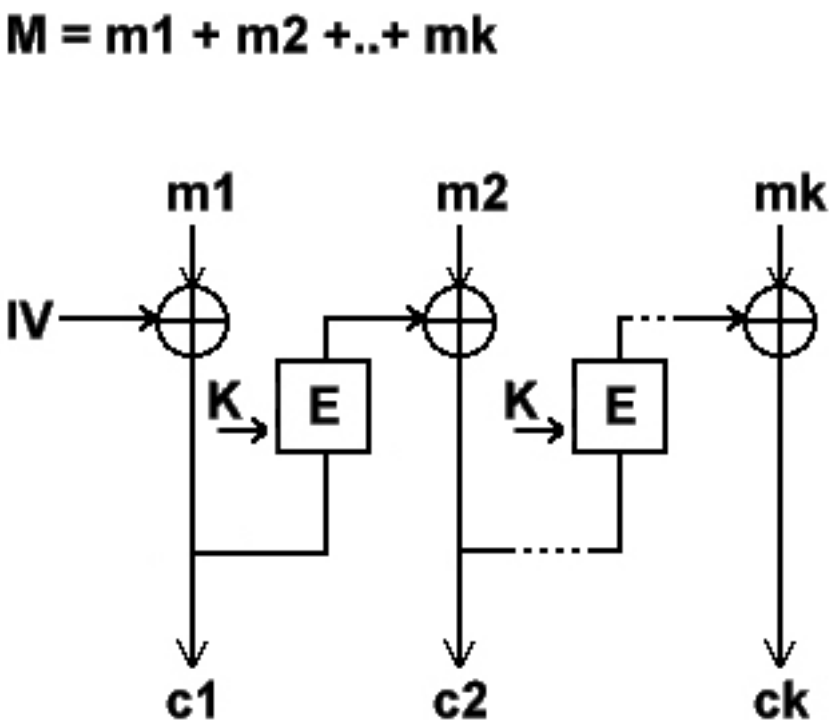
- Le mode Electronic CodeBook (ECB) est le plus simple des modes et s'applique aux block ciphers. Il revient à crypter un bloc indépendamment des autres ; cela permet entre autre de crypter suivant un ordre aléatoire (bases de données, etc...) mais en contre-partie, ce mode est très vulnérable aux attaques. Il est par exemple possible de recenser tous les cryptés possibles (code books) puis par recoupements et analyses statistiques recomposer une partie du message original sans avoir tenté de casser la clé de chiffrement. Il demeure que si la clé fait 128 bits ou plus, cette attaque n'es pas exploitable en pratique de nos jours. Cette technique est sensible à l'inversion ou la duplication de blocs sans que le destinataire s'en apperçoive. On peut l'utiliser pour pipeliner du hardware.

- Le mode Cipher Block Chaining (CBC) peut-être utilisé par les algorithmes en bloc. C'est d'ailleurs le mode le plus courant. Il permet d'introduire une complexité supplémentaire dans le processus de cryptage en créant une dépendance entre les blocs successifs ; autrement dit, le cryptage d'un bloc va être -d'une manière ou d'une autre- lié à ou aux blocs/chiffrés précédents. Le schéma de base sera le suivant :



Le message initial M est divisé en n blocs mi conformément aux spécifications de l'algorithme (par exemple en blocs de 64 bits). Chaque bloc donne un chiffré correspondant (ci) après cryptage suivant le même algorithme E utilisant la même clé K. Comme expliqué ci-dessus, le mode CBC introduit une dépendance entre deux cycles de cryptage : le chiffré obtenu au rang i-1 est utilisé pour obtenir le chiffré du rang i. Concrètement, ce chiffré ci-1 subit un XOR avec le bloc mi. On peut se demander ce qu'il se passe lors du premier cycle d'encodage, lorsqu'il n'y a pas encore de chiffré à xorer avec notre premier bloc. La réponse est que l'on utilise une valeur par défaut prédéfinie appelée Vecteur d'Initialisation (Initialization Vector, IV). Ce vecteur d'initialisation change à chaque session, et doit être transmis au destinataire. Par contre, il n'est pas nécessaire de le chiffrer avant de l'envoyer : il peut être connu de l'adversaire. Il évite l'attaque sur le mode ECB en multipliant la taille de la base de données précalculées. Il ne faut néanmoins pas négliger l'importance de ce vecteur qui peut constituer une faille sérieuse s'il est mal choisi et compromettre ainsi l'intégrité de l'ensemble malgré l'utilisation de composantes fortes (algos, clés, etc). Le déchiffrement est auto-synchonisé comme le mode EBC. Si on pert un bloc de chiffré, on pourra se resynchoniser en ne perdant que deux blocs.

- Le mode Cipher FeedBack (CFB) est un mode destiné aux block ciphers dans le but d'en autoriser une utilisation plus souple, qui s'apparente plus à celle des algorithmes en continu. On peut le considérer comme un intermédiaire entre les deux. En effet, en partant d'un algorithme en bloc utilisant une longueur standard de n bits/blocs, le mode CFB va permettre de crypter des blocs dont la longueur pourra varier de n à 1 bits/blocs. Sachant que dans ce dernier cas, il serait plus économique en calculs d'utiliser directement un algorithme en continu. Quant au cas où la longueur est celle de l'algorithme (à savoir n), le schéma de CFB se simplifie et ressemble quelque peu à celui de CBC (à quelques nuances près) :



- Le mode Output FeedBack (OFB) est une variante de mode CFB précédemment abordé. Il est d'ailleurs parfois appelé internal feedback. Il présente beaucoup de problèmes de sécurité et il est peu conseillé sauf dans le cas où sa longueur est égale à celle de l'algorithme utilisé.

2.3 Les Fonctions de Hachage

2.3.1 Background

Une fonction de hachage est aussi appelée fonction de hachage à sens unique ou "one-way hash function" en anglais. Ce type de fonction est très utilisé en cryptographie, principalement dans le but de réduire la taille des données à traiter par la fonction de cryptage. En effet, la caractéristique principale d'une fonction de hachage est de produire un haché des données, c'est-à-dire un condensé de ces données. Ce condensé est de taille fixe, dont la valeur diffère suivant la fonction utilisée : nous verrons plus loin les tailles habituelles et leur importance au niveau de la sécurité.

2.3.2 Destruction d'information - Conservation de propriétés

Prenons l'exemple des empreintes digitales : dans la perception que nous en avons à l'heure actuelle, une empreinte digitale est unique et représente un individu d'une façon si certaine que nous pouvons la qualifier de sure. Pourtant la connaissance de cette empreinte ne permet pas à elle-seule de remonter à l'individu, ni de reconstituer cet individu. Il faut que la correspondance ait été préalablement établie dans une base de données pour que l'identification puisse avoir lieu par comparaison. C'est exactement ce genre de propriétés que présente une fonction de hachage. En effet, le haché est caractéristique d'un texte ou de données uniques. Différentes données donneront toutes des condensés différents. De plus, tout comme l'empreinte digitale, le condensé ne contient pas assez d'informations en lui-même pour permettre la reconstitution du texte original : c'est pour cela que l'on parle d'ailleurs de fonction à sens unique (l'opération de hachage est destructrice dans

le sens où elle conduit à une perte d'information). Mais il faut bien comprendre que le but d'un condensé n'est pas de véhiculer ou de transporter de l'information. Il est juste représentatif d'une donnée particulière et bien définie. D'autant que les algorithmes de hachage les plus courants sont publics et ne représentent pas en eux-mêmes un secret.

2.3.3 Pourquoi hacher ?

Le but d'un condensé est simple : représenter des données de façon certaine tout en réduisant la taille utile qui sera réellement chiffrée. Prenons l'exemple de la cryptographie asymétrique ; tout le monde admet qu'elle est très sure, fiable et durable. Néanmoins, sa complexité (calcul sur des nombres premiers de plusieurs centaines de chiffres par exemple) entraine une inévitable lourdeur d'emploi (charge CPU, etc...). On évite donc de l'utiliser pour de grandes masses de données ou pour des chiffrements de flux. Par contre imaginez que vous souhaitiez envoyer un fichier par mail, mais que ce fichier est de taille importante. Vous souhaitez de plus rassurer le destinataire sur la provenance de ce fichier (vous) et sur son contenu. Plutôt que de chiffrer votre fichier directement avec votre clé privée, vous aller hacher votre fichier et chiffrer le condensé obtenu avec votre clé privée. Vous enverrez ensuite votre fichier original ainsi que le condensé chiffré (la signature) à votre destinataire. Celui-ci va, lors de la réception, hacher d'une part le fichier reçu et d'autre part déchiffrer le condensé reçu (au moyen de votre clé publique). S'il n'y a pas égalité entre les 2 résultats, cela signifiera :

- soit que la signature n'est plus la votre, donc que quelqu'un a intercepté le fichier (pour le modifier ou le remplacer, etc...)
- soit que le fichier n'est plus le même que l'original (mais la signature n'a pas été remplacée) ; dans ce cas, la hachage ne peut plus donner le même condensé ce qui conduit au rejet lors du test de comparaison.

Dans les 2 cas, ni l'intégrité ni l'authentification du fichier n'ont été vérifiées. Il ne faut donc pas faire confiance au fichier. Nous voyons comment dans ce cas simple, l'utilisation d'une fonction de hachage permet de s'assurer de l'intégrité des données et indirectement de les authentifier. Il existe bien sûr de nombreuses autres applications pour les fonctions de hachage, comme les MACs (message authentication code), certificats, etc...

2.3.4 Fonctions de hachage usuelles

- MD4 et MD5 (Message Digest) furent développées par Ron Rivest. MD5 produit des hachés de 128 bits en travaillant les données originales par blocs de 512 bits.
- SHA-1 (Secure Hash Algorithm 1), comme MD5, est basé sur MD4. Il fonctionne également à partir de blocs de 512 bits de données et produit par contre des condensés de 160 bits en sortie. Il nécessite donc plus de ressources que MD5.
- SHA-2 (Secure Hash Algorithm 2) a été publié récemment et est destiné à remplacer SHA-1. Les différences principales résident dans les tailles de hachés possibles : 256, 384 ou 512 bits. Il sera bientôt la nouvelle référence en termes de fonction de hachage.
- RIPEMD-160 (Ripe Message Digest) est la dernière version de l'algorithme RIPEMD. La version précédente produisait des condensés de 128 bits mais présentait des failles de sécurité importantes. La version actuelle reste pour l'instant sure ; elle produit comme son nom l'indique des condensés de 160 bits. Un dernier point la concernant est sa relative gourmandise en termes de ressources et en comparaison avec SHA-1 qui est son principal concurrent.

2.3.5 Importance de la taille des condensés

On peut se demander pourquoi il existe plusieurs tailles de condensés ou encore pourquoi celle-ci est fixe. Il faut garder à l'esprit le but ultime d'un haché qui est d'être le plus court possible, tout en gardant ses propriétés. Or, cela nous amène tout naturellement au problème des collisions, également connu sous la dénomination de théorême ou paradoxe des anniversaires. Je n'aborderai pas ici l'explication originale de ce théoreme.

Prenons donc notre haché H, qui présente une longueur de n bits. Nous pouvons d'ores et déjà déduire qu'il n'existe que 2^n hachés de ce type possibles (puisque chaque bit n'a que 2 valeurs possibles, 0 ou 1). Le problème survient quand on se rend compte que de l'autre côté, nous pouvons avoir une infinité de textes ou données initiaux (dont la taille, elle, n'est pas fixée). Nous risquons donc, un jour ou l'autre, de produire un haché qui pourrait correspondre à un autre texte original (ou à plusieurs) : c'est la perte de la propriété principale d'un condensé, qui est l'unicité. Nous avons trouvé une collision.

Le théorême des anniversaires prouve qu'il faut $2^{n/2}$ essais pour trouver une collision au hasard. C'est le chiffre qui sera utilisé pour évaluer la force d'une fonction de hachage. Pourtant, il ne faut pas négliger le fait que la collision citée précédemment a été obtenue au hasard, ce qui n'est pas exploitable par une personne malveillante. En effet, le but serait de trouver un message significatif et bien formé conduisant au même haché, ce qui augmente considérablement les essais et calculs nécessaires (et le rend quasiment impossible). Quoiqu'il en soit, cela suffit en théorie pour briser la propriété d'unicité de notre condensé...

D'un point de vue pratique, et dans l'état de l'art actuel, il est généralement accepté que 2^{56} calculs représentent un défit réalisable. Comme exemple, les clés DES de 56 bits sont réellement faibles et crackables. En conséquence, avec n/2=56 et n=112, le théorême des anniversaires nous indique que les hachés de 112 bits sont faibles et donc insuffisants à l'heure actuelle. De la même manière, les hachés de 128 bits (n/2=64) ne représentent plus une sécurité à moyen terme. C'est pour cela que la norme actuelle est à 160 bits (n/2=80) voire plus dans le cas de SHA-2.

2.4 L'AES

2.4.1 Background

L'AES (Advanced Encryption Standard) est, comme son nom l'indique, un standard de cryptage symétrique (c.f. fiche sur les algorithmes symétriques) destiné à remplacer le DES (Data Encryption Standard) qui est devenu trop faible au regard des attaques actuelles.

2.4.2 Présentation générale

Historiquement, le développement de l'AES a été instigué par le NIST (National Institue of Standards and Technology) le 2 janvier 1997. L'algorithme a été choisi il y a peu de temps : il s'agit de l'algorithme Rijndael (prononcer "Raindal"). Cet algorithme suit les spécifications suivantes :

- l'AES est un standard, donc libre d'utilisation, sans restriction d'usage ni brevet.
- c'est un algorithme de type symétrique (comme le DES)
- c'est un algorithme de chiffrement par blocs (comme le DES)
- il supporte différentes combinaisons [longueur de clé]-[longueur de bloc] : 128-128, 192-128 et 256-128 bits (en fait, Rijndael supporte également des tailles de blocs variables, mais cela n'est pas retenu dans le standard)

En termes décimaux, ces différentes tailles possibles signifient concrètement que :

3.4 x 10^{38} clés de 128-bit possibles

6.2 x 10^{57} clés de 192-bit possibles

1.1 x 10^{77}clés de 256-bit possibles

Pour avoir un ordre d'idée, les clés DES ont une longueur de 56 bits (64 bits au total dont 8 pour les controls de parité), ce qui signifie qu'il y a approximativement 7.2 x 10^{16} clés diférentes possibles. Cela nous donne un ordre de 10^{21} fois plus de clés 128 bits pour l'AES que de clés 56 bits pour le DES. En supposant que l'on puisse construire une machine qui pourrait cracker une clé DES en une seconde (donc qui puisse calculer 2^{55} clés par seconde), alors cela prendrait encore 149 mille milliards d'années pour cracker une clé AES. Pour donner un ordre d'idée plus concret, l'univers est vieux de 20 milliards d'années au maximum.

Pour conclure sur cet aspect, on voit que le standard AES répond aux mêmes exigences que le DES mais il est également beaucoup plus sûr et flexible que son prédécesseur.

2.4.3 Caractéristiques et points forts de l'AES

Le choix de cet algorithme répond à de nombreux critères plus généraux dont nous pouvons citer les suivants :

- securité ou l'effort requis pour une éventuelle cryptanalyse.
- facilité de calcul : cela entraine une grande rapidité de traitement besoins en ressources et mémoire très faibles
- flexibilité d'implémentation : cela inclut une grande variété de platformes et d'applications ainsi que des tailles de clés et de blocs supplémentaires (c.f. ci-dessus).
- hardware et software : il est possible d'implémenter l'AES aussi bien sous forme logicielle que matérielle (cablé)
- simplicité : le design de l'AES est relativement simple

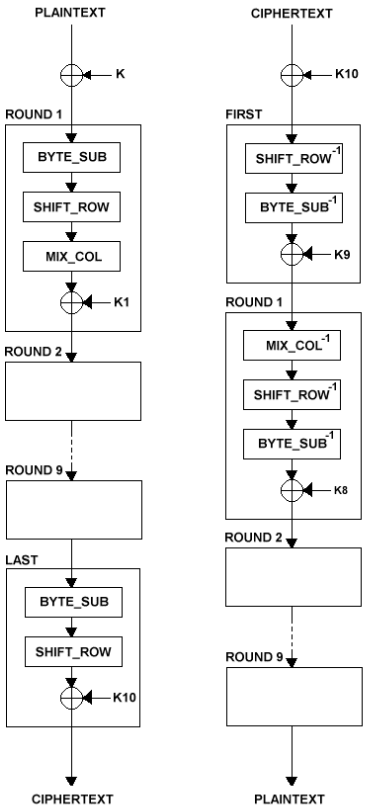
Si l'on se réfère à ces critères, on voit que l'AES est également un candidat particulièrement approprié pour les implémentations embarquées qui suivent des règles beaucoup plus strictes en matière de ressources, puissance de calcul, taille mémoire, etc... C'est sans doute cela qui a poussé le monde de la 3G (3ème génération de mobiles) à adopter l'algorithme pour son schéma d'authentification "Millenage".

2.4.4 Détails techniques

L'AES opère sur des blocs de 128 bits (plaintext P) qu'il transforme en blocs cryptés de 128 bits (C) par une séquence de Nr opérations ou "rounds", à partir d'une clé de 128, 192 ou 256 bits. Suivant la taille de celle-ci, le nombre de rounds diffère : respectivement 10, 12 et 14 rounds. Le schéma suivant décrit succintement le déroulement du chiffrement :

- BYTE_SUB (Byte Substitution) est une fonction non-linéaire opérant indépendament sur chaque bloc à partir d'un table dite de substitution.
- SHIFT_ROW est une fonction opérant des décalages (typiquement elle prend l'entrée en 4 morceaux de 4 octets et opère des décalages vers la gauche de 0, 1, 2 et 3 octets pour les morceaux 1, 2, 3 et 4 respectivement).
- MIX_COL est une fonction qui transforme chaque octet d'entrée en une combinaison linéaire d'octets d'entrée et qui peut être exprimée mathématiquement par un produit marticiel sur le corps de Galois (2^8).
- le + entouré d'un cercle désigne l'opération de OU exclusif (XOR).
- Ki est la ième sous-clé calculée par un algorithme à partir de la clé principale K.

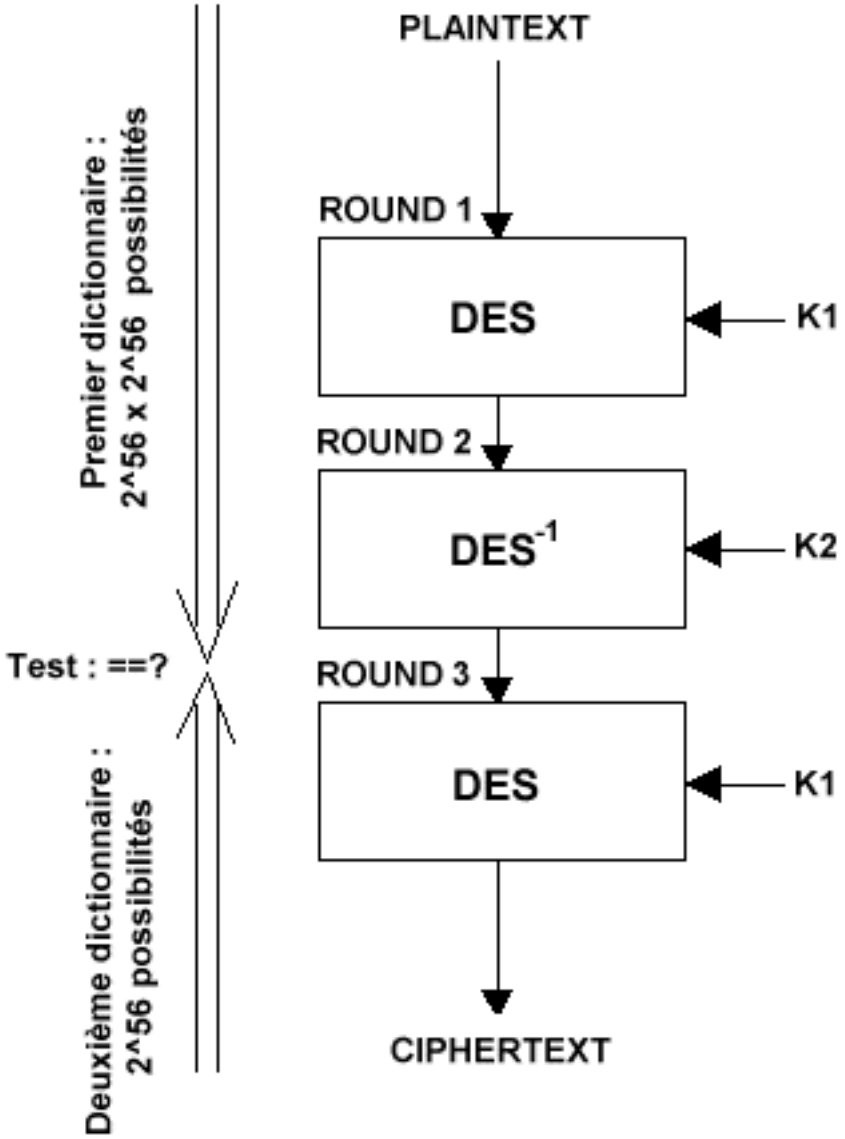
Le déchiffrement consiste à appliquer les opérations inverses, dans l'ordre inverse et avec des sous-clés également dans l'ordre inverse.



2.4.5 Comparaisons des algorithmes au niveau de la sécurité

Attaques par dictionnaires

Nous allons comparer ici l'AES au 3DES qui est son concurrent le plus direct (le DES n'étant pratiquement plus utilisé dans sa forme simple). Le 3DES est, comme son nom l'indique, l'enchainement de 3 DES simples dans l'ordre DES / DES^{-1} / DES. Il est évident à prime abord que chaque opération utilise une clé distincte, car sans cela les 2 premières s'annuleraient (DES / DES^{-1}). Mais en pratique, on n'utilise que 2 clés différentes (que l'on alterne) car l'utilisation d'une troisième clé ne rajoute aucune sécurité. En effet, l'attaque la plus courante contre le triple DES consiste à créer des dictionnaires multiples de facon à scinder le schéma en 2 parties et diminuer ainsi d'autant le nombre de possibilités à tester. En pratique, on séparera les 2 premières opérations DES de la 3ième et dernière.



La première partie conduit à l'élaboration d'un dictionnaire dont la taille est définie par le calcul suivant : le premier DES utilise une clé de 56 bits, il y a donc 2^{56} cas possibles. C'est pareil pour le deuxième DES, sauf que qu'il faut le multiplier au premier cas, soit un total de 2^{112} possibilités.

La deuxième partie ne comporte qu'un seul DES, donc 2^{56} possibilités pour la clé. Il suffit ensuite de faire correspondre ces 2 dictionnaires pour trouver la valeur qui est commune aux 2, nous donnant ainsi la bonne combinaison de clés.

De manière générale et arrondie, la sécurité de l'algorithme peut donc être évaluée à 2^{113}.

En ce qui concerne l'AES, c'est un algorithme qui ne présente qu'un seule étape, donc le calcul est simple : comme cité précédement, il y 2^{128} clés possibles (dans la version minimale ou la clé ne fait "que" 128 bits de long). C'est directement la force de l'algorithme.

Attaques par cryptanalyse différentielle (DC)

L'attaquant choisit des textes clairs présentant une différence fixe, calcule les chiffrés (en ayant accès au système) et leurs différences puis assigne des probabilités à certains types de clés. Plus le nombre d'essais augmente, plus la probabilité de la bonne clé devient forte. Dans le cas du DES simple, cette attaque nécessite 2^{47} textes clairs et 2^{47} chiffrements pour retrouver la clé ; néanmoins, les textes clairs doivent être soigneusement choisis. L'AES est lui résistant à ce type d'attaque.

Attaques par cryptanalyse linéaire (LC)

L'attaquant utilise des approximations linéaires pour décrire les opérations conduisant au chiffré. Comme précédemment, plus le nombre d'essais augmente, plus la probabilité de la bonne clé devient forte. Cette attaque est actuellement la plus performante puisqu'elle ne nécessite que 2^{43} textes clairs et 2^{43} chiffrements pour retrouver une clé DES (simple). L'AES est lui résistant à ce type d'attaque.

2.4.6 Conclusion

En conclusion, l'AES est plus sûr que le 3DES car il présente, entre autres, une plus grande résistance aux attaques par dictionnaires de clés. Les autres attaques ne sont pas appliquables dans son cas.

2.5 SSL

2.5.1 Qu'est-ce que c'est ?

SSL(Secure Socket Layer) est un protocole de sécurisation des échanges, développé par Netscape. Il a été conçu pour assurer la sécurité des transactions sur Internet (notamment entre un client et un serveur), et il est intégré depuis 1994 dans les navigateurs. Il existe plusieurs versions : la version 2.0 développée par Netscape ; la version 3.0 qui est actuellement la plus répandue, et la version 3.1 baptisée TLS (transport Layer Security, RFC 2246) et standardisée par l'IETF (Internet Engineering Task Force). SSL fonctionne de manière indépendante par rapport aux applications qui l'utilisent ; il est obligatoirement au dessus de la couche TCP et certains le considèrent comme un protocole de niveau 5 (couche session).

2.5.2 Pourquoi SSL ?

SSL permet d'assurer les services de sécurité suivants :
- confidentialité : elle est obtenue par l'utilisation d'algorithmes à chiffrement symétrique de blocs comme DES, FORTEZZA, IDEA, 3DES ou RC2, ou par des algorithmes à chiffrement symétrique de flots comme RC4 (la longueur des clés peut être limitée suivant les législations propres à l'exportation, et le padding correspondant s'applique aux algorithmes par blocs).
- intégrité : l'intégrité des données est assurée par l'utilisation de MACs (Message Authentication Code) basés sur les fonctions de hachage MD5 (16 octets) ou SHA-1 (20 octets).
- authentification : SSL permet l'authentification des 2 entités (authentification client facultative) basé sur des certificats X.509, et l'authentification des données grâce aux MACs.

2.5.3 Les sous-protocoles de SSL

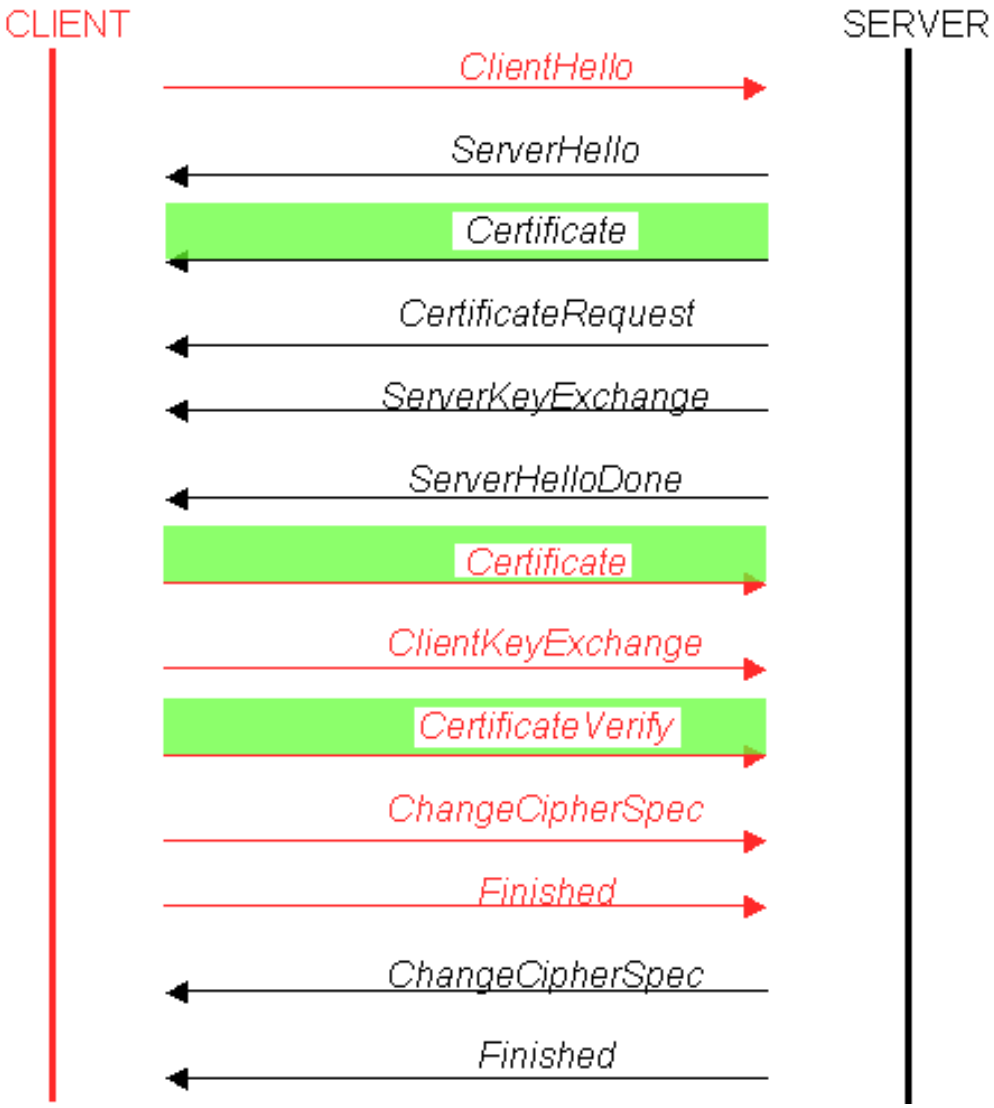
Le protocole SSL est constitué de quatre sous-protocoles :

1. Handshake qui permet l'authentification mutuelle du client et serveur, la négociation des algorithmes de chiffrement, de hachage, et l'échange des clés symétriques qui assurent le chiffrement.

2. le protocole SSL Change Cipher Spec
3. le protocole SSL Alert
4. le protocole SSL Record

2.5.4 Déroulement dés échanges SSL

Déroulement habituel d'un handshake SSL avec authentification mutuelle :
– en noir, les échanges initiés par le serveur.
– en rouge, les échanges initiés par le client.
– en jaune, les échanges correspondants à l'authentification du client.



Les échanges définis par le protocole SSL se déroulent en deux phases :

1. Première phase : authentification du serveur Suite à la requête d'un client, le serveur envoie son certificat au client et lui liste les algorithmes cryptographiques, qu'il souhaite négocier. Le client vérifie la validité du certificat à l'aide de la clé publique du CA (Certificate Authority) contenue dans le navigateur. Si le certificat est valide, le client génère un pré-master secret (PMS) de 48 octets qui servira à dériver le master secret (MS) de même taille, 48 octets,

ce qui représente l'entropie maximale de la clé. Ce PMS est chiffré avec la clé publique du serveur puis transmis à ce dernier. Les données échangées par la suite entre le client et le serveur sont chiffrées et authentifiées à l'aide de clés dérivées de la clé maître.

2. Deuxième phase : authentification (optionnelle) du client Le serveur (et seulement lui) peut demander au client de s'authentifier en lui demandant tout d'abord son certificat. Le client réplique en envoyant ce certificat puis en signant un message avec sa clé privée (ce message contient des informations sur la session et le contenu de tous les échanges précédents).

Remarques :
- L'authentification du client est facultative et au bon vouloir du serveur. Elle est en fait rarement utilisée dans la vie courante (qui possède une paire de clés RSA certifiée ?).
- L'authentification du réseau intervient dans tous les cas avant celle du client, ce qui est un gage de sécurité accrue.
- Comme dans IPSec (IKE phase 1), l'authentification reprend les échanges précédents et valide ainsi tout le handshake.
- Notez enfin que seule la clé publique du serveur est utilisée pour faire du chiffrement ; celle du client ne sert que pour de la signature.

2.5.5 Les variables d'état d'une session SSL

Une session SSL est définie par les variables suivantes :
- Session ID (l'dentifiant de session) une séquence arbitraire de 32 octets choisie par le serveur pour identifier une session.
- Peer certificate (le certificat du pair) c'est un certificat X 509 du correspondant (soit pour un serveur ou un client).
- Compression method l'algorithme de compression utilisé, NULL pour l'instant (ce champ reste vide)
- Cipher spec (la suite de chiffrement) définit les algorithmes de chiffrement et de hachage
- MasterSecret c'est un clé de 48 octets partagée entre le client et le serveur.
- Is resumable (le drapeau) c'est un flag qui indique si il est possible d'ouvrir de nouvelles connexions sur la session en question.

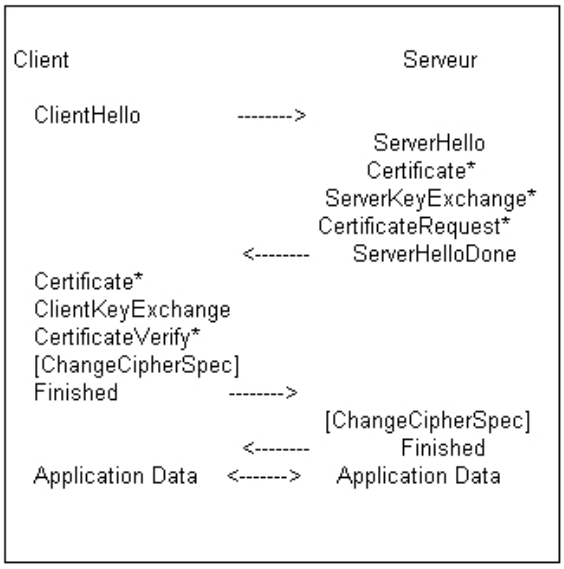
2.5.6 Les variables d'état d'une connexion SSL

Les paramètres qui définissent une connexion SSL sont ceux qui se seront rafraîchis pendant une session lors d'établissement d'une nouvelle connexion. Ces paramètres sont :
- Server_random et Client_random : deux nombres aléatoires de 32 octets, générés par le client et le serveur lors de chaque connexion.
- Server_MAC_write_secret : clé secrète utilisé par le serveur pour calculer les MACs
- Client_MAC_write_secret : clé secrète utilisé par le client pour calculer les MACs
- Server_write_key : clé symétrique utilisé par le serveur pour le chiffrement des données.
- Client_write_key : clé symétrique utilisé par le client pour le chiffrement des données.
- Initialization vectors : vecteur d'initialisation pour le chiffrement par bloc en mode CBC (Cipher Bloc Chaining), l'un du coté serveur et l'autre du coté client.
- Sequence number : chaque message est numéroté, l'un pour le serveur, l'autre par le client, et chacun codé sur 8 octets.

2.5.7 Le protocole Handshake

Ce protocole permet l'authentification obligatoire du serveur, du client est optionnelle, et de négocier pour choisir les suites de chiffrement qui seront utilisées lors de la session.

2.5.8 Déroulement des échanges du Handshake



* message optionnel

- ClientHello : ce message contient : la version : version du protocole SSL,client_random : nombre aléatoire, session_id :l'identificateur de session, cipher suite :la liste des suites de chiffrement choisies, et algo décompression : la liste des méthodes de compression.
- ServerHello : ce message contient : la version : version du protocole SSL,Server_random : nombre aléatoire, session_id :l'identificateur de session, cipher suite : la liste des suites de chiffrement choisies, et algo décompression : la liste des méthodes de compression.
- Certificate : ce message contient soit le certificat de serveur
- ServerKeyExchange : contient le certificat de signature
- CertificateRequest : le serveur réclame un certificat au client
- ServerHelloDone : la fin de l'envoi de message
- ClientKeyExchange : ce message contient le PreMastersecret chiffré à l'aide de la clé publique du serveur.
- CertificateVerify : vérification explicite du certificat du client
- Finished : fin du protocole Handshake et le début de l'émission des données

2.5.9 Le protocole ChangeCipherSpec (CCS)

Ce protocole comprend un seul et unique message (1 octet) qui porte le même nom que le protocole, il permet d'indiquer au protocole Record la mise en place des algorithmes de chiffrement qui viennent d'être négociés.

2.5.10 Le Protocole SSLRecord

Ce protocole intervient après l'émission du message ChangeCipherSpec.il permet de garantir :

- la confidentialité à l'aide de chiffrement des données
- l'intégrité à l'aide de génération d'un condensât.

2.5.11 Le protocole SSL Alert

Ce protocole génère des messages d'alerte suite aux erreurs que peuvent s'envoyer le client et le serveur. Les messages sont composés de 20 octets, le premier étant soit fatal soit warning. Si

le niveau de criticité du message est fatal, la connexion SSL est abandonnée. Le deuxième octet est utilisé pour le code d'erreur.

Liste des Messages du protocol Alert :

- Les erreurs fatales sont :
- bad_record_mac : réception d'un MAC erroné
- decompression_failure : les données appliquées à la fonction de compression sont invalides
- handshake_failure : impossibilité de négocier les bons paramètres
- illegal_parameter : un paramètre échangé au cours du protocole Handshake ne correspondait pas avec les autres paramètres
- unexpected_message : message non reconnu.

Les warnings sont :

- bad_certificate : le certificat n'est pas bon
- certificate_expired : certificat périmé
- certificat_revoked : certificat révoqué
- certificat_unknown : certificat invalide pour des raisons précisés au dessus
- close_notify : la fin d'une connexion
- no_certificate : réponse négative à une demande de certificat
- unsupported_certificate : le certificat reçu n'est pas reconnu

2.5.12 Les ports utilisées par SSL

Nom	Port
HTTPS (HTTP en SSL)	443
SMTPS (SMTP en SSL)	465
NNTPS	563
LDAPS (LDAP en SSL)	636
POP3S	995
IMAPS	995
TELNETS	992

En pratique, pour accéder à un serveur qui utilise les services SSL, on ajoute un "s" lors de la spécification du protocole.

Exemple : https ://www.monserveur.com

2.5.13 Implémentations

plusieurs offres commerciales du serveur SSL sont disponibles, par exemple :

- SSLeay
- Netscape Entreprise Server
- Apache
- Oracle Web Apllication Server
- Internet Information Server (IIS)
- Lotus Domino d'IBM
- Java Server de Sun Microsystems

2.5.14 Aspects cryptographiques de SSL

Le handshake SSL permet aux 2 entités de choisir une suite d'algorithmes cryptographiques pour assurer la sécurité de leurs échanges. Cette suite sera de la forme SSL_X_WITH_Y_Z où :

- X désigne l'algorithme utilisé pour l'échange de clés : RSA ou Diffie-Hellman avec signature DSS ou RSA. Notez que DH n'est supporté que par la version 3.0 et non par la 2.0 de SSL. D'autre part, son implémentation n'est pas sensible aux attaques type man-in-the-middle car les paramètres d'exponentiation sont authentifiés par les 2 extrémités.
- Y désigne l'algorithme de chiffrement (voir le paragraphe Pourquoi SSL ?)
- Z désigne l'algorithme de hachage (voir le paragraphe Pourquoi SSL ?)

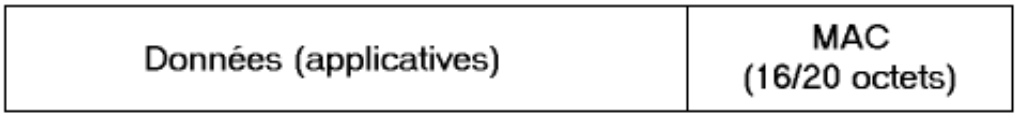
La version 3.0 de SSL permet 31 possibilités de suites d'algorithmes cryptographiques différentes. Une de ces possibilités est de ne rien utiliser, ce qui revient à utiliser des algorithmes fictifs ne modifiant pas les données, ce que nous désignons comme suit :

SSL_NULL_WITH_NULL_NULL

Un autre exemple est :

SSL_RSA_WITH_DES_CBC_MD5

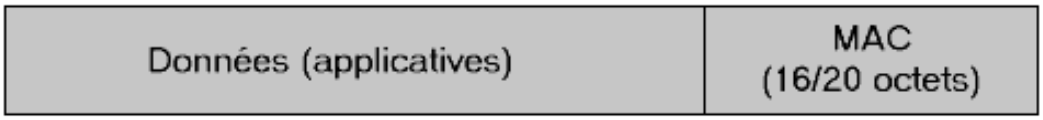
Intégrité et authentification



L'intégrité est assurée au moyen de MACs (messages authentication codes), qui ont la double spécificité d'assurer à la fois intégrité et authentification des données. Un MAC allie une fonction de hachage (ici MD5 ou SHA-1, d'où une longueur finale de 16 ou 20 octets) à une clé secrète partagée par les 2 entités. Une façon habituelle de générer des MACs est d'utiliser la fonction HMAC (RFC 2104) ; ici, SSL définit sa propre fonction pour MACer, mais nous n'entrerons pas dans les détails. Attention, les illustrations font abstraction des headers, qui ne sont pas inclus dans les données protégées.

Confidentialité

Le chiffrement peut-être soit par flots, soit par blocs, suivant l'algorithme choisi. Prenons l'exemple du chiffrement par flot (en gris les données chiffrées) :



Après génération du MAC, les données sont directement processées dans la fonction de chiffrement.

Quant au chiffrement par blocs, il est nécessaire de rajouter du padding avant le traitement, pour que la longueur totale des données à traiter soit un multiple de la taille d'un seul bloc (n x 64 bits par exemple).

2.5.15 Evolutions du standard

La prochaine version du standard aura quelques changements, parmi lesquels on retrouve :

– Support obligatoire de DSA et D-H ; RSA devrait également être supporté en facultatif.

- Remplacement de la fonction prorpiétaire de génération de MACs par le standard HMAC (RFC 2104)
- Protection du numéro de version de SSL par les opérations de hachage (pour éviter les attaques par rollback par exemple)
- Nouveau générateur de nombres aléatoires (basé MD5 et SHA-1 à la fois), nouveaux messages d'alerte ("Decryption failed")
- ...

2.5.16 Les attaques et faiblesses de SSL

- SSL est théoriquement vulnérable aux attaques par force brute en cas d'utilisation de clés 40 bits, il est donc conseillé d'utiliser des clés de 128 bits.
- SSL est très vulnérable aux attaques par le milieu (man in the middle) : l'attaquant intercepte (physiquement) la requête du client et se fait passer pour le serveur auprès de lui, tout en se faisant passer pour un client auprès du serveur légitime. Il reçoit donc la totalité du flux supposé protégé.
- SSL est faible dans le sens où il n'impose pas l'authentification client (ce serait d'ailleurs difficilement gérable en pratique).
- SSL est faible enfin car il présente des souplesses dans son implémentation, notamment en ce qui concerne la vérification des certificats des serveurs. En effet, le client devrait vérifier la signature, la validité ainsi que le statut de révocation du certificat (au moyen de CRLs ou du protocole OCSP) et devrait s'assurer que le CA concerné appartient bien aux CAs auxquels il fait confiance. Or cela n'est pas obligatoire, et peut permettre à un attaquant de substituer sa propre clé publique à celle du serveur original puis de rediriger le traffic vers son faux site tout en faisant croire au client qu'il communique bien avec son serveur légitime. Il est donc important de bien vérifier les URLs par exemple.
- SSL est éventuellement vulnérable à des attaques plus poussées, basées sur des propriétés cryptographiques. L'utilisation de certificats par exemple nécessite une grande vigilance (date de validité, multitude de CAs inconnus certifiant tout et n'importe quoi... Il suffit de jeter un coup d'oeil à la liste contenue dans votre navigateur pour s'en rendre compte) ; il existe également des attaques par brute-force sur l'échange de la clé symétrique (2^20 essais) ou des attaques dites de rollback dans lesquelles l'attaquant cherche à modifier le choix des algos d'échanges de clés de façon à ce que les 2 entités n'utilisent pas les mêmes (RSA et DH par exemple). Ainsi, cet attaquant pourra déchiffrer le message car les paramètres fournit par le serveur dans le cas d'un algorithme n'offrent aucune sécurité si on les applique à un autre... Cette attaque peut-être réalisée si une session 3.0 est résumée en session 2.0 par exemple.

2.5.17 Conclusion

Le protocole SSL est actuellement le seul protocole de sécurisation déployé et utilisé à grande échelle, son grand avantage étant sa transparence par rapport au protocole TCP. Il garantit l'authentification, la confidentialité et l'intégrité des données. Avec son architecture modulaire, il ne se limite pas à des applications traditionnelles, puisque il intègre les réseaux sans fil comme le WAP (Wirless Transport Layer) sous le nom WTLS (Wireless Transport Layer Security).

2.5.18 Références

www.netscape.com/eng/ssl3

2.6 Les PKI

2.6.1 Qu'est-ce que c'est ?

PKI (Public Key Infrastructure) est un système de gestion des clefs publiques qui permet de gérer des listes importantes de clefs publiques et d'en assurer la fiabilité, pour des entités généralement dans un réseau. Elle offre un cadre global permettant d'installer des éléments de sécurité tels que la confidentialité, l'authentification, l'intégrité et la non-répudiation tant au sein de l'entreprise que lors d'échanges d'information avec l'extérieur. Une infrastructure PKI fournit donc quatre services principaux :

- fabrication de bi-clés.
- certification de clé publique et publication de certificats.
- Révocation de certificats.
- Gestion la fonction de certification.

2.6.2 Techniquement

Une infrastructure à clé publique utilise des mécanismes de signature et certifie des clés publiques qui permettent de chiffrer et de signer des messages ainsi que des flux de données, afin d'assurer la confidentialité, l'authentification, l'intégrité et la non-répudiation.

Signature

Dans la signature nous avons une bi-clé : une clé (privé) pour la création de siganture et une clé (publique) pour la vérification de signature, pour signer un message voici comment se passe :

1) à l'aide de la clé privée de signature de l'expéditeur, une empreinte connue sous le nom "message digest" est générée par hachage en utilisant l'algorithme SHA-1 ou MD5, le plus utilisé étant SHA-1. Cette empreinte est ensuite cryptée avec cette clé privée de signature.

2) on joint au message l'empreinte et le certificat contenant la clé publique de signature.

3) le destinataire vérifie la validité du certificat et sa non révocation dans l'annuaire.

4) le destinataire transforme l'empreinte avec la clé publique de signature ainsi validée. Cette opération permet de s'assurer de l'identité de l'expéditeur.

5) ensuite le destinataire génère une empreinte à partir de message reçu en utilisant le même algorithme de hachage. Si les deux empreintes sont identiques, cela signifie que le message n'a pas été modifié.

Donc la signature vérifie bien l'intégrité du message ainsi que l'identité de l'expéditeur.

Exemples d'algorithme de signature : RSA,DSA

Défintions

- Confidentialité : Les informations échangées deviennent illisibles,cette confidentialité est assurée par le chiffrement
- Authentification : identification de l'origine de l'information.
- Non-répudiation : l'émetteur des données ne pourra pas nier être à l'origine du message.
- Intégrité : fonction permettant d'assurer que l'information n'a pas subi de modification .

Chiffrement

Il y a deux types de chiffrement possible

Chiffrement à clé secrète (symétrique) :

Lémetteur utilise une clé pour chiffrer le message et le destinataire utilise la même clé (le même algorithme mais en sens inverse) pour déchiffrer le message.

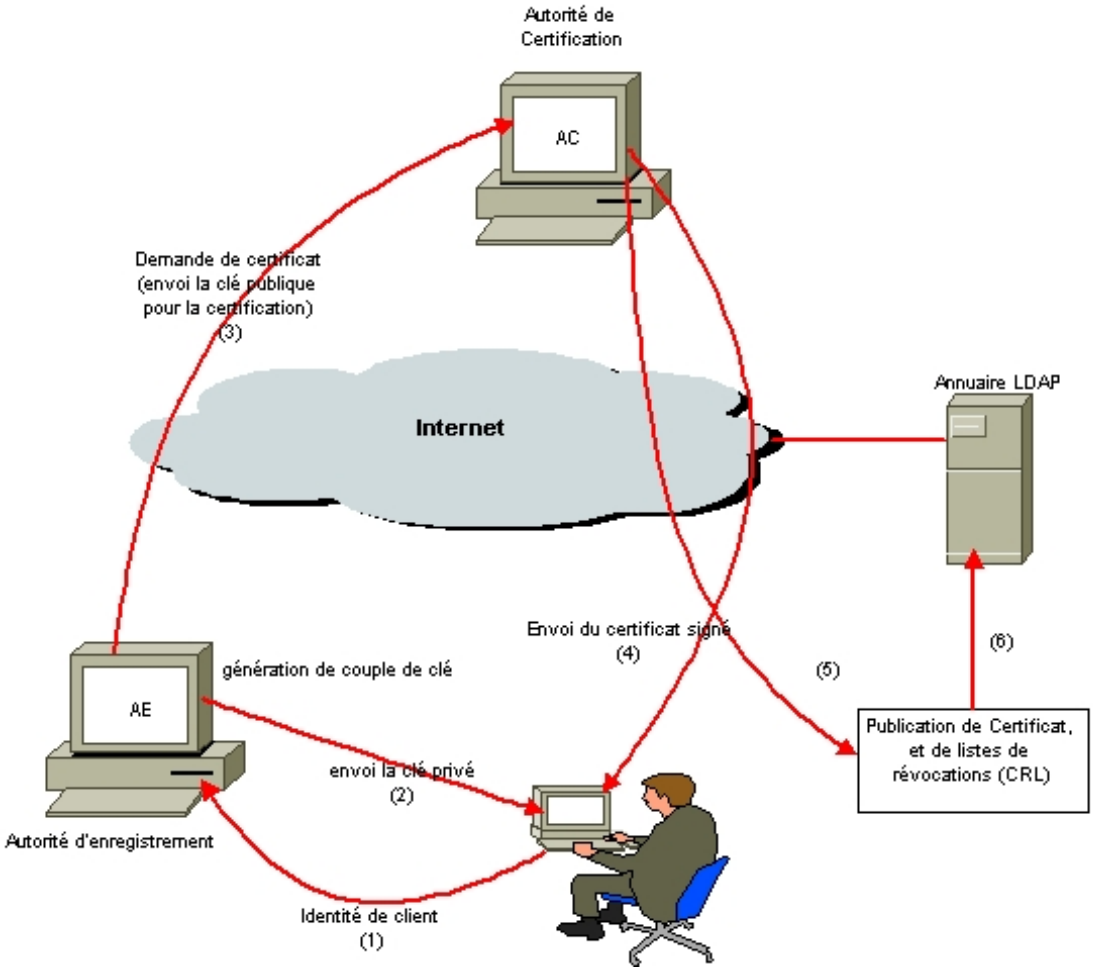
Chiffrement à clé publique (asymétrique) :

Un message chiffré avec une clé publique donnée ne peut être déchiffré quavec la clé privée correspondante. Par exemple si A souhaite envoyer un message chiffré à B, il le chiffrera en utilisant

la clé publique de B (qui peut être publié dans l'annuaire). La seule personne qui déchiffre le message est le détenteur de la clé privée de B.

Exemples d'algorithme de chiffrement : - Symétrique : DES ;AES - Asymétrique : RSA

2.6.3 Organisation d'une PKI



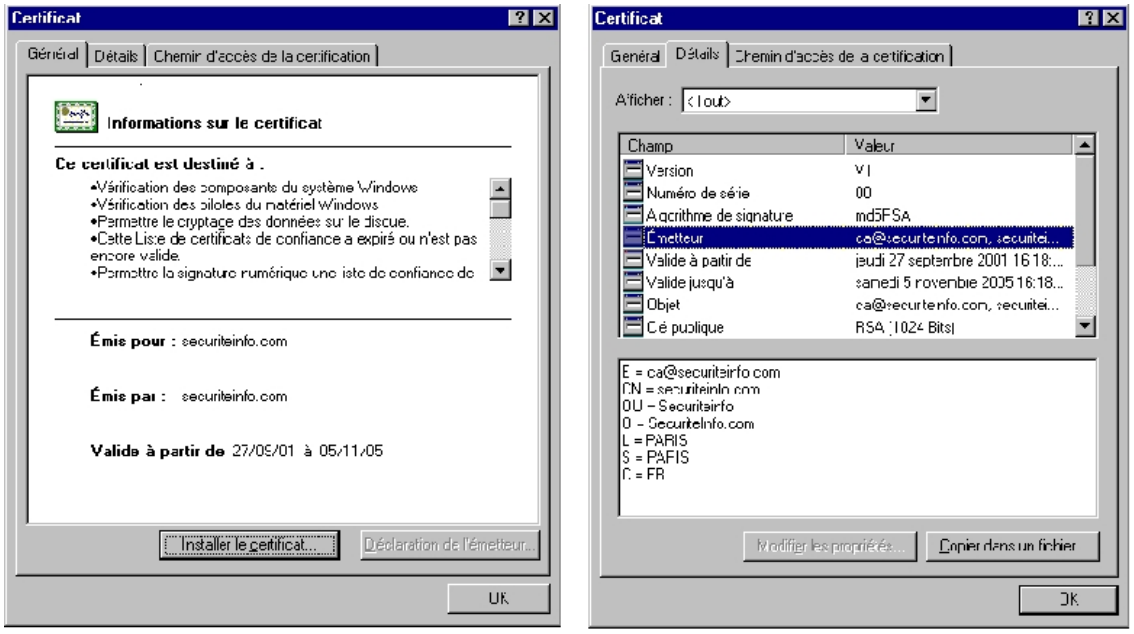
Dans une infrastructure à clé publique ; pour obtenir un certificat numérique, l'utilisateur fait une demande auprès de l'autorité d'enregistrement. Celle-ci génère un couple de clé (clé publique, clé privée), envoie la clé privée au client, applique une procédure et des critères définis par l'autorité de certification qui certifie la clé publique et appose sa signature sur le certificat, parfois fabriqué par un opérateur de certification.

2.6.4 La Structure d'un Certificat numérique selon la norme X509

- Version : indique à quelle version de X509 correspond ce certificat
- Numéro de série : Numéro de série du certificat Algorithme de signature : identifiant du type de signature utilisée
- Emetteur : Distinguished Name (DN) de l'autorité de certification qui a émis ce certificat.
- Valide à partir de : la date de début de validité de certificat
- Valide jusqu'à : la date de fin de validité de certificat
- Objet : Distinguished Name (DN) de détenteur de la clef publique

- Clé publique : infos sur la clef publique de ce certificat
- Contraintes de base : extensions génériques optionnelles
- Utilisation de la clé : l'objet d'utilisation de la clé
- Algorithme thumbprint : alogrithme de signature
- Thumbprint : signature numérique de l'autorité de certification sur l'ensemble des champs précédents.

2.6.5 Exemple d'un Certificat X509 généré par OPENSSL



2.6.6 Gestion des clefs

L'utilisation de bi-clefs certifiés entraîne la nécessité de la publication en toute confiance de la clef publique. Cette publication doit assurer la validité de clé et l'appartenance de cette clé à la bonne personne. La publication des certificats (des clefs publiques) est faite en utilisant les structures d'annuaires de type LDAP (Lightweight Directory Access Protocol) (RFC 2251), les certificats révoqués sont regroupés dans des listes de révocations (CRL) qui sont des structures de données signées et dont le format est défini par le protocole X509 V2, ce format peut permettre une distribution des CRL via les annuaires LDAP comme Netscape Directory Server d'iplanet ou bien openldap.

2.6.7 Quelques offres commerciales d'infrastructure à clé publique

- Baltimore Technologies : Unicert
- Entrust Technologies : Entrust/PKI
- Certplus : PKI/offre initiale
- TrustyCom : TrustyKey
- Sagem : Confidence

2.6.8 PKI gratuite (OpenPKI)

- www.openssl.org : openssl

2.6.9 L'utilisation du PKI

Prenons un exemple de l'authentification mutuelle : la partie cliente d'une application présente un certificat à la partie serveur, qui en vérifie la validité auprès de l'annuaire des certificats révoqués, à l'inverse, le serveur peut lui-même envoyer un certificat au client, ce qui permet une authentification mutuelle, par exemple un web sécurisé. Une telle procédure est intéressante pour un VPN (Virtual private Network), dans le cas des équipements réseaux cette intégration est normalisée par le protocole IPSEC IKE (Internet Key Exchange). Pour les intranets, les éditeurs de PKI s'appuient sur les protocoles SSL (Secure Socket Layers) que les navigateurs et les serveurs web supportent nativement.

2.6.10 Conclusion

Une PKI est une infrastructure qui se construit, c'est donc une structure à la fois technique et administrative, avec 80% d'organisationnelle et 20% de technique. Le domaine des PKI est intéressant : il est possible de les utiliser pour des applications tels que mail chiffré, web sécurisé VPN (notamment IPSEC), commerce électronique... Et comme les PKI intègrent la cryptographie à clef publique et certificat numérique, elles peuvent se confier à des tiers de confiance et doivent recevoir l'agrément du DCSSI (Direction Central de la Sécurité des Systèmes d'Information), pour avoir une portée nationale. Actuellement l'absence de standards pour l'implantation des PKI, engendre des problèmes d'interopérabilité entre les offres du marché.

2.6.11 Lexique

- Autorité de certification : entité qui crée des certificats. C'est une autorité morale qui définit les règles d'entrée des ressources et des individus dans la PKI. En pratique, il s'agit de définir les critères et les modalités d'attribution de certificats numériques.
- Autorité d'enregistrement : entité chargée de recueillir les demandes de certificats et de contrôler l'identité de la personne ainsi que les critères d'attribution.
- Certificat : Une identité électronique qui est émise par une tierce partie de confiance pour une personne ou une entité réseau. Chaque certificat est signé avec la clé privée de signature d'une autorité de certification. Il garantit l'identité d'un individu, d'une entreprise ou d'une organisation. En particulier, il contient la clé publique de l'entité et des informations associées à cette entité.
- Certificat Auto signé : un certificat auto signé contient comme tout certificat une clé publique. Sa particularité réside dans le fait que ce certificat est signé avec la clé secrète associée. Dans ce cas précis, l'autorité de certification est donc le détenteur du certificat.
- Certificat X.509 : Il s'agit d'une norme sur les certificats largement acceptée et conçue pour supporter une gestion sécurisée et la distribution des certificats numériquement signés sur le réseau Internet sécurisé. Le certificat X.509 définit des structures de données en accord avec les procédures pour distribuer les clés publiques qui sont signées numériquement par des parties tierces.
- Certificat X.509v3 : Les certificats X.509v3 ont des extensions de structures de données pour stocker et récupérer des informations pour les applications, des informations sur les points de distribution des certificats, des CRLs et des informations sur les politiques de certification. Chaque fois qu'un certificat est utilisé, les capacités de X.509v3 permettent aux applications de vérifier la validité de ce certificat. Il permet aussi à l'application de vérifier si le certificat est dans une CRL. Ces certificats et CRLs sont normalisés auprès de l'IETF dans la RFC 2459.
- Clé : Une quantité utilisée en cryptographie pour chiffrer/déchiffrer et signer/vérifier des données.
- CRL : (Certificat Revocation List) se sont les listes de révocations de certificats.
- Clé Publique : quantité numérique, attachée à une ressource ou un individu, qui la distribue aux autres afin qu'ils puissent lui envoyer des données chiffrées ou déchiffrer sa signature.

- Clé Privée : quantité numérique secrète attachée à une ressource ou à un individu, lui permettant de déchiffrer des données chiffrées avec la clé publique correspondante ou d'apposer une signature au bas de messages envoyés vers des destinataires.
- Bi-clé : couple de clés composé d'une clé privée et d'une clé publique
- MD5 : Message Digest 5 c'est un algorithme de hachage RSA : Algorithme de chiffrement à clef publique et de signature inventé par R.Rivest, A.Shamir et l.Adleman
- DSA : Digital Signature Algorithm. AES :Advanced Encryption Standard.
- SHA-1 : Secure Hash Algorithm Number 1.SHA est un algorithme de hachage
- SSL : (Secure Socket Layer), c'est un protocole de sécurisation conçu par Netscape qui se situe entre la couche transport (TCP) et les protocoles de la couche application. Il assure les services de sécurité suivantes : confidentialité, 'intégrité et 'authentification du serveur et du client.

2.7 802.11b ou le WEP remis en cause

2.7.1 Introduction

802.11b est un protocole réseau wireless qui est actuellement de plus en plus utilisé pour les réseaux locaux (entreprises, conférences, particuliers, etc). Contrairement au protocole Bluetooth, 802.11 permet des débits élevés (11 Mbit/s dans sa version b) à de grandes distances (plusieurs centaines de mètres). Il intègre en option un protocole de sécurité au niveau liaison, le Wired Equivalent Privacy ou WEP ; celui-ci est très simple à administrer et facile à utiliser mais malheureusement peu sûr. Le but de ce document est d'en exposer les faiblesses au travers des récentes études sur le sujet.

2.7.2 Failles du protocole

Chaque périphérique 802.11 (cartes, etc) utilise une clé qui est soit un mot de passe, soit une clé dérivée de ce mot de passe. La même clé est utilisée par tous les éléments accédant au réseau, le but est donc d'interdire l'accès à toutes les personnes ne connaissant pas ce mot de passe. La faille provient de la façon dont l'algorithme de chiffrement (RC4) est implémenté et plus précisément de la façon dont sont spécifiés les vecteurs d'initialisation (IV). Certaines cartes utilisent des IVs à 0 puis les incrémentent de 1 à chaque utilisation ; cela implique nécessairement des réutilisations de vecteurs et donc des flots de données similaires (c.f. la formule du chiffrement ci-dessous). Les attaques inhérentes à ces problèmes sont très simples mais peu généralisables [3]. L'autre type d'attaques, plus efficace, a d'abord été présenté sous forme théorique par Fluhrer, Mantin et Shamir [1]. Il a récemment été implémenté très facilement, démontrant ainsi que le protocole WEP n'est pas du tout sécurisé.

2.7.3 Approche théorique

De façon très succincte, le chiffrement utilisé par WEP peut-être décrit comme suit : la clé partagée est notée K. Au moment de la transmission des données M, celles-ci sont d'abord concaténées avec leur checksum c(M). Parallèlement à cela le vecteur d'initialisation est concaténé à la clé K, et passé en entrée à la fonction de chiffrement RC4. Le résultat subit un XOR avec les données :

$$C = (M \,||\, c(M)) \text{ XOR } RC4\ (IV \,||\, K)$$

La structure du RC4 se compose de 2 parties distinctes ; la première, ou key scheduling algorithm, génère une table d'état S à partir des données secrètes, à savoir soit 64 bits (40 bits de clé secrète et 24 bits d'IV) ou 128 bits (104 bits de clé secrète et 24 bits d'IV). La deuxième partie de l'algorithme RC4 est le générateur de données en sortie, qui utilise la table S et 2 compteurs. Ces données en sortie forment une séquence pseudo-aléatoire.

Fluhrer, Mantin et Shamir présentent 2 faiblesses dans la spécification de l'algorithme RC4. La première repose sur le fait qu'il existe de larges ensembles de clés dites faibles, c'est-à-dire des clés

dont quelques bits seulement suffisent à déterminer de nombreux bits dans la table d'état S (avec une forte probabilité), ce qui affecte directement les données produites en sortie ; c'est l'attaque nommée «invariance weakness».

La deuxième attaque de Fluhrer, Mantin et Shamir est la «known IV attack ». Elle nécessite la connaissance de l'IV ce qui est le cas puisqu'il circule en clair sur le réseau, et la connaissance du premier octet de M (à deviner). Dans un certain nombre de cas (« les cas résolus », suivant l'expression de Fluhrer, Mantin et Shamir) , la connaissance de ces 2 éléments permet de déduire des informations sur la clé K.

Selon les auteurs, ces 2 attaques sont applicables et peuvent permettre une récupération complète de la clé avec une efficacité bien supérieure à l'attaque par recherche exhaustive.

2.7.4 Mise en pratique

L'implémentation de cette deuxième attaque par Stubblefield, Ioannidis et Rubin [2] a pris une semaine, requis 2h de codage et 100$ d'investissement. Leur principale difficulté a été de deviner le premier octet des données brutes (le plaintext M) ; or malgré les différents types de protocoles utilisés (notamment de l'ARP et de l'IP), il s'est avéré que 802.11 rajoute une couche supplémentaire en encapsulant tous ses paquets (header SNAP de 802.2). Ainsi, tous les paquets capturés commençaient par le même octet 0xAA. Selon les auteurs, 256 cas «résolus» suffisent pour retrouver l'intégralité de la clé de 128 bits ; ils ont également optimisé leur méthode d'attaque et ont estimé qu'un jour ou deux suffiraient à un attaquant inexpérimenté pour arriver au même résultat. Une des optimisations a consisté à tester directement des caractères simples, c'est-à-dire mémorisables par les utilisateurs. En effet, d'une part la passphrase était utilisée à l'état brut (sans hachage) dans le cas étudié, et d'autre part cette passphrase se devait d'être suffisamment simple pour être retenue par tous les utilisateurs.

2.7.5 Conclusion

En conclusion, les auteurs ont souligné les points suivants : - La couche liaison de 802.11 n'offre aucune sécurité. - Il faut utiliser des protocoles de sécurité supplémentaires tel que IPSec, SSL ou SSH et en aucun cas s'appuyer sur WEP pour assurer la sécurité. - Toutes les entités utilisant 802.11 doivent être considérées comme externes et donc placées à l'extérieur du firewall. - Il faut toujours avoir à l'esprit que toute personne se trouvant dans le rayon d'émission (et même au-delà grâce à des dispositifs amplifiants) peut être susceptible de communiquer sur le réseau en tant qu'utilisateur valide.

2.7.6 Bibliographie

- [1] FLUHRER, MANTIN et SHAMIR, Weaknesses in the key scheduling algorithm of RC4, English Annual Workshop on Selected Areas in Cryptography (08/2001).
- [2] STUBBLEFIELD, IOANNIDIS et RUBIN, Using the Fuhrer, Mantin and Shamir Attack to Break WEP, AT&T Labs Technical Report TD-4ZCPZZ (08/2001).
- [3] BORISOV, GOLDBERG et WAGNER, Intercepting mobile communications : the insecurity of 802.11, MOBICOM 2001 (2001)

2.8 Les mots de passe à usage unique

2.8.1 Qu'est-ce que c'est ?

De nombreux systèmes informatiques se basent sur une authentification faible de type login/mot de passe. Ce type d'authentification a de nombreuses contraintes :

- Un mot de passe choisi par l'utilisateur est souvent simple à retenir. Il peut donc être attaqué par dictionnaire.

- Un mot de passe "classique" a une durée de vie généralement longue (plusieurs semaines au minimum). Même si le mot de passe est correctement choisi pour être difficile à trouver, il reste néanmoins attaquable par la méthode de force brute.
- Souvent les mots de passe sont transférés en clair sur le réseau. Par un sniffer, il est possible de récupérer certains mots de passe.
- Si un mot de passe est chiffré avant d'être transmis sur le réseau, avec un sniffer, on peut récupérer le mot de passe chiffré, et faire une attaque par force brute sur le chiffré.

Les mots de passe à usage unique (one time password ou OTP en anglais) sont un système d'authentification forte basés sur le principe de challenge/réponse. Le concept est simple : Utiliser un mot de passe pour une et une seule session. De plus, le mot de passe n'est plus choisi par l'utilisateur mais généré automatiquement par une méthode de précalculé (c'est à dire que l'on précalcule un certain nombre de mot de passe qui seront utilisés ultérieurement). Cela supprime les contraintes de :

- Longévité du mot de passe. Le mot de passe est utilisé une seule fois
- Simplicité du mot de passe. Le mot de passe est calculé par l'ordinateur et non pas choisi par un utilisateur
- Attaque par dictionnaire ou par force brute : Pourquoi essayer de cracker un mot de passe obsolète ?
- Sniffer et chiffrement du mot de passe : Le mot de passe à usage unique peut être envoyé en clair sur le réseau : Lorsqu'un sniffer en détecte un, il est déjà trop tard, car il est utilisé, et non réexpoitable.

2.8.2 Comment cela fonctionne-t-il ?

D'un point de vue du serveur

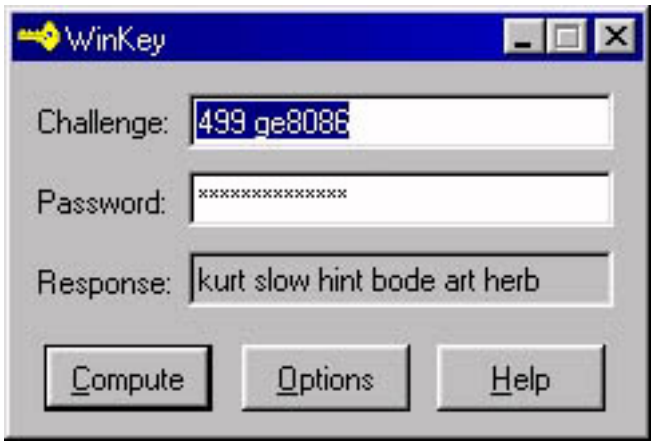
L'administrateur du serveur (c'est à dire de l'ordinateur qui va recevoir la connexion et qui va déclencher le processus d'authentification) doit utiliser un outil qui va permettre de générer un certain nombre de mot de passe à usage unique. Ensuite, il utilisera des outils adaptés à cette méthode d'authentification, par exemple il remplacera les programmes login ou su sous Unix, ftp, etc... Ces OTP générés sont calculés à partir de trois données :

- Une donnée publique (mot ou phrase courte par exemple), choisie par l'administrateur. On appèle cela aussi la semence, ou seed en anglais.
- Un numéro de séquence : C'est un nombre relatif au numéro de connexion. C'est un compteur, tout simplement. C'est ce paramètre qui rend le mot de passe unique : L'OTP de la connexion n°587 n'est pas le même que la connexion n°586 ou 588.
- Un mot de passe utilisateur, connu de lui seul, qui servira à l'authentifier.

Voici comment cela se passe : L'utilisateur exécute le programme opiepasswd. Le système lui demande son mot de passe, et ensuite il lui demande de calculer le premier challenge.

```
scrap $ /usr/local/bin/opiepasswd
Adding scrap :
You need the response from an OTP generator.
New secret pass phrase :
        otp-md5 499 ge8086
        Response :
```

L'utilisateur va alors calculer le challenge avec un logiciel :



L'utilisateur fait ensuite un copier/coller, et les OTP sont générés. Le système est opérationnel.

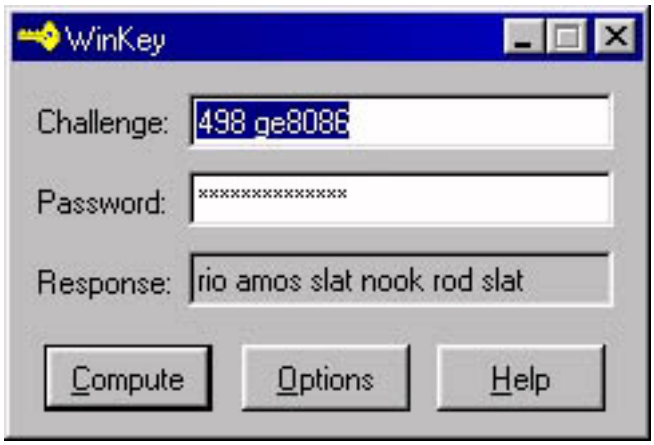
```
scrap $ /usr/local/bin/opiepasswd
Adding scrap :
You need the response from an OTP generator.
New secret pass phrase :
        otp-md5 499 ge8086
        Response :KURT SLOW HINT BODE ART HERB
ID mint OTP key is 499 ge8086
KURT SLOW HINT BODE ART HERB
scrap $
```

D'un point de vue du client

L'utilisateur va se connecter normalement sur le système distant (telnet, ftp, etc...) Ensuite, il recevra un challenge à résoudre :

```
Trying 212.43.230.175...
Connected to securiteinfo.com.
Escape character is '^]'.
Red Hat Linux release 4.2 (Biltmore)
Kernel 2.0.32 on an i386
login : scrap
        otp-md5 498 ge8086 ext
        Response :
```

L'utilisateur va ensuite utiliser un programme pour résoudre ce challenge : Il y rentre le challenge en question, et entre son mot de passe personnel.



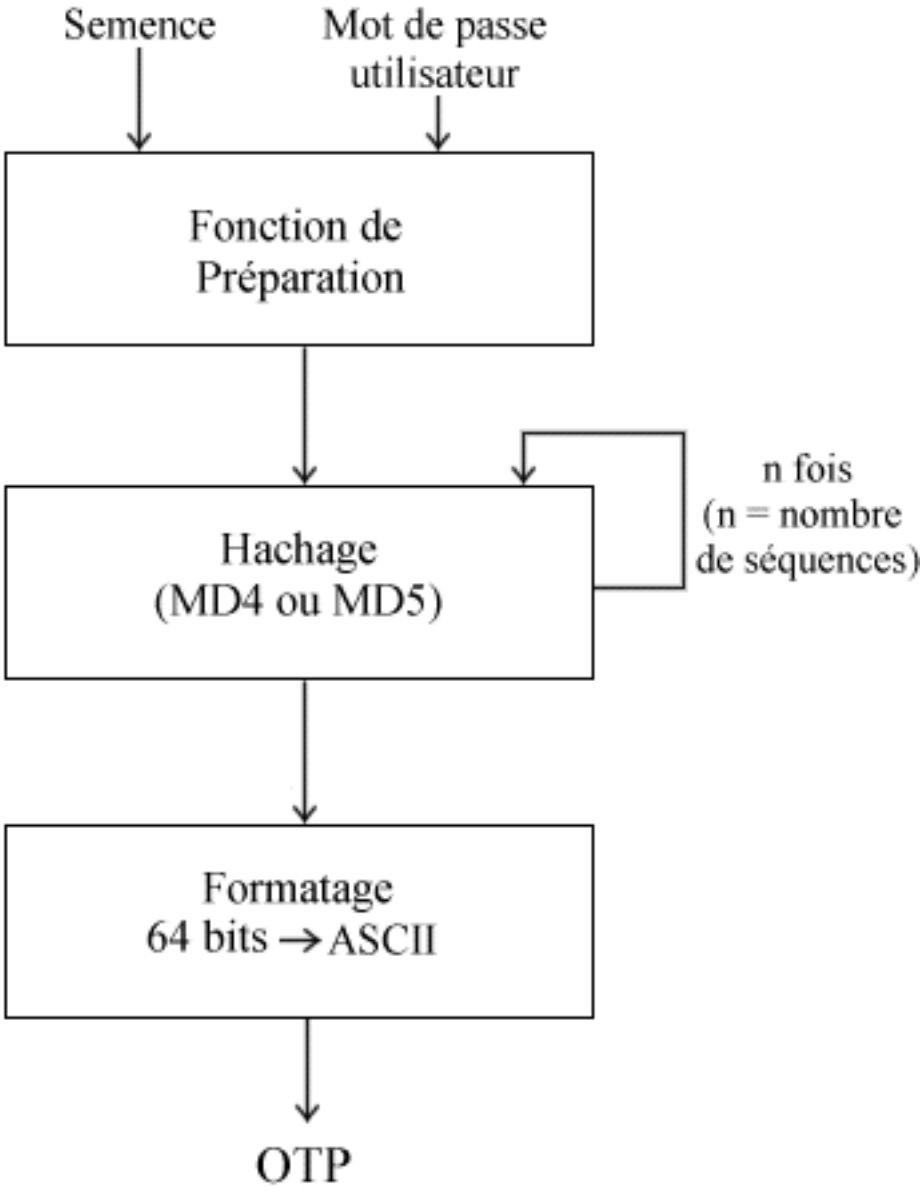
L'utilisateur n'a plus qu'à copier/coller l'OTP généré pour s'authentifier.

```
Trying 212.43.230.175...

Connected to securiteinfo.com.

Escape character is '^]'.

Red Hat Linux release 4.2 (Biltmore)

Kernel 2.0.32 on an i386

login : scrap

        otp-md5 498 ge8086 ext

        Response : RIO AMOS SLAT NOOK ROD SLAT

scrap #
```

D'un point de vue technique

Le principe des OTP repose sur une fonction de hachage sécurisée. Ce hachage ne peut être "remonté" pour revenir à l'original. Une fonction de hachage permet d'obtenir une information sur 64 bits. Généralement les mots de passe à usage uniques sont générés à partir des fonctions de hachage MD4 et MD5. Génération des OTP La génération des OTP se font en trois étapes :

- Une préparation, qui va prendre en compte toutes les données utilisateurs : la semence, et le mot de passe.
- La génération, qui consiste à appliquer la fonction de hachage n fois sur elle même, n étant le nombre de sequences.
- Le formatage du résultat, qui transforme la donnée obtenue de longueur égale à 64 bits en un mot de passe à base de caractères ASCII, c'est à dire lisible par l'homme.



Vérification des OTP : Authentification de l'utilisateur

Comme nous l'avons vu dans l'exemple utilisateur, le challenge en lui même (généré par le serveur) est composé de trois parties :

otp-md5 498 ge8086

- otp-md5 : indique que c'est un OTP calculé en MD5.
- 498 : c'est le numéro de séquence actuelle.
- ge8086 : c'est la semence.

L'utilisateur doit donc calculer la réponse au challenge grâce à un programme (par exemple WinKey). Il envoie ensuite le résultat de son calcul au serveur. Du côté serveur, le système a un fichier qui contient, pour chaque utilisateur, le dernier OTP valide utilisé pour une authentification. Sur les systèmes Unix, ce fichier est /etc/skeykeys. Pour vérifier la validité de l'OTP utilisateur, le serveur a une astuce : Le numéro de séquence (498 dans notre exemple) est décrémenté à chaque authentification valide. Aussi, si le client calcule pour un numéro d'itération n, le serveur, lui, a n+1 stocké dans son fichier. Dans notre exemple, le serveur a donc stocké l'OTP correspondant au

numéro de séquence 499. Le système n'a donc plus qu'une tâche à effectuer : Calculer le hachage de l'OTP que l'utilisateur vient de lui envoyer. Si le résultat est égal au résultat n+1 stocké dans le fichier, l'utilisateur est authentifié.

2.8.3 Les failles des mots de passe à usage unique

Malgré tout, les mots de passe à usage unique comportent des failles exploitables. Elles sont présentées par ordre croissant de difficulté pour l'attaquant :

Social engineering, trashing, etc...

L'utilisateur, l'éternelle faiblesse. Quelquefois, il est possible de récupérer le mot de passe de l'utilisateur en regardant sous le clavier, dans ses tiroirs, ou même en fouillant ses poubelles... Une fois que vous avez le mot de passe, vous pouvez répondre au challenge... Une autre technique : Certains programmes, comme WinKey, comportent un trou de sécurité : Tant que le programme de calcul n'est pas fermé, celui-ci garde en mémoire le mot de passe, visible sous forme d'étoiles. (voir photos d'ecrans ci-dessus). Il existent des programmes dans l'environnement Win32 qui permettent de rendre clair le contenu des étoiles. Cinq minutes d'inattention suffisent...

Attaque par cheval de Troie et keylogger

Il "suffit" de mettre un cheval de Troie ou un keylogger sur le poste de l'utilisateur victime, pour essayer de récupérer le mot de passe de celui-ci.

Le fichier skeykeys

Si le hacker a la main sur le serveur, il est possible qu'il exploite une faille : Si le fichier /etc/skeykeys est disponible en lecture pour le hacker, voici ce qu'il y trouve :

login	numéro de séquence	semence	Représentation hexa de la réponse du challenge (le dernier OTP utilisé)	Date	Heure
victime	985	xv 5235b	5b3c89552aa09435	Jun 25, 2001	05 :30 :15

L'attaquant récupère alors : Le login et la représentation hexa de l'OTP. Avec cette représentation, il peut créer un outils qui générera la réponse au challenge en fonction d'un dictionnaire, ou par force brute (voir ci dessous). Pour contrer cette attaque, vérifiez que ce fichier n'est pas en lecture pour tout le monde (encore moins en écriture !).

Attaque par sniffing

L'OTP est envoyé vers le serveur en clair. Il est donc possible de récupérer la réponse du challenge et de faire une attaque par dictionnaire ou par force brute (voir ci-dessous).

Attaque par dictionnaire

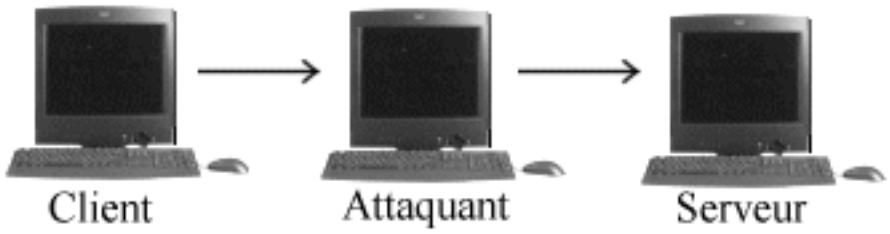
L'attaque par dictionnaire permet de calculer la réponse au challenge en fonction d'un nombre de mots préalablement stockés dans un gros fichier. Ces mots serons utilisés comme mot de passe de l'utilisateur. Si le résultat du calcul est identique à la réponse d'un challenge passé, alors c'est que le mot utilisé dans le dictionnaire est le mot de passe de l'utilisateur.

Attaque par force brute

Exactement le même principe que pour l'attaque par dictionnaire. C'est plus long, mais cela reste toujours possible...

Attaque par le milieu

Cette attaque consiste pour l'attaquant à avoir la main sur une machine qui transmet l'authentification de la session. C'est une machine intermédiaire. Dans ce cas, l'attaquant peut écouter les trames qui transitent et repérer le challenge et la réponse du challenge. Là aussi il devra utiliser le cracking par dictionnaire ou par brute force. Cette méthode revient à du sniffing.



Attaque par spoofing

La technique du spoofing, dans ce cas, est de se faire passer pour le serveur. C'est une attaque par le milieu (voir ci-dessus), mais un peu plus évolué. En effet, il faut en plus simuler exactement le comportement du serveur. On récupère ainsi le compteur (numéro de séquence) et la réponse au challenge Il est alors possible d'exploiter cela par deux techniques différentes :

- Soit essayer de trouver le mot de passe utilisateur par dictionnaire et brute force
- Soit l'attaquant a utilisé sciemment un numéro de séquence inférieur à ce qu'attend vraiment le serveur. Dans ce cas, l'attaquant peut alors utilisé un nombre de connexion, au détriment de l'utilisateur, égal à la différence entre le vrai numéro de séquence serveur, et le faux numéro de séquence de l'attaquant.

Cette attaque est facilement décelable pour l'utilisateur victime, car il se rend compte que le serveur ne fonctionne pas correctement. Par contre, pour l'attaquant, les données récupérées sont directement exploitables. Cette attaque est donc à prendre très au serieux, d'autant plus qu'elle n'est pas très difficile à réaliser.

Attaque par faille temps réel

Il se peut qu'il y ait un problème dans l'implémentation d'un OTP. Il est possible que, lorsque deux ouverture de session arrivent en même temps sur le serveur, celui-ci ne sache pas comment les gérer. C'est un problème purement temps réel. Il y a alors trois cas de figures : Le serveur rejette les deux connexions. L'attaquant n'a aucune chance d'avoir accès. L'utilisateur non plus ! Ce cas est peu probable. Le serveur accepte une connexion (la première arrivée en général), et refuse la seconde. Dans ce cas, l'attaquant a 50% de chance de réussir. Ce cas est probable par l'utilisation d'une sémaphore. Le serveur accepte les deux connexions. L'attaquant a 100% de chance de réussir. Ce cas est peu probable.

Attaque par Hi-Jacking

L'authentification se fait au moment de la connexion. Passé cette authentification, il n'y a plus d'autres moyens de vérifier si l'authentification est toujours correcte. Aussi, il est possible de détourner la communication TCP/IP. C'est une attaque de type Hi-jacking. Une fois le détournement effectué, le serveur dialogue avec l'attaquant,et l'attaquant utilise la session de la victime.

2.8.4 Conclusion

Le maillon faible des OTP est le mot de passe utilisateur. Comme on l'a vu, il y a divers moyens pour le récupérer. Les mots de passe à usage uniques ne sont donc pas fiables à 100% Ceci dit, cette méthode est tout de même beaucoup plus sécurisée que la méthode traditionnelle

login/password. De plus il est possible de l'implémenter sans avoir de coûts excessifs. Il faudrat néanmoins penser à former les utilisateurs, qui n'ont pas l'habitude de manipuler un logiciel de chiffrement pour se loguer :)

2.8.5 Références

- La RFC 1760 : The S/KEY One-Time Password System
- La RFC 1321 : The MD5 Message-Digest Algorithm
- La RFC 1320 : The MD4 Message-Digest Algorithm

2.9 PGP : Chiffrement de données

> "C'est personnel. C'est privé. Et cela ne regarde personne d'autre que vous. Vous pouvez être en train de préparer une campagne électorale, de discuter de vos impôts, ou d'avoir une romance secrète. Ou vous pouvez être en train de communiquer avec un dissident politique dans un pays répressif. Quoi qu'il en soit, vous ne voulez pas que votre courrier électronique (e-mail) ou vos documents confidentiels soient lus par quelqu'un d'autre. Il n'y a rien de mal à défendre votre vie privée. La confidentialité est aussi fondamentale que la Constitution." - Phil Zimmermann (Inventeur de PGP).

2.9.1 Qu'est-ce que c'est ?

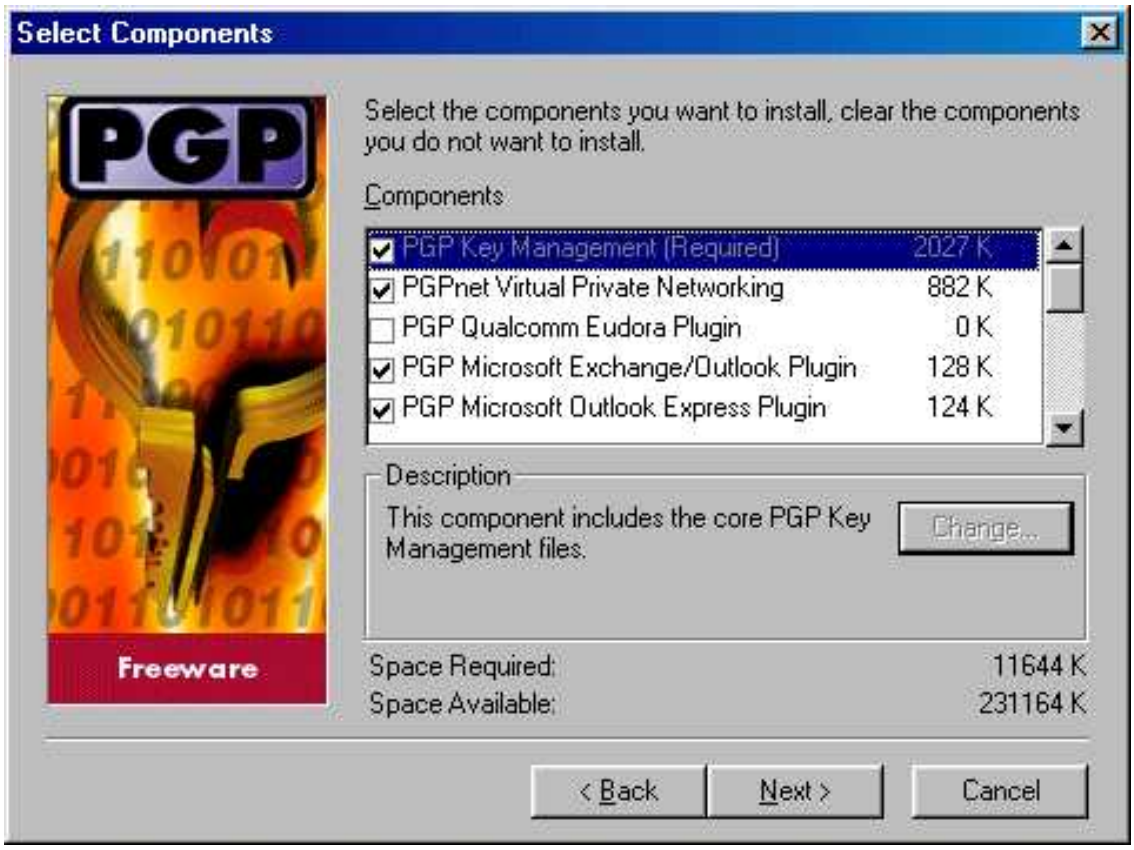
PGP de PRETTY GOOD PRIVACY est le logiciel le plus connu dans le domaine de la protection des données sur Internet, il est souvent utilisé pour protéger le courrier électronique et il est très facile d'utilisation. Cette fiche va vous apprendre en quelques étapes à installer et à utiliser PGP afin de sécuriser vos données circulant sur Internet ou tout autre document que vous voulez garder secret.

2.9.2 Etape 1 : Installation et paramétrages

PGP est gratuit, il est proposé en libre téléchargement sur de nombreux sites (www.telecharger.com) pour la version Windows. Il est disponible aussi sur les autres plates-formes, BeOS, Linux, MacOS, (voir le site de l'éditeur : http ://web.mit.edu/network/pgp.html) et bien sur le site officiel.

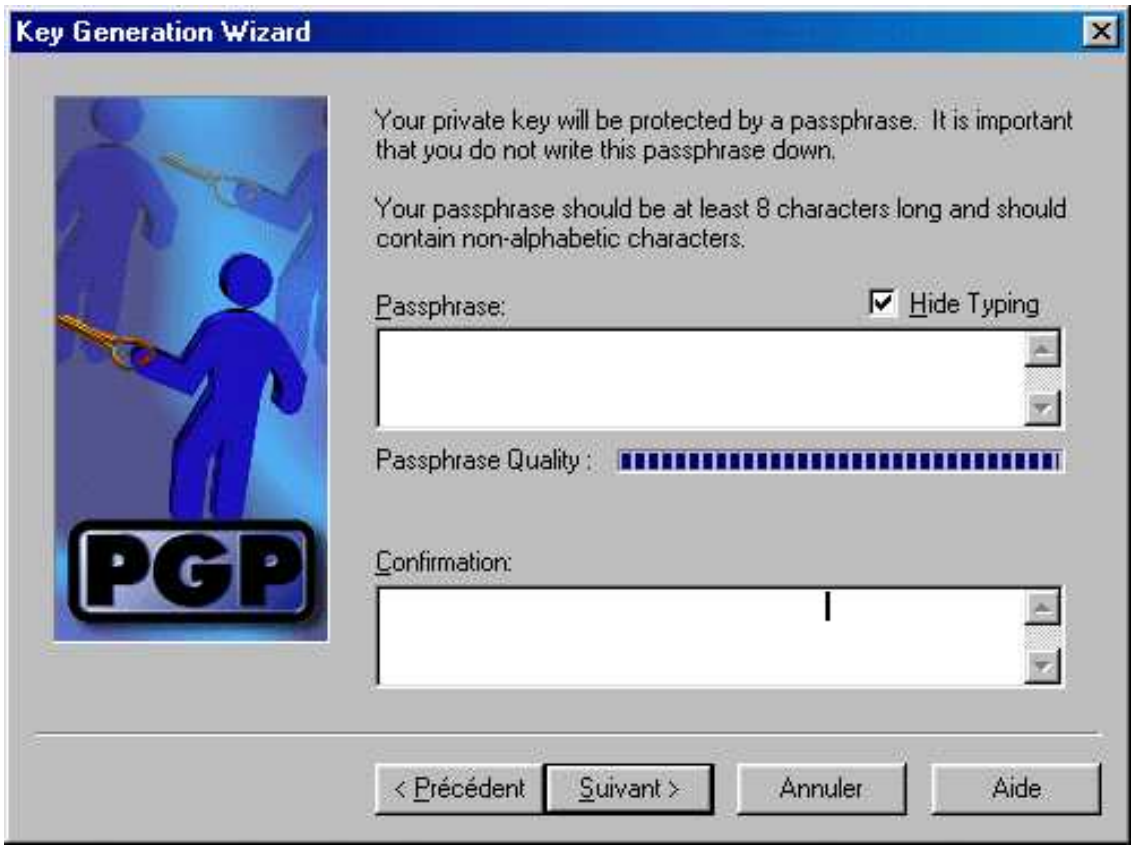
Enregistrement

Lors de l'installation, PGP commence par vous demander des renseignements sur l'utilisateur : Nom, E-Mail. Puis il demande de choisir les composants à installer en fonction du client de messagerie électronique, le plus simple est de tous les installer comme cela vous pouvez changer à tout moment de client sans être inquiété de la compatibilité.



PGPKeys

L'installation continue et vous demande si vous voulez lancer PGPKeys. Une fois lancé, PGPKeys va d'abord créer une nouvelle paire de clefs, la clef privée et la clef publique, si toutefois l'assistant de création ne se lance pas tout seul, cliquez sur l'icône de la barre d'outils "Générer une nouvelle paire de clef", puis entrez votre Nom et votre E-Mail. Choisissez "Diffie-Hellman/DSS" en type de clef et 2048 bits pour la longueur de clef. Une autre option s'offre à vous, vous pouvez déterminer une date d'expiration pour votre paire de clef (ce n'est pas obligatoire). Pour finir PGPKeys vous demande de rentrer une phrase secrète, qui sert à générer votre clef privée. Une barre indicatrice,vous permet de mesurer la complexité de votre phrase secrète, le mieux étant d'y introduire le plus de caractères spéciaux ou accentués possible. Attention, mémorisez impérativement cette phrase et sa syntaxe, elle est indipensable pour se servir de PGP. Ensuite vous pouvez choisir d'envoyer votre clef tout de suite ou plus tard au serveur.



PGP vous permet de sauvegarder les clefs créées sous 2 fichiers, secring.skr (clef privée) et pubring.pkr (clef publique). Pratique dans le cas d'une panne système ou d'un formatage. Toujours dans un soucis de sécurité, il est très important de ne pas garder ces fichiers sur le disque dur car si quelqu'un recupère votre fichier, il peut le brute forcer pour obtenir votre phrase secrète, s'il réussi, il pourra déchiffrer tous vos documents.

Et voila l'installation est terminée.

2.9.3 Etape 2 : Clef privée et publique, quelle est la différence ?

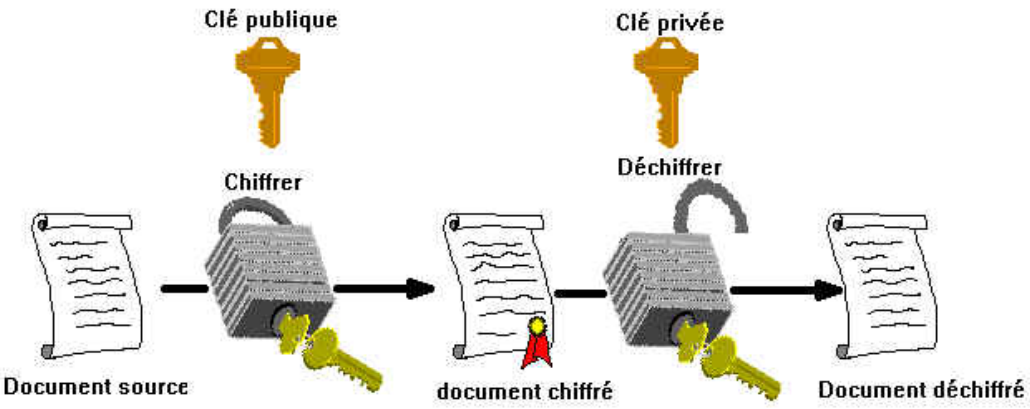
Avant d'utiliser PGP, il faut comprendre la différence entre la clef publique et la clef privée.

Clef privée

La clef privée sert uniquement à déchiffrer les données qui ont été chiffrées avec votre clef publique.

Clef publique

La clef publique sert uniquement à chiffrer les données et à vous identifier auprès du destinataire.

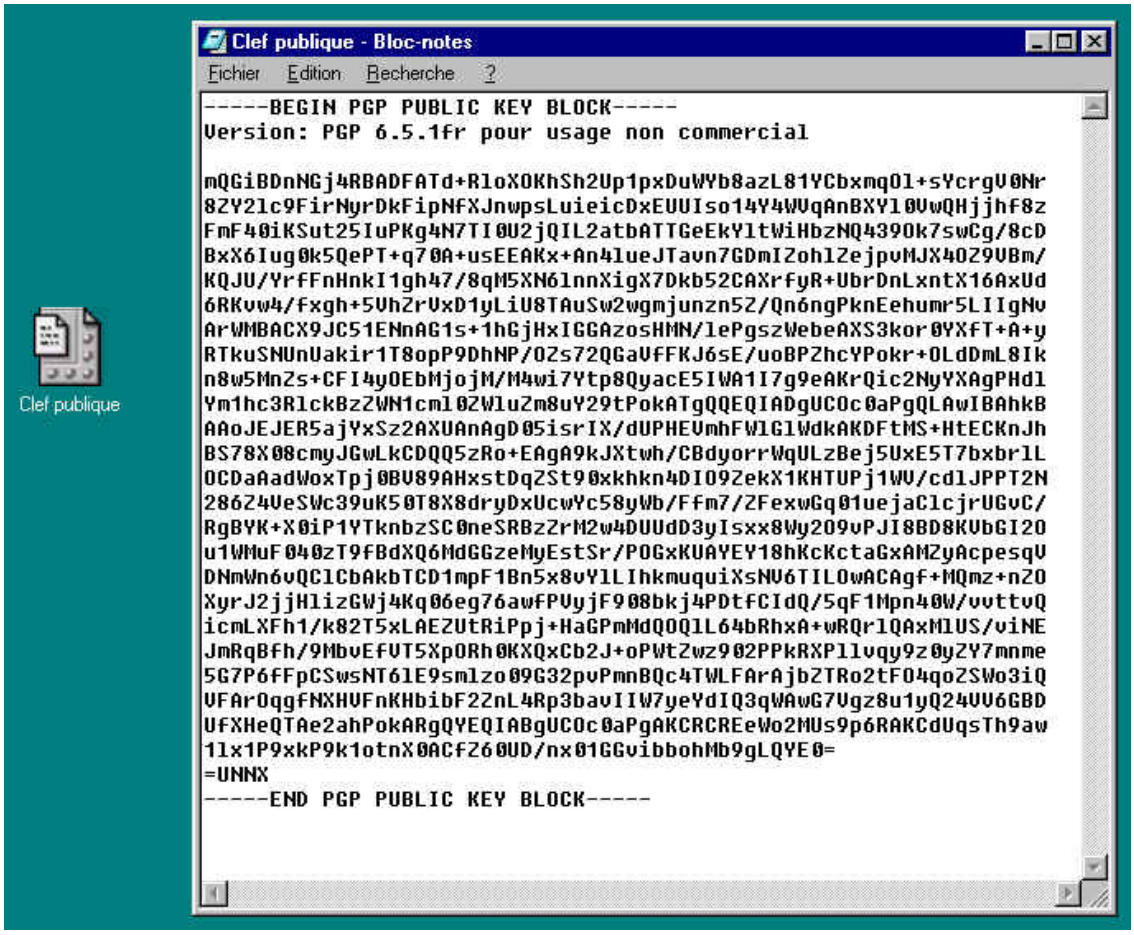


2.9.4 Etape 3 : Exporter et Importer une clef publique

Pour pouvoir chiffrer un message, votre correspondant doit posséder PGP et votre clef publique, pour cela il est nécessaire d'exporter votre clef publique et par conséquent votre correspondant doit l'importer vers son PGP, il en va de même pour vous, vous devrez importer la clef publique de votre correspondant. Mais sachez néamoins que plusieurs solutions sont possibles, la clef publique du destinataire n'est pas indispensable, mais nous verrons cela plus loin.

Exporter

Deux méthodes sont possibles. La première méthode consiste à sélectionner votre clef et à la copier grâce au menu édition, pour ensuite la coller dans un fichier texte sous Word par exemple ou tout simplement dans le mail destiné à votre correspondant. La seconde consiste à créer un fichier ASC grâce à la fonction EXPORTER du menu Clés de PGPKeys, ce fichier comporte toutes les informations relatives à votre clef publique, ensuite il suffit de le joindre à un mail pour l'envoyer à votre correspondant.



Importer

Quand votre correspondant a reçu votre clef publique, il ne lui reste plus qu'à l'importer dans PGP. Pour cela, il doit activer la commande IMPORTER du menu Clés de PGPKeys. PGP demande alors d'indiquer le fichier contenant la clef publique, puis une nouvelle fenêtre s'ouvre, et il ne reste plus qu'à cliquer sur le bouton Importer. A partir de maintenant votre correspondant peut utiliser votre clef publique pour chiffrer des documents que vous serez le seul à pouvoir déchiffrer.

La signature

Comme nous l'avons dit plus haut, nous pouvons identifier la provenance des données chiffrées grâce à la clef publique. Lorsque vous chiffrez vos données, si vous désirez insérer votre signature, PGP vous demandera d'entrer votre Phrase Secrète, le chiffrement de la signature ce faisant automatiquement à partir de votre clef privée. PGP assure ainsi que personne d'autre ne puisse se faire passer pour vous.

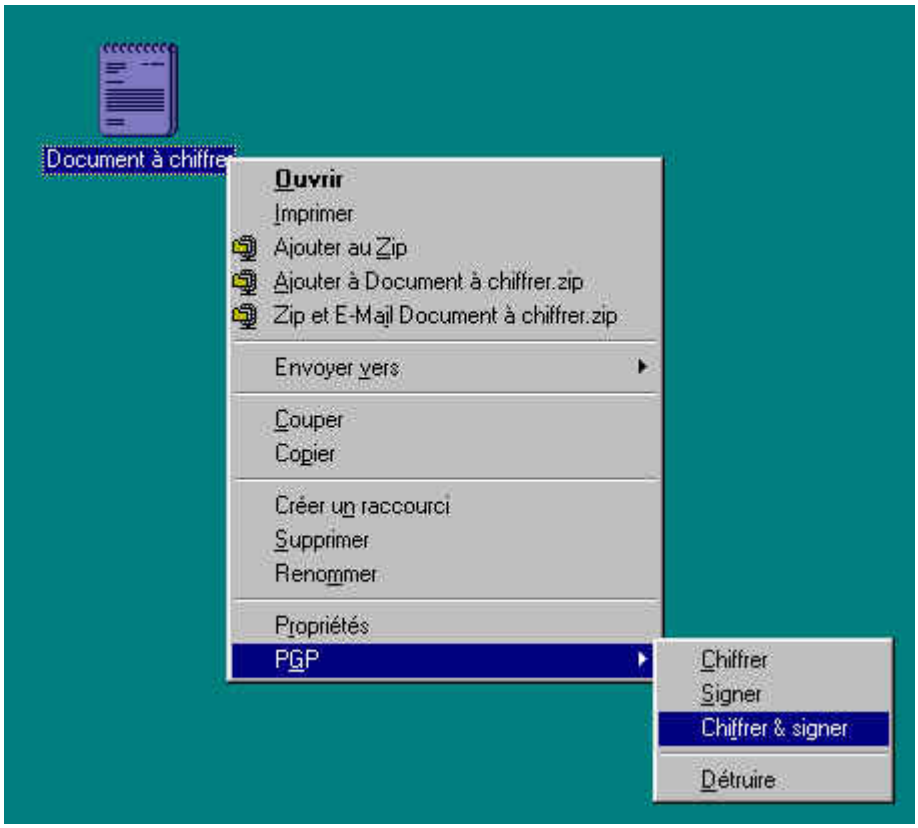
2.9.5 Etape 4 : Chiffrer et déchiffrer des données

Ceci est l'étape la plus importante car nous allons voir comment chiffrer et déchiffrer des données avec PGP.

Chiffrer

Pour chiffrer des données, ouvrez PGPTools. Différentes méthodes vous sont accessibles, vous pouvez chiffrer vos données, seulement les signer ou encore les chiffrer et les signer. Privilégiez la dernière méthode ainsi votre correspondant recevra vos données chiffrées avec l'assurance qu'elles viennent bien de vous. Pour cela cliquez sur le 4ème bouton, PGP vous demande alors d'indiquer le chemin du Fichier que vous voulez chiffrer et signer.

Une fenêtre s'ouvre et vous permet de sélectionner la clef publique du destinataire (ou même de plusieurs destinataires) et c'est ici que l'on peut choisir son type de chiffrement. Comme il a été dit plus haut la clef publique n'est pas indispensable. Vous pouvez chiffrer votre document par clef publique, c'est à dire que vous utilisez la clef publique de votre correspondant pour chiffrer le Document. Par conséquent il devra utiliser sa clef privée pour le déchiffrer. Ou vous pouvez utiliser le Chiffrement Conventionnel. Cette méthode chiffre le document en fonction d'une phrase qui sert de clef de chiffrement (n'inscrivez pas votre Phrase Secrète ! ! !). C'est cette phrase que votre correspondant devra posséder pour pouvoir déchiffrer vos données. Une fois effectuée la sélection du type de chiffrement vous pouvez aussi opter pour l'effacement automatique du document original dès le chiffrement terminé. Pour finir, si vous avez demandé à signer votre document, PGP vous demande de saisir votre Phrase Secrète (celle entrée lors de la création des clefs), afin de vérifier que c'est le bon interlocuteur qui utilise les clefs et que personne ne se fait passer pour vous. Une fois cela fini, PGP chiffre automatiquement le fichier et l'enregistre au même endroit que l'original. Mais il n'est pas indispensable de passer par PGP pour chiffrer un document vous pouvez aussi le faire à partir du menu contextuel en cliquant sur le document avec le bouton droit de la souris. Là, vous trouverez des commandes qui permettent d'utiliser directement PGP sur le document.

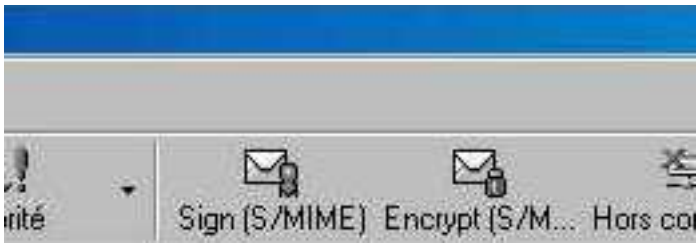


Déchiffrer

Toujours à partir de PGPTools, cliquez sur le cinquième bouton "Déchiffrer & Vérifier" et indiquez l'endroit où se trouve le fichier à déchiffrer. Si le document a été chiffré avec le chiffrement conventionnel, PGP vous demande alors de saisir la phrase qui sert de clef de chiffrement, puis une fois qu'elle est reconnue PGP vous demande l'endroit où il doit sauvegarder le document déchiffré, dans la cas d'un chiffrement par clef publique, le déchiffrement s'effectue directement et comme pour l'autre cas, PGP vous demande où il doit sauvegarder le document.

2.9.6 Etape 5 : Intégration de PGP dans un client de messagerie

L'un des principaux usages de PGP étant de sécuriser le contenu de vos messages échangés sur Internet, il est évident de voir PGP s'intégrer dans votre Client de Messagerie Electronique. PGP est compatible avec tous les clients sauf Netscape Messenger, et donc automatiquement lors de l'installation lorsque vous avez sélectionné les clients de messagerie, un icône est apparut dans la barre d'outils de votre logiciel, ce qui vous permet d'ouvrir rapidement les outils de PGP. Il s'intègre aussi facilement dans les nouveaux messages, ainsi après avoir rédigé votre message comme d'habitude, avant de l'envoyer vous pouvez cliquer sur le bouton "Chiffrer & Signer", un cadenas et un cachet s'affichent à côté des champs A et Cc. Il ne vous reste plus qu'à cliquer sur envoyer et PGP s'ouvre pour Chiffrer votre mail de la même façon que décrite plus haut. Une fois terminé, votre mail est codé et incompréhensible pour les personnes qui veulent le lire sans déchiffrement préalable.



2.9.7 Conclusion

PGP est très puissant, gratuit et très simple d'utilisation. C'est l'outil indispensable dans la quête d'intimité sur internet et chez soi.

Chapitre 3

Les conseils

3.1 La Sécurité Informatique

3.1.1 Introduction à la sécurité informatique

Le but de cette fiche est de familiariser les lecteurs avec les notions de base de la sécurité informatique, telles que définies dans la norme ISO 7498-2 par exemple.

3.1.2 Les objectifs de la sécurité informatique

La sécurité informatique à plusieurs objectifs, bien sûr liés aux types de menaces ainsi qu'aux types de ressources, etc... Néanmoins, les points principaux points sont les suivants :

- empêcher la divulgation non-autorisée de données
- empêcher la modification non-autorisée de données
- empêcher l'utilisation non-autorisée de ressources réseau ou informatiques de façon générale

3.1.3 Les champs d'application de la sécurité informatique

Ces objectifs s'appliquent dans différents domaines ou champs d'applications, chacun faisant appel à des techniques différentes pour atteindre le ou les mêmes objectifs ; ces champs sont :

- la sécurité physique
- la sécurité personnelle
- la sécurité procédurale (audits de sécurité, procédures informatiques...)
- la sécurité des émissions physiques (écrans, cables d'alimentation, courbes de consommation de courant...)
- la sécurité des systèmes d'exploitation la sécurité des communications

3.1.4 Terminologie de la sécurité informatique

La sécurité informatique utilise un vocabulaire bien défini que nous utilisons dans nos articles. De manière à bien comprendre ces articles, il est nécessaire de définir certains termes :

- Les vulnérabilités : ce sont les failles de sécurité dans un ou plusieurs systèmes. Tout système vu dans sa globalité présente des vulnérabilités, qui peuvent être exploitables ou non.
- Les attaques (exploits) : elles représentent les moyens d'exploiter une vulnérabilité. Il peut y avoir plusieurs attaques pour une même vulnérabilités mais toutes les vulnérabilités ne sont pas exploitables.
- Les contre-mesures : ce sont les procédures ou techniques permettant de résoudre une vulnérabilité ou de contrer une attaque spécifique (auquel cas il peut exister d'autres attaques sur la même vulnérabilité).

- Les menaces : ce sont des adversaires déterminés capables de monter une attaque exploitant une vulnérabilité.

Pour d'autres définitions, consultez la norme ISO 7498-2 qui définit pas moins de 59 termes ; d'autres définitions sont également disponibles dans notre lexique.

3.1.5 Types d'attaques

Les attaques peuvent à première vue être classées en 2 grandes catégories :

- les attaques passives : consistent à écouter sans modifier les données ou le fonctionnement du réseau. Elles sont généralement indétectables mais une prévention est possible.
- les attaques actives : consistent à modifier des données ou des messages, à s'introduire dans des équipements réseau ou à perturber le bon fonctionnement de ce réseau. Noter qu'une attaque active peut être exécutée sans la capacité d'écoute. De plus, il n'y a généralement pas de prévention possible pour ces attaques, bien qu'elles soient détectables (permettant ainsi une réponse adéquate).

3.1.6 Profils et capacités des attaquants

Les attaquants peuvent être classés non-seulement par leur connaissances (newbies, experts, etc...) mais également suivant leurs capacités d'attaques dan sune situation bien définie. Ainsi, on dénombrera les capacités suivantes :

transmission de messages sans capacité d'écoute (IP spoofing...)

- écoute et transmission de messages
- écoute et perturbation des communications (blocage de paquets, DoS et DDoS...)
- écoute, perturbation et transmissions de messages
- écoute et relai de messages (attaques type man-in-the-middle)

Une autre caractéristique des attaquants va être leur emprise uni-directionnelle ou bi-directionnelle sur les communications, du fait de la nature asymétrique de celles-ci. En effet, la plupart des canaux de transmissions sur Internet ou sur tout autre réseau hétérogène sont uni-directionnels et empruntent des chemins différents suivant les règles de routage. Ainsi, de nombreux protocoles de sécurité sont également unidirectionnels et il faut établir plusieurs canaux pour permettre un échange en "duplex". Ces canaux qui sont au nombre de 2 minimum, sont la plupart du temps gérés de façon totalement indépendante par les protocole de sécurité. C'est le cas pour SSL/TLS mais également pour IPSec dont les associations de sécurité (SA) sont unidirectionnelles et indépendantes, chacune définissant sont propre jeu de clés, algorithmes, etc...

3.1.7 Services principaux de la sécurité informatique

Pour remédier aux failles et pour contrer les attaques, la sécurité informatique se base sur un certain nombre de services qui permettent de mettre en place une réponse appropriée à chaque menace. A ce niveau, aucune technique n'est encore envisagée ; il ne s'agit que d'un niveau d'abstraction visant à obtenir une granularité minimale pour déployer une politique de sécurité de façon optimale (les aspects pratiques tels qu'analyses de risques, solutions technologiques et coûts viendront par la suite. Voir le "Site Security Handbook", RFC 1244 pour plus de détails). Décrivons les principaux services de sécurité :

- confidentialité
- authentification (entité, origine des données)
- intégrité
- machines (tamper-résistance, tamper-proofness, exécution sécurisée...)
- données (avec possibilité de récupération)
- flux :
- mode non-connecté, au niveau paquet (échanges de type requête-réponse, comme UDP)
- mode orienté-connexion (ensemble de l'échange, comme TCP)
- intégrité de séquences partielles (VoIP, applications, etc... permet d'éviter les DoS par exemple)

- contrôle d'accès (= autorisation, à différentier de l'authentification)
- non-répudiation (avec preuve d'émission ou avec preuve de réception)

Notez que le chiffrement, les signatures digitales et autres techniques correspondent au niveau d'abstraction inférieur, décrit comme l'ensemble des méchanismes de sécurité permettant de réaliser les services décrits ci-dessus. Plusieurs méchanismes peuvent par exemple réaliser le service d'authentification (schémas d'authentification, chiffrement, signatures digitales...). Néanmoins, ces méchanismes de sécurité ne correspondent pas encore aux solutions finales qui seront réellement implémentées. Il faudra pour cela effectuer un dernier raffinement, consistant à choisir les algorithmes symétriques, les algorithmes asymétriques, la tailles des clés, etc...

Enfin, il existe d'autres notions qui ne peuvent être classées directement dans ces listes ; la confiance (trust) est un bon exemple. En effet, bien qu'elle soit très couteuse, la confiance est obligatoire pour garantir l'efficacité des méchanismes de sécurité mis en place. Citons l'exemple d'un protocole d'encapsulation chiffrée (tunneling), développé en soft, permettant d'échanger des données tout en préservant leur confidentialité. Or, si seules les données sont protégées, il est plus simple pour un attaquant de s'introduire dans l'une des machines aux extrémités (PC ou autres), de modifier les librairies correspondantes de façon à fausser le méchanisme de sécurité (nombres aléatoires forcés à une valeur constante, valeurs de clés prédéfinies, algorithmes NULL) et enfin de pouvoir accéder à loisir aux données transmises. D'où la nécessité de mettre en place un schéma de confiance visant à interdire ce type d'attaque ; il est nécessaire de pouvoir faire confiance aux équipements de sécurité car dans le cas contraire, l'utilité même des méchanismes de sécurité est remise en question.

3.2 La Biométrie

3.2.1 Introduction

La biométrie est une technique globale visant à établir l'identité d'une personne en mesurant une de ses caractéristiques physiques. Il peut y avoir plusieurs types de caractéristiques physiques, les unes plus fiables que d'autres, mais toutes doivent être infalsifiables et uniques pour pouvoir être représentatives d'un et un seul individu. D'autre part, comme nous allons le voir, les caractéristiques physiques sont loins d'être si parfaites et si précises, et l'on atteint très vite des limites pour ces techniques.

3.2.2 Usage

Les techniques basées sur la biométrie jouissent à l'heure actuelle d'un engouement général favorisé par un phénomène de mode, principalement véhiculé par les films au cinéma et à la télévision. Ainsi, il n'est pas rare de voir des scanners rétiniens avec de superbes lasers rouges, des lecteurs d'empreintes digitales avec de très jolis voyants -clignotants-, etc... tout cela représentant le summum de la technologie du contrôle d'accès. Or, les techniques de biométrie sont belle et bien en train de se répandre dans notre vie quotidienne, et ce tout en gardant une image quelque peu trompeuse. Car le problème est bien de savoir quelles techniques existent réellement, et quelles sont leurs limites. Cet article ne se veut pas exhaustif sur un sujet aussi vaste que la biométrie, mais il a tout de même pour vocation de sensibiliser au maximum les lecteurs et de leur donner quelques bases indispensables.

3.2.3 Caractéristiques physiques

Il existe plusieurs caractéristiques physiques qui se révèlent être uniques pour un individu, et il existe également pour chacune d'entre elles plusieurs façons de les mesurer :

empreintes digitales (finger-scan)

la donnée de base dans le cas des empreintes digitales est le dessin représenté par les crêtes et sillons de l'épiderme. Ce dessin est unique et différent pour chaque individu. En pratique, il est quasiment impossible d'utiliser toutes les informations fournies par ce dessin (car trop nombreuses pour chaque individu), on préferera donc en extraire les caractéristiques principales telles que les bifurcations de crêtes, les "îles", les lignes qui disparaissent, etc... Une empreinte complète contient en moyenne une centaine de ces points caractéristiques (les "minuties"). Si l'on considère la zone réellement scannée, on peut extraire environ 40 de ces points. Pourtant, là encore, les produits proposés sur le marché ne se basent que sur une quinzaine de ces points (12 au minimum vis-à-vis de la loi), voire moins pour beaucoup d'entre eux (jusqu'à 8 minimum). Pour l'histoire, le nombre 12 provient de la règle des 12 points selon laquelle il est statistiquement impossible de trouver 2 individus présentants les mêmes 12 points caractéristiques, même en considérant une population de plusieurs dizaines de millions de personnes.

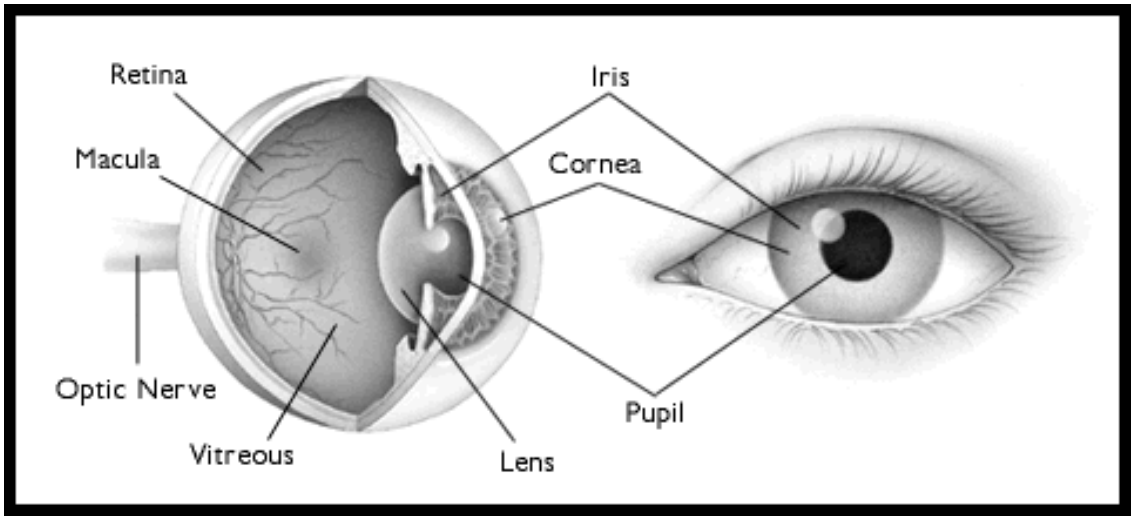
Les techniques utilisées pour la mesure sont diverses : capteurs optiques (caméras CCD/CMOS), capteurs ultrasoniques, capteurs de champ électrique, de capacité, de température... Ces capteurs sont souvent doublés d'une mesure visant à établir la validité de l'échantillon soumis (autrement dit, qu'il s'agit bien d'un doigt) : mesure de la constante diélectrique relative du l'échantillon, sa conductivité, les battements de coeur, la pression sanguine, voire une mesure de l'empreinte sous l'épiderme...

géométrie de la main / du doigt (hand-scan)

ce type de mesure biométrique est l'un des plus répandus, notament aux Etats Unis. Cela consiste à mesurer plusieurs caractéristiques de la main (jusqu'à 90) tel que la forme de la main, longueur et largeur des doigts, formes des articulations, longueurs inter-articulations, etc... La technologie associée à cela est principalement de l'imagerie infrarouge ; d'une façon générale, le système présente des FAR (False Acceptation Rate, voir plus bas) assez élevés, surtout entre personnes de la même famille ou bien encore des jumeaux.

iris (iris-scan)

pour les 2 techniques suivantes, il faut tout d'abord faire la distinction entre l'iris et la rétine :



Source : American Academy of Ophthalmology

Autrement dit, l'étude de l'iris va se porter sur la partie de l'oeil visible ci-dessous :

En ce qui concerne l'iris, l'individu se place en face du capteur (caméra CCD/CMOS) qui scanne son iris. Celui-ci représente quelquechose de très intéressant pour la biométrie car il est à la fois toujours différent (même entre jumeaux, entre l'oeil gauche et le droit, etc...), indépendant du code génétique de l'individu, et très difficilement falsifiable. En effet, l'iris présente une quasi-infinité de points caractéristiques (que certains comparent en nombre à ceux de l'ADN), qui ne varient pratiquement pas pendant la vie d'une personne contrairement à la couleur de l'iris qui, elle, peut changer. Mais cela n'a aucune influence car les images d'iris obtenues par les capteurs sont en noir et blanc. Le seul problème de cette technique est liée à la mesure en elle-même, qui peut être source d'erreurs ou de problèmes. Ainsi, on peut quasiment dire que le nombre de problèmes rencontrés lors de cette mesure augmente proportionnellement avec la distance entre l'oeil et la caméra. D'autres problèmes se posent à cause des reflets (nécessité d'avoir un éclairage restreint et maitrisé), et lors de la détection de faux yeux (photos) et autres fraudes. Pour ces dernières, on peut faire appel à certaines caractéristiques dynamiques de l'oeil qui prouveront son authenticité : réactivité de la pupille (dilatation/rétraction) par rapport à la quantité de lumière, étude de l'iris dans l'infrarouge et l'ultraviolet, etc...

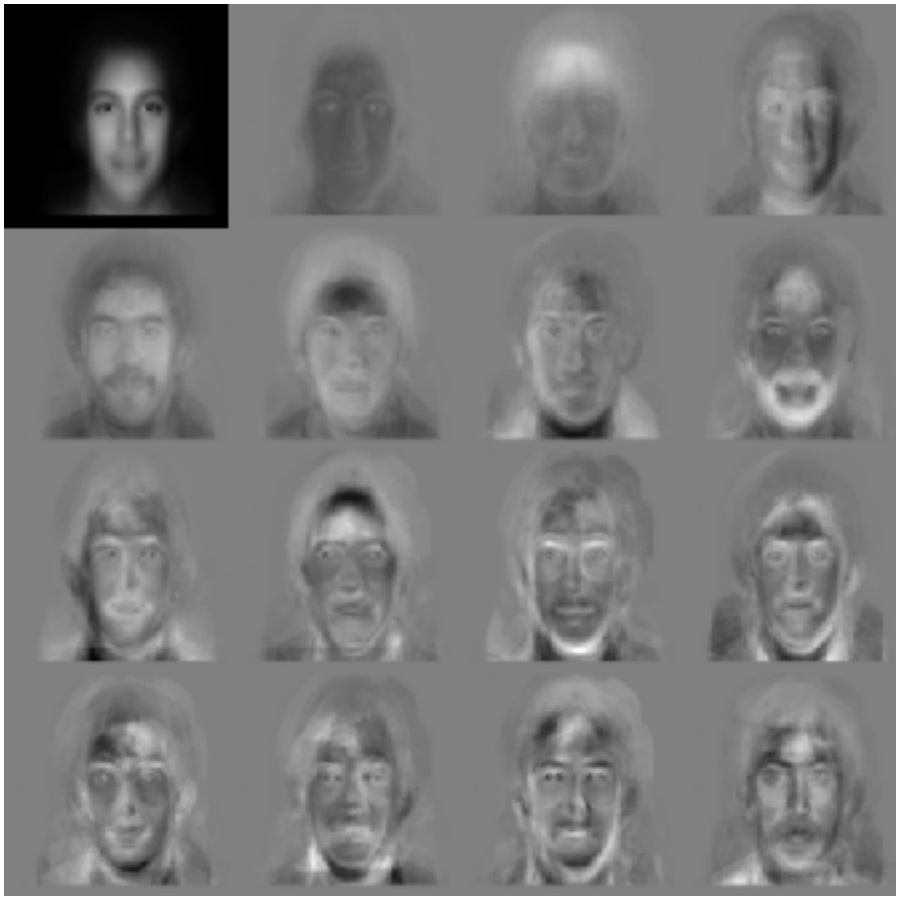
rétine (retina-scan)

cette mesure biométrique est plus ancienne que celle utilisant l'iris, mais elle a été moins bien acceptée par le public et les utilisateurs, sans doute à cause de son caractère trop contraignant : la mesure doit s'effectuer à très faible distance du capteur (quelques centimètres), qui effectue ensuite un balayage de la rétine. Il est physiquement impossible d'effectuer une mesure rétinienne

à une distance de 30cm ou plus sur un sujet mobile comme on peut le voir dans certains films. Cette méthode requiert des sujets coopératifs et entrainés. Pourtant cette technique semble être tout aussi fiable que celle de l'iris ; elle se base sur le fait que le schéma et le dessin formé par les vaisseaux sanguins de la rétine (la paroi interne et opposée de l'oeil) est unique pour chaque individu, différent entre jumeaux et assez stable durant la vie de la personne. La mesure peut ainsi fournir jusqu'à 400 points caractéristique du sujet, que l'on peut comparer aux 30 à 40 points fournis par une empreinte digitale ! En conclusion, la mesure rétinienne est la plus difficile à utiliser mais également la plus dure à contrefaire.

visage (facial-scan)

il s'agit ici de faire un photographie plus ou moins évoluée pour en extraire un ensemble de facteurs qui se veulent propres à chaque individu. Ces facteurs sont choisis pour leur forte invariabilité et concernent des zones du visage tel que le haut des joues, les coins de la bouche, etc... on évitera d'autre part les types de coiffures, les zones occupées par des cheveux en général ou toute zone sujette à modification durant la vie de la personne. Il existe plusieurs variantes de la technologie de reconnaisance du visage. La première est développée et supportée par le MIT et se nomme "Eigenface". Elle consiste à décomposer le visage en plusieurs images faites de nuances de gris, chacune mettant en évidence une caractéristique particulière :



Source : MIT Face Recognition Demo Page

Une autre technique appelée "feature analysis" se base sur la précédente en y rajoutant des informations sur les distances inter-éléments, leurs positions, etc... Elle se dit plus souple quant aux éventuelles modifications pouvant survenir : angle de prise de vue, inclinaison de la tête, etc... Viennent ensuite des techniques moins utilisées à l'heure actuelle, basée sur des réseaux

neuronaux, sur des méthodes plus techniques et moins souples.

système et configuration des veines (vein pattern-scan)

cette technique est habituellement combinée à une autre, comme l'étude de la géométrie de la main. Il s'agit ici d'analyser le dessin formé par le réseau des veines sur une partie du corps d'un individu (la main) pour en garder quelques points caractéristiques.

3.2.4 Caractéristiques comportementales

Outre les caractéristiques physiques, un individu possède également plusieurs éléments liés à son comportement qui lui sont propres :

dynamique des frappes au clavier (keystroke-scan)

les frappes au clavier sont influencées par plusieurs choses ; tout d'abord, selon le texte que l'on tape et, de manière plus générale selon sa nature, on aura tendance à modifier sa façon de taper au clavier. C'est d'ailleurs un des moyens utilisés par certaines attaques (timing attacks) pour essayer d'inférer le contenu ou la nature du texte tapé de façon à remonter jusqu'à un mot de passe par exemple. Ces techniques sont assez satisfaisantes mais restent néanmoins statistiques. Ensuite, le facteur comportemental entre en jeu, et ce facteur va être -lui- différent pour chaque individu. Les facteurs sont à peu de chose près identiques à ceux évoqués précédement : ce sont les durées entre frappes, la fréquence des erreurs, durée de la frappe elle-même... La différence se situe plus au niveau de l'analyse, qui peut être soit statique et basée sur des réseaux neuronaux, soit dynamique et statistique (comparaison continuelle entre l'échantillon et la référence).

reconnaissance vocale (voice-scan)

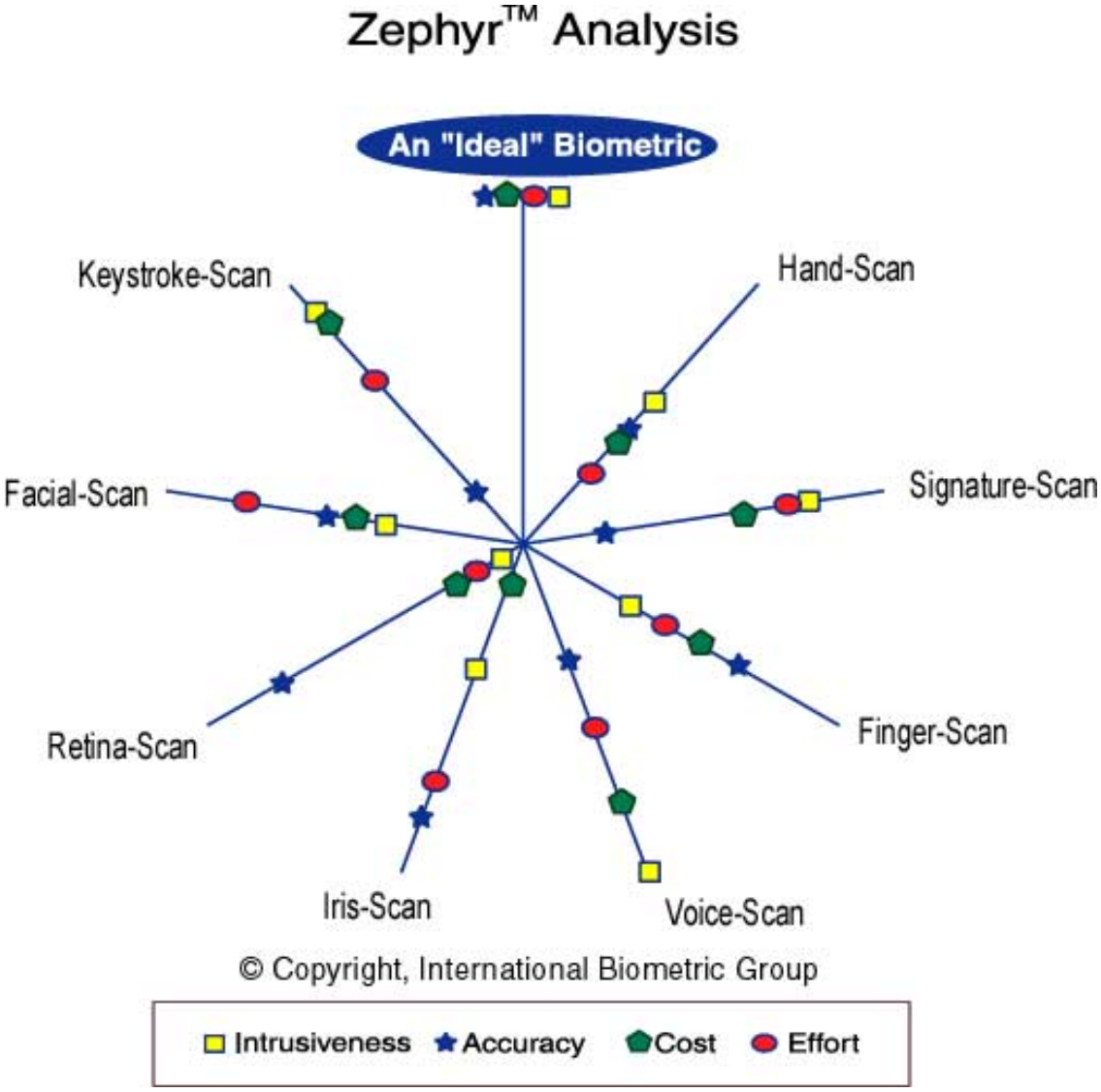
les données utilisées par la reconnaissance vocale proviennent à la fois de facteurs physiologiques et comportementaux. Ils ne sont en général pas imitables.

dynamique des signatures (signature-scan)

ce type de biométrie est à l'heure actuelle peu utilisé mais ses défenseurs espèrent l'imposer assez rapidement pour des applications spécifiques (documents électroniques, rapports, contrats...). Le procédé est habituellement combiné à une palette graphique (ou équivalent) munie d'un stylo. Ce dispositif va mesurer plusieurs caractéristiques lors de la signature, tel que la vitesse, l'ordre des frappes, la pression et les accélérations, le temps total, etc... Bref tout ce qui peut permettre d'identifier une personne de la façon la plus sure possible quand on utilise une donnée aussi changeante que la signature.

3.2.5 Résumé et nouvelles techniques

Voici à titre indicatif le résultat d'une étude effectuée par une companie américaine, l'International Biometric Group (a New York based integration and consulting firm), présentant les différents citères pour chaque type de technique biométrique :



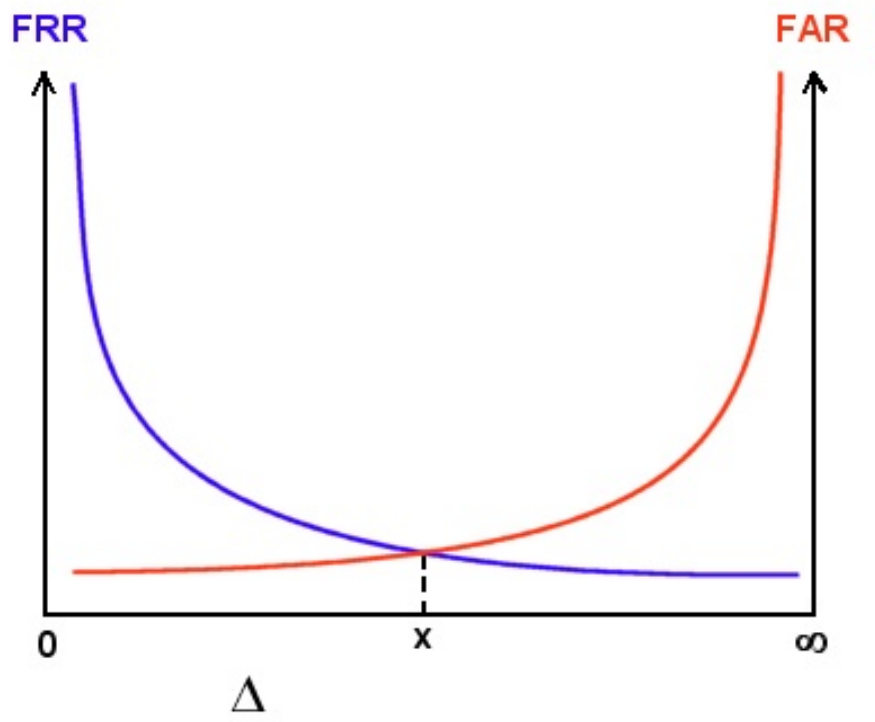
Légende :

- Effort : effort requis pour l'utilisateur lors de la mesure.
- Intrusiveness : décrit dans quelle mesure l'utilisateur perçoit le test comme intrusif.
- Cost : coût de la technologie (lecteurs, capteurs, etc...)
- Accuracy : efficacité de la méthode (capacité à identifier quelqu'un)

Il existe plusieurs techniques en cours de développement à l'heure actuelle ; parmi celles-ci, citons la biométrie basée sur la géométrie de l'oreille, les odeurs, les pores de la peau et les tests ADN. Sur ce dernier point, il est interessant de souligner que le procédé peut se révéler menaçant tant au niveau de la vie privée des personnes, de leur liberté que des dérives informatiques éventuelles (et autres Big Brothers). En effet, même si cela dépend de la technique mise en oeuvre, le test ADN est quelquechose qui peut se révéler comme exact et sûr à 100%, autorisant des FRR et FAR nuls (c.f. plus bas). Il est également reconnu de façon universelle et permettrait très facilement d'effectuer des recoupements entre bases de données. Autrement dit, ce serait le moyen idéal pour "cataloguer" les personnes et détruire ainsi la vie privée que nous avons respectée jusqu'à présent. Le site de la CNIL est un passage incontournable pour ceux que cela intéresse.

3.2.6 Inconvénient de la biométrie : égalité vs similitude

La biométrie présente malheureusement un inconvénient majeur ; en effet aucune des mesures utilisées ne se révèle être totalement exacte car il s'agit bien là d'une des caractéristique majeure de tout organisme vivant : on s'adapte à l'environnement, on vieillit, on subit des traumatismes plus ou moins importants, bref on évolue et les mesures changent. Prenons le cas le plus simple, celui des empreintes digitales (mais on notera que la même chose s'applique à toute donnée physique). Suivant les cas, nous présentons plus ou moins de transpiration ; la température des doigts est tout sauf régulière (en moyenne, de 8 à 10° Celsius au-dessus de la température ambiente). Il suffit de se couper pour présenter une anomalie dans le dessin de ses empreintes. Bref, dans la majorité des cas, la mesure retournera un résultat différent de la mesure initiale de référence. Or il faut pourtant bien réussir à se faire reconnaitre, et en réalité cela marchera dans la plupart des cas car le système autorise une marge d'erreur entre la mesure et la référence. Le but de ce dispositif est simple : les fabricants ne recherchent nullement la sécurité absolue, ils veulent quelquechose qui fonctionne dans la pratique. Ils cherchent donc à diminuer le taux de faux rejets (False Rejection Rate, FRR), tout en maintenant un taux relativement bas de fausses acceptations (False Acceptation Rate, FAR). Explications : une FR est le fait de rejeter une personne autorisée en temps normal car sa mesure biométrique présente trop d'écart par rapport à la mesure de référence pour cette même personne. Un système fonctionnel aura un FRR le plus bas possible. D'autre part, une FA est le fait d'accepter une personne non-autorisée. Cela peut arriver si la personne a falsifié la donnée biométrique ou si la mesure la confond avec un autre personne. Un systême sûr aura un FAR le plus bas possible. Dans la vie courante, les industriels cherchent principalement à avoir un compromis entre ces 2 taux, FRR et FAR, qui sont eux liés suivant une relation illustrée ici :



Ce graphe est purement démonstratif; delta représente la marge d'erreur autorisée par le système, variant de 0 à l'infini. Très succintement, on voit que plus la marge d'erreur autorisée est importante, plus le taux de fausses acceptations augmente, c'est-à-dire que l'on va accepter de plus en plus de personnes qui ne sont pas autorisées (et donc la sécurité du système diminue). Par contre on voit que le taux de rejet des personnes autorisées diminue également, ce qui rend le système plus fonctionnel et répond mieux aux attentes des utilisateurs. A l'autre extrémité, si l'on diminue la marge d'erreur acceptée par le procédé de mesure biométrique, les tendances des 2 taux sont inversées : on va de moins en moins accepter des individus essayant de frauder mais on va aussi, par la même occasion, avoir un taux de rejet sur des personnes autorisées qui sera trop important pour être toléré dans la plupart des cas. Le compromis habituel est de prendre la jonction des courbes, c'est à dire le point x où le couple (FAR, FRR) est minimal. En conclusion, toute la biométrie peut se résumer pour les plus pessimistes à ce seul compromis qui fausse toute la confiance que l'on pourrait porter à cette technologie.

3.2.7 Exemple de vulnérabilité

le cas des empreintes digitales Les empreintes digitales représentent sans aucun doute les données biométriques les plus courament utilisées. De fait, on trouve un grand nombre de produits disponibles sur le marché mais également beaucoup de travaux sur le sujet et de contrefaçons dans ce domaine. Nous allons voir quelques unes des techniques utilisées ainsi que la façon dont elles sont contournées. Attention, cette rubrique ne se veut pas exhaustive; d'une part, elle se base sur les technologies actuelles qui sont par nature variables et évolutives ; et d'autre part son but est de sensibiliser le lecteur de manière générale plutôt que de le former à une quelconque technique.

Il convient tout d'abord de se procurer les données fondamentales de la mesure, c'est à dire les points caractéristiques de l'empreinte digitale que l'on veut contrefaire, en fabriquant un faux doigt (ou fine couche de silicone reproduisant la géométrie de doigt). Nous ne donnerons pas ici le mode opératoire, mais sachez qu'il est tout à fait possible et simple de créer un faux doigt à partir d'une simple empreinte (sur un verre par exemple, ou sur un clavier, une poignée, etc...). Ensuite, examinons les cas pour chaque type de capteur :

- capteur de température : la fine couche de silicone ne fait varier la température que de 1 à 3° Celsius en moyenne, ce qui n'est pas détectable par les capteurs sous peine d'avoir une FRR trop élevée (surtout en extérieur).
- capteur de battements cardiaques : la fine couche de silicone permet au capteur de fonctionner normalement. De plus, toute discrimination basée sur cette mesure est physiquement impossible et infaisable. Infaisable car dans le cas de sportifs par exemple, leur rythme cardiaque peut descendre jusqu'à 40 battements/minute, ce qui suppose une mesure durant plus de 4 secondes pour pouvoir évaluer la fréquence cardiaque. Impossible enfin car quoi de plus changeant qu'un rythme cardiaque ? le moindre effort le modifie, ce qui le rend inutilisable dans notre cas.
- capteur de conductivité : suivant le type de capteur, on estime la valeur normale pour la peau à 200 kOhms. Néanmoins, cette valeur sera de plusieurs MOhms pendant l'hiver (sec) pour descendre à quelques kOhms durant un été humide. Dans ces conditions, il est évident qu'un faux doigt pourra passer le test sans trop de soucis.
- constante diélectrique relative : très succintement, cette constante identifie dans quelle mesure un matériau concentre les lignes électrostatiques de flux. Ici, le silicone sera rejeté puisque présentant une valeur par trop différente de celle de la peau. Or, il s'avère que la valeur de cette constante pour la peau se situe entre celle de l'eau (80) et celle de l'alcool (24). Autrement dit, il suffit d'enduire le faux doigt d'un mélange eau-alcool (80%/20% ou 90%/10%), de poser le doigt sur le capteur et d'attendre que l'alcool s'évapore. En effet, lorsque l'alcool s'évapore, la valeur de la constante va remonter vers celle de l'eau (de 24 à 80) et atteindre au passage celle de la peau. CQFD....

De manière générale, les faiblesses de ces systèmes ne se situent pas au niveau de la particularité physique sur laquelle ils reposent, mais bien sur la façon avec laquelle ils la mesurent, et la marge

d'erreur qu'ils autorisent. Là encore, il convient de ne pas se laisser impressionner par une image illusoire de haute technologie - produit miracle.

3.2.8 Limites de cette technologie

- les données biométriques ne devraient pas être utilisées seules pour de l'authentification forte car elles ne sont pas modifiables puisque par nature propres à chaque individu. On ne peut donc pas se permettre de se baser uniquement dessus, d'autant plus que nous avons vu qu'elles ne sont pas fiables à 100% (FAR/FRR). En règle générale, on préfèrera utiliser la biométrie dans le cadre d'un schéma d'identification plutôt que pour faire de l'authentification.
- les données biométriques sont comparables à tout autre système de contrôle d'accès comme des mots de passe, etc..., car du point de vue du système informatique, ce ne sont rien d'autres que des séries de bits comme toute donnée. Autrement dit, la difficulté réside dans la contrefaçon de la caractéristique physique et biologique que l'on mesure, mais en aucun cas dans sa valeur numérisée (digitale). Prenons l'exemple de notre vieil ami, le login/mot de passe. Ce système est souvent décrit comme peu sûr car une des principales attaques consiste à épier les transactions durant un processus de login pour récupérer les données utiles et les rejouer. On voit que même dans le cas des techniques basées sur la biométrie, cela reste possible ! A quoi bon se compliquer la tâche en essayant de reproduire une empreinte alors que l'on peut récupérer les données numérisées directement ? Ou si l'on peut attaquer les bases de données contenant toutes les données biométriques de référence ?

3.2.9 Conclusion

On retiendra plusieurs fait marquants concernant la biométrie :

- il ne suffit pas de remplacer un login/mot de passe par une mesure de biométrie ; il faut également repenser tout le système et sécuriser l'architecture complète.
- il ne faut pas utiliser une mesure biométrique seule pour procéder à une authentification ; on préfera la coupler avec une carte à puce, un token sécurisé (petit élément de stockage présentant une grande résistance aux attaques, même physiques), un mot de passe voire un OTP (c.f. article concernant les OTP).
- on utilisera la biométrie de préférence pour les opérations d'identification plutôt que d'authentification.
- enfin, perdons une fois pour toute cette image de technologie ultra sûre faussement propagée par les médias. La biométrie n'est nullement une "solution miracle et universelle" !

3.3 Les Firewalls

3.3.1 Principes

Derrière le mot "garde barrière" / Pare-feu (dans la suite désigné par GB) se cache plutôt un concept qu'un matériel ou un logiciel. Nous dirons qu'un GB peut être généralement constitué de plusieurs éléments parmi lesquels on distinguera :

- un (des) routeur(s) assurant les fonctionnalités de filtrage,
- une (des ?) machine(s) dite(s) "système bastion" (SB) qui entre autre assure(nt) les fonctions :

- de passerelle applicative, (ou "proxy") pour les applications, les plus connus sont Telnet, Rlogin, Mail, Ftp, X11, Gopher, W3, etc,

- d'authentification des appels entrants, avec éventuellement mise en oeuvre de systèmes comme S/KEY,

- d'audit, log, trace des appels entrants ainsi que des sessions mail, W3, etc.

Le rôle d'un environnement GB est d'assurer un certain niveau de protection du réseau interne, tout en permettant de travailler sans trop de contraintes.

3.3.2 Le pourquoi (les raisons) d'un garde barrière ?

Plusieurs raisons parmi lesquelles :

Se protéger de malveillances "externes" : les curieux qui génèrent du trafic, qui font plus de peur que de mal, mais qui, quelques fois, finissent par coûter cher, les vandales, ceux qui veulent embêter pour embêter, (saturation de liaisons, saturation de CPU, corruptions de données, mascarade d'identité, etc), les "espionnages", (problèmes de confidentialité de l'information).

- Restreindre le nombre de machines à surveiller et à administrer sur le bout des doigts, (ceci ne signifiant pas pour autant que les autres machines soient gérées par dessous la jambe !).

Par conséquent, l'investissement (minimum) dans une solution intelligente peut s'avérer rentable pour la suite.

- Avoir un point de passage obligé permettant : de bien vérifier si les règles de sécurité telles que spécifiées dans la politique sécurité d'établissement sont réellement celles qui sont appliquées, de contrôler le trafic entre le réseau interne et externe, d'auditer/tracer de façon "centrale" ce trafic, aider à prévoir les évolutions du réseau (statistiques possibles). éventuellement d'avoir une vue de la consommation Internet des diférents utilisateurs/services.

- Possibilité de mettre en oeuvre des outils spécifiques que l'on ne pourrait activer sur tous les systèmes (exemple : systèmes d'authentification à mots de passe uniques, statistiques/comptabilité, etc.). - Economiser les adresses IP ! En effet, dans certaines configurations, un réseau interne derrière un GB, peut utiliser des adresses IP du RFC 1918 lesquelles adresses ne sont pas ni connues, ni "routées" sur Internet.

3.3.3 Les fonctionnalités d'un coupe-feu

Les coupe feu ont su s'adapter aux besoins de sécurité liés au raccordement des réseaux d'entreprises à Internet. Outre le contrôle d'accès, les fonctions de confidentialité et d'intégrité des données sont désormais intégrées dans les solutions. En effet, de simple gestionnaires d'adresses autorisées, les coupe-feu on évolué vers : - le contrôle d'accès aux applications, - l'isolement du réseau extérieur et authentification des utilisateurs, - chiffrage des échanges.

Contrôler les accès

Le contrôle d'accès permet non seulement d'accepter ou de rejeter des demandes de connexions transitant par le coupe feu, mais aussi d'examiner la nature du trafic et de valider son contenu, grâce aux mécanismes de filtrages. On distingue deux types de filtrages : le filtrage statique et le filtrage dynamique.

Le filtrage statique (ou de paquets) est une des premières solutions coupe-feu mise en oeuvre sur Internet. Ce système inspecte les paquets IP (en-tête et données) de la couche réseau afin d'en extraire le quadruplet (adresse source, port source, adresse destination, port destination), qui identifie la session en cours. Cette solution permet de déterminer la nature du service demandé et de définir si le paquet IP doit être accepté ou rejeté. Le principal intérêt du filtrage statique réside dans sa transparence vis-à-vis des postes utilisateurs, ainsi que dans la vitesse des traitements.

La configuration d'un filtre s'effectue généralement au travers de liste de contrôle d'accès (Access Control List ou ACL), constitué par la mise bout à bout des différentes règles à suivre. Cette liste est lue séquentiellement jusqu'à la dernière règle applicable qui est retenue. Par exemple, une première règle indique que toutes les machines peuvent se connecter au serveur Web sur le port 80, et la suivante autorise le serveur Web à répondre à tous les clients du service (sur un port supérieur à 1024). Ces règles permettent à toutes les machines d'accéder au Web, sans pour autant bloquer les connexions ou autres applications. Il convient de rajouter en fin de liste une règle spécifiant que toute communication est interdite sur l'ensemble des services et des machines qu'elle soit en entrée ou en sortie du réseau protégé. Le principe consiste en fait à interdire tout ce qui n'est pas explicitement autorisé.

Cet exemple est celui d'un service reposant sur TCP, un protocole destiné à établir des circuits virtuel. La différenciation entre appel entrant et appel sortant repose sur une information de

l'en-tête (le bit ACK) qui caractérise une connexion établie. Ce type de distinction n'existe pas pour le protocole de datagramme UDP ; différencier un paquet valide d'une tentative d'attaque sur le service s'avère donc irréalisable. Cette problématique se retrouve également avec certaines applications qui répondent aux requêtes des clients sur les ports alloués dynamiquement. C'est le cas, notamment, de FTP (un circuit pour les commandes et un pour les données) ou RPC (Remote Procedure Call) qui utilisent un service distinct pour répondre aux demandes de localisation.

Il est généralement impossible de gérer de façon satisfaisante ce type de protocole sans ouvrir l'accès à un plus grand nombre de ports, et donc de rendre le réseau plus vulnérable.

Le filtrage dynamique reprend le principe de travail du filtrage statique au niveau de la couche réseau, ainsi que la transparence de sa mise en place. En revanche, son efficacité s'étend à la quasi totalité des protocoles couramment utilisés (TCP, UDP, RPC), grâce à l'implémentation de tables d'état pour chaque connexion établie. Les performances de traitement des paquets s'améliorent en raison d'une identification rapide des paquets correspondants à une session déjà autorisée. Pour gérer l'absence de circuit UDP et l'allocation dynamique de port implantés par service, le filtrage dynamique examine en détail les informations jusqu'à la couche applicative. Il interprète ainsi les particularités liées à l'application, et crée dynamiquement les règles pour la durée de la session, facilitant le passage du paquet IP entre les deux machines autorisées. Il est également possible de contrôler le sens des transferts pour FTP (put ou get), ainsi que le mode de connexion HTTP (post ou get).

Isoler le réseau et authentifier les utilisateurs

Les besoins de contrôle portent de plus en plus sur la nature même des données échangées. Les relais applicatifs s'interposent entre les clients et les applications pour une surveillance spécifique. Le principe des relais applicatifs consiste à ce que le relais apparaisse pour le client comme le serveur légitime ; il contrôle et relaie alors toutes les requêtes vers le serveur demandé, puis transmet les réponses au client initial. Ce mécanisme agit donc à la fois comme client et comme serveur. Cela implique qu'un relais ait été développé pour chaque application (messagerie, serveur web, téléchargement etc.). Les relais applicatifs permettent d'isoler le réseau de l'extérieur, avec les serveurs applicatifs aucun flux d'information ne circule en direct entre le réseau protégé et les autres réseaux. Toutes les données sont obligatoirement acheminées vers le relais, qui décide quelle action mener. Les relais applicatifs permettent de vérifier l'intégrité des données et authentifier les échanges. Les possibilités offertes par ce mécanisme vont bien au-delà du contrôle d'accès de la base. En effet, le positionnement au niveau applicatif autorise le relais à vérifier l'intégrité des données et à examiner les particularités de l'application. En outre, sa proximité avec l'interface utilisateur facilite la mise en oeuvre d'authentification. Une autre fonctionnalité des coupe feu réside dans l'authentification des utilisateurs. L'authentification demeure indispensable pour effectuer un contrôle d'accès fiable portant sur l'identité des usagers des services. Le mécanisme le plus couramment employé consiste à associer un mot de passe à l'identifiant d'une personne. Toutefois, dans les communications réseaux, ce mot de passe est souvent envoyé en clair. Sur un réseau public comme Internet cette situation n'est pas acceptable. Pour y remédier, les coupe-feu proposent des solutions basées sur le chiffrement. Les techniques de chiffrement seront développées dans le paragraphe ci-dessous.

Chiffrer les données pour relier les réseaux via Internet

Les réseaux privés virtuels utilisent Internet pour interconnecter de façon sûre les différents réseaux des filiales et partenaires d'une entreprise. Les moyens de chiffrement sont la base des mécanismes mis en oeuvre par les coupe-feu (Firewall).

Pour pallier le manque de confidentialité sur Internet et sécuriser les échanges, des solutions poste à poste telles que le protocole SSL (couche transport) ou les GPSS-API (couche application) oint été développées. Basées sur des principes de chiffrement, elles assurent la confidentialité et l'intégrité des échanges client-serveur. Grâce à Internet, les sociétés peuvent interconnecter leurs différents sites, et fournir à leurs employés en déplacement un accès contrôlé à moindre coût. C'est

pourquoi les coupe-feu (tunnel chiffrant établie entre les deux sites d'une société), ou entre client et coupe-feu.

Diverses solutions de chiffrement émergent, mais qui ne sont pas inter-opérables entre eux. Le groupe IPSEC (IP-SECURE), mis en place par l'IETF (Internet Engineering Task Force), travaille donc à une standardisation de ses mécanismes. Chargé de soumettre des solutions de sécurité pour le protocole IP, IPSEC a développé la norme ESP (Encapulsating Security Payload) qui propose deux alternatives offrant confidentialité et intégrité des données.

La première chiffre le datagramme IP dans son intégralité. Cette technique, qui intervient au niveau des passerelles d'interconnexion, permet alors de masquer les adresses IP, et la nature des flux entre des machines communiquant à travers le tunnel ainsi crée.

La seconde alternative, destinée à préserver les performances de routage, ne chiffre que les en-tête de la couche transport (TCP/UDP), préservant ainsi les en-tête IP.

Ces technologies reposent sur des méthodes de chiffrement, les quelles nécessitent algorithmes employés, ainsi que la négociation de clés de session entre les deux parties communicantes.

3.3.4 Quelques exemples de firewalls

- Firewall-1 de CheckPoint
- Mwall de Matranet
- Pix de Cisco
- Netwall d'Evidian

3.4 L'authentification htaccess

3.4.1 Concept d'authentification

Ce système d'authentification est fréquemment utilisé pour restreindre l'accès au contenu de répertoires spécifiques, sur un intranet ou sur Internet. Le fichier contenant les informations de configuration relatives aux personnes ou groupes de personnes habilitées à accéder les données protégées, ainsi que leurs droits, se nomme ".htaccess" par défaut. Il est stocké dans le même répertoire où résident les données à portéger.

3.4.2 Fonctionnement

La méthode d'authentification htaccess a été développée en même temps que les programmes destinés à récupérer des données "Web" sur Internet, tel que les démons HTTPd. Ainsi, dans une url (Universal Resource Locator), la commande "http ://" va être interprétée par le démon (il s'agit du programme sur le serveur Web attendant toute connexion ou requête pour la traiter) ; celui-ci dispose d'un fichier global de gestion des accès stocké à la racine le plus souvent. Les fichiers .htaccess représentent des niveaux additionnels dans la gestion des accès, et apportent un raffinement lié à chaque répertoire.

Ainsi, si le démon trouve un fichier .htaccess dans l'arborescence à parcourir pour accéder au fichier requis par le client, il va procéder suivant les informations contenues dans ce fichier : il va soit interdire l'accès et refuser la requête, soit demander une authentification de l'utilisateur via login/password. Il est intéressant de noter que la plupart du temps les données d'authentification (à l'inverse des données de configuration) sont stockées à un autre endroit dans l'arborescence, protégées de tout accès via le Web (par exemple avec un fichier .htaccess où est spécifié "Deny from All"). Le démon va comparer ces données avec celles renvoyées par l'utilisateur lors de la requête d'authentification et autoriser, suivant le résultat du test, l'utilisateur à accéder ou non à la page web.

3.4.3 Aspects pratiques : le fichier .htaccess

On retrouve ce type d'authentification dans la plupart des distributions : Apache permet l'utilisation de fichiers nommés ".htaccess" par défaut. Netscape utilise des fichiers nommés ".nsconfig" dont la syntaxe varie quelque peu. Du côté de cette syntaxe, nous allons voir celle qui est la plus habituelle, à savoir celle utilisée par Apache ou NCSA HTTPd ; un exemple typique de fichier .htaccess est le suivant :

```
AuthUserFile /repertoire_protege/.htpasswd
AuthGroupFile /dev/null
AuthName Area_51
AuthType Basic
require user roswell
```

Nous utilisons ici un fichier ".htpasswd" qui est placé dans le répertoire "/repertoire_protege", et qui contient nos paires de login/password de références. Nous verrons ce fichier un peu plus loin ; nous n'utilisons pas de fichier définissant des groupes d'utilisateurs : nous sommes dans un cas simple (le paramètre "/dev/null" correspond au device null sous Unix, autrement dit à quelquechose d'inexistant). "Area_51" est le nom que nous donnons à cette authentification (éviter les espaces) et "Basic" est le type d'authentification.

La deuxième partie du fichier est celle où nous allons définir les droits requis pour accéder au contenu du répertoire dans lequel se trouve notre fichier .htaccess . Ainsi dans le cas présent, nous n'autorisons que l'utilisateur "roswell" (attention à la casse) à accéder à notre répertoire. Lors de l'authentification, cet utilisateur donnera son mot de passe qui sera alors comparé à la valeur contenue dans notre fichier de mots de passe, à savoir .htpasswd.

Nous allons voir maintenant qu'il est possible d'affiner ces droits en limitant suivant le cas les hôtes, requêtes HTTP, fichiers accédés, protocoles, etc...

- premier cas, la limitation des requêtes HTTP. HTTP est un protocole de transfert de données utilisé pour le web (c.f. la rfc 2616), qui comporte un nombre limité de fonctions ; il est possible de n'accepter que certains types de fonctions pour que, par exemple, les utilisateurs ne puissent qu'accéder en lecture au répertoire. C'est le cas dans l'exemple suivant où nous limitons l'accès aux requêtes (fonctions HTTP) de type GET (lecture) pour les utilisateurs roswell et mulder :

```
<Limit GET>
require user roswell mulder
</Limit>
```

Plus généralement, nous aurons :

```
<Limit GET>
require valid-user
</Limit>
```

où tout utilisateur présent dans la liste du fichier .htpasswd sera autorisé à effectuer des requêtes GET sur le répertoire protégé.

- autre cas, limitation des fichiers accédés :

```
<Files index.html>
require valid-user
</Files>
```

Nous limitons ici l'accès au fichier spécifié, "index.html", en excluant le reste du répertoire. Cet accès est lui-même limité aux utilisateurs valides (autorisés).

- cas suivant, les restrictions suivant les hôtes :

Expliquons tout d'abord les options utilisées : Order, Deny et Allow. Order permet de spécifier un ordre d'évaluation des critères de test. Allow signifie autoriser les entités satisfaisant le test correspondant et Deny signifie rejeter les entités satisfaisant également le test correspondant. On utilise généralement un combinaison des 2, et suivant l'ordre, la politique de sécurité varie quelque

peu. Dans l'ordre Deny,Allow, les directives Deny sont évaluées avant celles de la clause Allow. Le défaut est d'autoriser l'accès. Tout client qui ne correspond pas à la directive de déni ou qui satisfait au test d'autorisation spécifique Allow se verra autorisé l'accès au serveur web. Dans l'ordre Allow,Deny, les directives Allow sont évaluées avant celles de la clause Deny. Le défaut est d'interdire l'accès. Tout client qui ne correspond pas à la directive d'autorisation ou qui satisfait au test de déni se verra refusé l'accès au serveur web.

Exemple :

```
Order Allow,Deny
Allow from apache.org
Deny from foo.apache.org
```

Tout le monde provenant du domaine apache.org est autorisé à accéder au serveur web sauf une sous partie qui est refusée (sous-domaine foo). Le reste du monde est refusé puisqu'il s'agit du défaut dans ce cas.

Variantes : pour autoriser seulement un groupe d'adresses IP : ici celles contenues dans la classe B 129.21.

```
Order Deny,Allow
Allow from 129.21
Deny from All
```

Pour autoriser seulement un groupe d'hôtes ou réseaux : ici le domaine rit.edu .

```
Order Deny,Allow
Allow from rit.edu
Deny from All
```

Pour exclure seulement un groupe d'adresses IP : ici celles contenues dans 129.21.3 .

```
Order Allow,Deny
Allow from All
Deny from 129.21.3
```

Pour exclure seulement un groupe d'hôtes ou réseaux : ici le domaine isc.rit.edu .

```
Order Allow,Deny
Allow from All
Deny from isc.rit.edu
```

- dernier cas, l'authentification sécurisée : elle fait appel au protocole SSL (ou sa version standardisée, TLS) pour les échanges de données, ce qui évite que les mots de passe circulent en clair sur le réseau. Pour l'utiliser, il faut faire des requêtes de type https ://... sur un serveur correctement configuré.

```
AuthDCE On
AuthType Basic
AuthName dce
require valid-user
```

3.4.4 Aspects pratiques

le fichier .htpasswd Le fichier htpasswd contient les login et mots de passe des utilisateurs autorisés à accéder aux pages web. Plusieurs fichiers htaccess peuvent utiliser le même fichier htpasswd comme base de secrets (credentials) centrale si la méthode d'authentification est basique. Mais dans tous les cas ce fichier doit être clairement protégé (bien qu'accessible en lecture par le démon pour lui permettre de l'utiliser) ; le plus souvent, on utilise là encore une protection par htaccess au moyen de la ligne de configuration : "Deny from All", ce qui signifie qu'aucun accès (du démon donc via le web) n'est autorisé.

Par contre, ce fichier reste accessible comme tout autre fichier par le système d'exploitation et donc via les autres services tel que FTP.

Pour créer le fichier ".htpasswd", il faut utiliser la commande *nix htpasswd ou utiliser un site web qui fournit le même service. La commande type est (-c pour création d'un nouveau fichier) :

```
htpasswd -c /répertoire_destination/.htpasswd login
```

Le système va ensuite demander le mot de passe associé à ce login qu'il va crypter et rajouter au fichier .htpasswd. Voici un exemple de ce que l'on peut trouver dans un fichier .htpasswd :

```
foobar :Z39sR$s9xLyx
karen :44KvbqBfLZ5Yw
```

La fonction htpasswd accepte plusieurs types de cryptage des mots de passe :

- -m utilise la fonction de hachage MD5 (128 bits). Attention, Apache utilise une version spécifique de l'algorithme, ce qui signifie qu'il n'est pas interoperable avec les autres serveurs web.
- -d utilise la fonction système crypt(). Pour rappel, cette fonction est basée sur le DES, et est également utilisée pour le cryptage des most de passe système (fichier passwd ou shadow).
- -s utilise la fonction de hachage SHA-1 (160 bits).
- -p laisse les mots de passe en clair.

3.4.5 Faiblesse htaccess : l'authentification HTTP

Comme nous l'avons vu précédemment, htaccess est un processus d'authentification qui va être reconnu par le démon HTTP lorsqu'il essayera d'accéder aux fichiers pour les envoyer au client. Mais cette authentification repose en fait complètement sur les fonctionnalités du protocole HTTP (c.f. la rfc 2617), et le démon va demander au client de s'authentifier via une requête particulière. Prenons l'exemple suivant :

```
GET http ://www.securiteinfo.com/restricted_zone/ HTTP/1.0
Accept : image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, appl
Accept-Language : fr
User-Agent : Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
Host : www.securiteinfo.com
Proxy-Connection : Keep-Alive

HTTP/1.1 401 Authorization Required
Via : 1.1 PROXY2
Connection : close
Content-type : text/html; charset=iso-8859-1
Date : Wed, 22 Aug 2001 15 :25 :40 GMT
Server : Apache/1.3.12 (Unix) Debian/GNU mod_perl/1.24
Www-authenticate : Basic realm="Acces Restreint"

GET http ://www.securiteinfo.com/restricted_zone/ HTTP/1.0
Accept : image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-excel, appl
Accept-Language : fr
Authorization : Basic QWxpY2U6TGFwaW4=
User-Agent : Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 4.0)
Host : www.securiteinfo.com
Proxy-Connection : Keep-Alive

HTTP/1.1 200 OK
```

```
Via : 1.1 PROXY2
Connection : close
Content-type : text/html
Date : Wed, 22 Aug 2001 15 :25 :45 GMT
Server : Apache/1.3.12 (Unix) Debian/GNU mod_perl/1.24
```

Explications :

- L'utilisateur souhaite accéder à une page web qui s'avère être protégée par htaccess. Concrètement, il y a 2 échanges requête/réponse HTTP qui sont éffectués pour donner accès à cette page.
- Le premier est une requête simple (un GET) signifiant que le client souhaite accéder à la page. La réponse est "401 Authorization Required", ce qui signifie que la page est protégée et nécessite une authentification (de type "Basic"). L'utilisateur ne voit pas cette réponse, seul le navigateur (browser) la voit et affiche en réponse une popup dans laquelle il demande à l'utilisateur de taper son login et mot de passe.
- Après cette opération, le navigateur réitère son GET en ajoutant cette fois-ci les informations de l'utilisateur ("Authorization : Basic QWxpY2U6TGFwaW4=") qui serviront au serveur pour l'authentifier.
- Si tout se passe bien, la requête est acceptée ("200 OK") et l'utilisateur peut accéder à la page web.

Nous avons ici le cas le plus simple : les informations de l'utilisateur circulent quasiment en clair sur le réseau. Prenons l'élément "crypté" : QWxpY2U6TGFwaW4=. Cet élément est en fait "login :password" uuencodé en base 64, ce qui n'est d'aucune protection (l'uuencodage est un procédé servant à coder du binaire en ASCII et inversement, autrement dit de passer de 24 à 32 bits tout en restant dans l'intervalle des caractères imprimables, ce qui permet de transférer des fichiers binaires sous forme de texte par exemple).

Copions cet élément dans un fichier type :

```
begin-base64 644 my_file
QWxpY2U6TGFwaW4=
====
```

Un simple passage dans la fonction *nix uudecode nous donne :

Alice :Lapin

Nous avons retrouvé le login : "Alice" et le mot de passe : "Lapin".

Il existe une autre méthode utilisée pour coder les informations transitant sur le réseau : on utilise l'algorithme de hash MD5 (c.f. la rfc 1321). Les propriétés d'une fonction de hachage rendent impossible le fait qu'un attaquant puisse remonter aux informations initiales (login/password) ; de plus, le hasché reste caractéristique de ces données, autrement dit un hasché correspond à un et un seul texte original (dans la limite de son intervalle de sortie, à savoir 2^128 possibilités pour MD5). Néanmoins, cela ne préserve pas des "replay-attacks", dans lesquelles l'attaquant va se contenter d'intercepter ce hasché et de l'utiliser à son propre compte comme s'il s'agissait du sien : il n'a nul besoin de posséder le texte original puisque c'est le hasché qui est demandé ! Pour remédier à cette faiblesse, l'authentification HTTP utilisant cet algorithme va donc rajouter des éléments -uniques- dans le calcul du hasché, le plus souvent en envoyant un challenge (le "nonce") au client lors de la requête d'authentification. Le client va rajouter ce challenge dans le calcul du hasché, le rendant unique par la même occasion (c.f. la rfc 2069 et suivantes).

```
HTTP/1.1 401 Unauthorized
(...)
WWW-Authenticate : Digest realm="testHash",
nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
opaque="5ccc069c403ebaf9f0171e9517f40e41"
```

```
Authorization header : Authorization : Digest
username="Alice",
realm="testHash",
nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
uri="/dir/index.html",
response="e966c932a9242554e42c8ee200cec7f6",
opaque="5ccc069c403ebaf9f0171e9517f40e41"
```

Malheureusement, cette deuxième méthode d'authentification est peu utilisée (cela dépend entre autres des fonctionnalités du serveur, du navigateur, etc...). Les navigateurs suivants supportent l'authentification basée sur MD5 :

- Internet Explorer 5.0 et +
- Amaya
- NCSA Mosaic/X 2.7
- Spyglass Mosaic

Alors que ceux-ci ne la supportent pas (liste non-exhaustive) :

- Netscape Communicator 4.5 (Mac) et 4.7 (PC)
- iCab Preview 1.7 (Mac)

3.4.6 Avantages et désavantages

La méthode d'authentification par htaccess permet de déléguer le contrôle d'accès au niveau local, et autorise ainsi plus de flexibilité pour créer et changer les droits d'accès suivant les besoins. Par contre, on comprendra que ce système devient rapidement ingérable lorsque le nombre d'utilisateurs et/ou de répertoires augmentent, rendant toute politique de sécurité globale impossible. D'autre part, le système reste faible dans son concept ; il est basé sur les services du Web à l'exclusion de tous les autres sevices d'Internet, ce qui n'est pas une hypothèse raisonnable. En effet, tout utilisateur malveillant qui a accès au serveur par un autre moyen ou service (ce qui n'est pas irréaliste) sera capable de modifier et corrompre complètement ce système d'authentification. Ainsi on peut dresser une liste (non-exhaustive bien sûr) qui permet de se faire une idée du nombre de menaces à prendre en compte lorsque l'on souhaite utiliser ce système d'authentification :

- accès via FTP (utilisateur autorisé)
- accès via FTP (utilisateur non-autorisé, buffer-overflows et autres exploits type wu-ftpd)
- accès via Telnet (sur services particuliers, la liste étant trop longue pour être citée...)
- accès via Web (script cgi non protégé type PHF, débordements)
- ...

3.4.7 Conclusion

Nous avons vu que la méthode d'authentification par htaccess comporte beaucoup d'inconvénients, voir de faiblesses, pour un nombre d'avantages assez restreint. Néanmoins, il ne faut pas oublier son principal intérêt qui est le fait qu'elle reste la seule méthode facilement utilisable par le client de base d'un fournisseur d'accès internet et d'espace web. En effet la plupart du temps cette personne n'a pas accès aux serveurs et ne peut donc pas faire appel à des services auxiliaires pour des types d'authentifications alternatives. Les quelques options restantes sont plus compliquées à implémenter (PHP/MySQL par exemple) surtout si l'on souhaite assurer un bon niveau de sécurité (modules cryptographiques en PHP). Et l'on ne parle même pas des solutions utilisées par la plupart des interfaces web de mail gratuit type Yahoo !, où l'on utilise un simple POST dans lequel le mot de passe est présent en clair !

Pour finir, la méthode htaccess peut très bien être sécurisée car elle en possède les moyens : cryptographie avec MD5 ou sécurisation de bout en bout grace à SSL... A chacun de s'assurer que ces mesures soient correctement utilisées.

3.5 Les mots de passe

3.5.1 Introduction

Les mots de passe ont été, dès le début de l'informatique, la solution la plus simple à mettre en oeuvre, et qui procure un minimum de sécurité. Encore aujourd'hui, les mots de passe font légion dans les logiciels, les systèmes d'exploitation, les systèmes embarqués, etc. Pourtant, on considère qu'environ 30% des mots de passe sont amenés à être découverts. Il faut donc les choisir avec soin pour minimiser les risques.

3.5.2 Choix d'un mot de passe

Choisir un bon mot de passe n'est pas si évident que ça en a l'air. Il faut respecter quelques règles :

- Ne jamais choisir un mot du langage courant. Des logiciels spéciaux de type dictionary cracking sont spécialisés dans ce domaine.
- Ne jamais prendre un mot qui est proche de vous : Votre prénom, le nom de jeune fille de votre femme, le nom du chien, des enfants, de votre hobby préféré...
- Ne jamais prendre un mot inférieur à 6 lettres. Des logiciels spéciaux de type brute force cracking sont spécialisés dans ce domaine.
- Un mot de passe ne doit jamais être écrit quelque part. La première chose que fait un pirate, est de fouiller dans vos affaires : Regarder dans votre agenda, sous l'écran, sous le clavier, dans votre poubelle, rechercher un fichier du type "mdp.txt" dans votre disque dur, etc.

Bref, le mieux est de prendre un mot de passe constitué de chiffres et de lettres, et éventuellement de majuscules et minuscules. Par exemple : diZmLvc4dKJvc51 est un excellent mot de passe ! Le problème est qu'il est difficile à retenir. Une bonne méthode est de constituer un anagramme. Par exemple, si l'on prend " bonjour toi 2000 ", on peut prendre une lettre sur deux (on oublie les espaces), et couper le chiffre en deux. Ca donne : 20bnoroojuti00. Il n'y a guère mieux comme mot de passe... Encore plus facile à retenir, on peut utiliser le verlant, cela donne : 20toijourbon00. Bien sûr, il existe d'autres méthodes, à vous d'imaginer la vôtre.

3.5.3 Conclusion

Maintenant que l'on a un bon mot de passe, il est préférable de prendre deux mesures supplémentaires :

- Prendre un mot de passe différent par application (comptes mail, login FTP, accès à un site web...).
- Changer régulièrement de mot de passe.

Les risques d'intrusion par détection de votre mot de passe sont maintenant considérablement réduits.

3.6 Les outils de détection d'intrusions gratuits

3.6.1 Tiger-2.2.4

Tiger est un ensemble de scripts qui recherche dans un système les faiblesses qui pourraient permettre à une utilisation non-autorisée d'en changer les configurations, d'accéder à la racine ou de modifier des fichiers systèmes importants. A l'origine, Tiger fut développé à l'université du Texas A&M. Il peut être chargé à partir de l'adresse suivante : tamu.edu. Il existe plusieurs versions de Tiger disponibles : ·

- Tiger-2.2.3 pl, la dernière version en date avec des scripts check_devs et cheek_rhost actualisées. ·
- Tiger-2.2.3 pl-ASC, une version disposant de contribution du Arctic Regional Supercomputer Center ; ·

– Tiger-2.2-4 pl, version du tiger -2.2.3 avec support linux.
Tiger balaye le système à la recherche de cron, inted, passwd, d'autorisation de fichiers, de pseudonymes et de variables PATH pour voir s'ils peuvent être utilisés pour accéder au répertoire principal. Il analyse les vulnérabilités du système via l'utilisation d'inted pour déterminer si un utilisateur peut accéder à distance au système. Il a recours également aux signatures digitales ; à l'aide de MD5, pour déterminer si les systèmes de programmes binaires clés ont été modifiés.

3.6.2 Logcheck 1.1.1

Logcheck est un script qui analyse les fichiers journaux des systèmes et recherche toute activité inhabituelle ainsi que les attaques. Bien entendu, cela veut dire qu'un intrus n'a pas encore obtenu l'accès au répertoire de l'hôte et ne peut donc modifier les fichiers journaux . L'un des gros problèmes dans la maintenance des fichiers journaux est la quantité d'information collectée sur des systèmes importants, l'analyse (par scanning) manuelle des fichiers journaux peut demander plusieurs jours. Logcheck simplifie le contrôle du journal système en classant les informations reprises dans le journal et en l'envoyant par e-mail à l'administration système. Logcheck peut être configuré de manière à n'envoyer dans un rapport que les informations que vous souhaitez et ignorer celles que vous ne désirez pas. Logcheck peut être chargé sur : Téléchargement Logcheck C'est l'un des éléments du projet Abacus, un système de prévention d'intrusion. Cependant, tous les éléments ne sont pas encore stables. Logcheck est basé sur un programme de contrôle de journal appelé " frequentchech.h ", un élément du Gauntlet Firewall Package. Le script logcheck.sh est installé sur /usr/local/etc/, ainsi que les fichiers des mots clés. Le script peut être chargé sur : Logcheck Script

3.6.3 Tripwire

Tripwire est un des outils les plus connus et les plus utiles dans la détection d'intrusion et la récupération qui s'en suit. Tripwire crée une base de données de signatures des fichiers dans le système et lorsqu'il est exécuté en mode comparaison, il prévient les administrateurs système des changements dans le système de fichiers. La différence entre Tripwire et Tiger est que le premier est spécialisé dans les programmes de signature de fichiers et peut utiliser de multiples fonctions de hashing pour générer des signatures digitales générales. Tripwire a été développé par le laboratoire Computer Opérations Audit and Security Technology (COAST). La version publique disponible Tripwire 1.2, est disponible sur : Tripwire.com

3.6.4 Snort

Snort est un système de détection d'intrusion, son auteur est Martin ROESH, il est léger, pas d'interface graphique, peu coûteux en ressources. Snort permet définition précise des signatures, détecte les entêtes et dans le contenu des paquets dans IP, TCP, UDP et ICMP, détection de Nmap (Scan, OS fingerprint), des petits fragments, dénis de service et de débordement de Buffer (script kiddies) . Snort peut être chargé sur : Snort.org

3.7 Les contrats de licence et la Sécurité Informatique

3.7.1 Introduction

Le but de ce papier est de démontrer en quoi les contrats de licence sur les logiciels informatiques influent sur la sécurité globale des systèmes. Aucun éditeur n'est particulièrement visé mais les logiciels les plus diffusés sont certainement les plus concernés par les remarques qui suivent. Dans toute la suite le terme logiciel englobe les applications de toutes natures ainsi que les systèmes d'exploitation.

3.7.2 Limitations de garantie

Avez-vous déjà lu en entier un contrat de licence portant sur un logiciel ? Plus particulièrement les sections consacrées aux garanties ? Ces sections sont selon moi édifiantes car sans être juriste on y comprend aisément que les éditeurs limitent soigneusement leurs responsabilités. En effet, ils stipulent explicitement qu'ils ne peuvent être tenus pour responsables des dommages que pourrait causer l'utilisation de leurs produits. De telles limitations de garanties peuvent-elles encore être considérées comme normales voire même morales à l'heure où de plus en plus d'éléments de notre vie sont contrôlés par des logiciels ? Des clauses du même genre feraient sourire ou frémir dans l'industrie automobile par exemple car si le système de freinage ne remplissait pas sa fonction une fois sur trois, il serait facile d'imaginer les conséquences pour le véhicule, son conducteur et aussi pour son constructeur.

3.7.3 Responsabilité et Qualité

L'une des conséquences majeures des limitations de responsabilité est selon moi une absence totale d'engagement sur la qualité des produits. En limitant sa responsabilité vis à vis du consommateur (utilisateur du logiciel) l'éditeur peut se contenter de standards de qualité très faibles puisque les défauts ou défaillances des produits ne peuvent pas lui être reprochés. En effet, l'objectif n'est plus alors de produire la meilleure qualité logicielle possible mais plutôt d'assurer le minimum ; ce minimum étant le plus bas niveau de qualité acceptable par le marché. Ainsi, le consommateur choisira non plus le meilleur produit mais le moins mauvais. Dans le monde du logiciel on est très loin des plans d'assurance qualité et des processus de certification tellement répandus dans l'industrie. L'absence d'une démarche de qualité globale et la réduction des cycles de commercialisation (time to market) conduisent à une réduction sensible des campagnes de tests fonctionnels et à une absence quasi totale de tests de sécurité (*). Dès lors, on ne peut s'attendre qu'à des produits peu fiables et dont la sécurité n'a pas été évaluée.

3.7.4 Qualité Fiabilité et Disponibilité

La richesse fonctionnelle et la complexité grandissante des logiciels rendent leur fiabilisation de plus en plus difficile. La fiabilité fait partie des propriétés que toute démarche qualité vise à améliorer. C'est pourquoi un engagement fort des éditeurs me semble nécessaire afin de mettre en oeuvre la rigueur et les processus qui permettront d'améliorer la fiabilité des systèmes. La disponibilité des systèmes repose pour beaucoup sur la fiabilité des logiciels qui y sont installés ; on peut comprendre aisément en quoi la dégradation de la qualité de ces derniers peut poser des problèmes de disponibilité. Améliorer la fiabilité des logiciels revient à oeuvrer pour rendre les systèmes informatiques globalement plus disponibles. Le problème se pose d'une façon encore plus cruciale en ce qui concerne les systèmes d'exploitation. En effet, il n'est pas possible d'espérer construire un ensemble fiable si le système d'exploitation sur lequel il repose n'offre aucune garantie de fiabilité.

3.7.5 Interopérabilité

Actuellement, à ma connaissance aucun éditeur n'accepte dans ses contrats de licence de responsabilités quant à l'interopérabilité de ses produits avec ceux d'autres éditeurs. Pourtant, rares sont les logiciels pouvant remplir l'ensemble de leurs tâches de façon complètement indépendante sans s'appuyer sur les fonctionnalités d'autres briques logicielles. D'une façon générale, les systèmes informatiques sont des ensembles complexes formés de nombreux éléments (logiciels pour la plupart) interagissant de manière continue. Les éditeurs des différents éléments logiciels ne garantissent donc en aucune manière que leurs produits ont la capacité de s'intégrer sans problèmes au sein des systèmes existants ou de remplir une quelconque fonction dans un environnement réel. Il est vrai que la diversité des environnements et la complexité grandissante des systèmes déjà en place rendent des tests d'intégration et d'interopérabilité exhaustifs difficilement envisageable

mais l'absence d'obligations fourni aux éditeurs un parapluie au dessous duquel il est très aisé de s'abriter.

3.7.6 Confiance

Dès lors que l'on permet à un logiciel de s'exécuter sur une machine il faut savoir qu'on lui accorde le contrôle total de celle ci. Cela veut dire que l'on fait confiance au logiciel pour qu'il exécute la tâche pour laquelle il a été acheté et pas autre chose. On lui fait aussi confiance pour exécuter sa tâche sans mettre en danger le fonctionnement des autres programmes présents sur la machine. Les contrats de licence tels qu'ils sont formulés actuellement font de cette confiance un risque réel. L'utilisateur prend le risque de compromettre sa machine en en confiant le contrôle à un programme dont les auteurs ne garantissent pas le bon comportement. Malheureusement la seule référence de l'utilisateur est un discours commercial car il n'a ni la possibilité (sauf pour les logiciels open source) ni la capacité de vérifier par lui même si sa confiance est bien placée. Une autre disposition intéressante des contrats de licence concerne le reverse engineering (ingénierie à rebours ;-)). Cette pratique qui pourrait sauver bien des situations est très souvent complètement proscrite par les éditeurs.

3.7.7 Conclusion

Les termes des contrats de licence permettent aux éditeurs de produire des logiciels sans réel engagement de qualité, engagement pourtant nécessaire à la fourniture de produits fiables. La concurrence et le marché sont les seuls éléments qui poussent encore les éditeurs à améliorer leurs produits. Pourtant au delà de l'assurance qualité qui déjà fait défaut, un engagement sur la sécurité et une responsabilisation des éditeurs seraient les éléments clés qui pourraient rendre les logiciels plus surs.

3.8 Les Classes de Fonctionnalité

Dans le but de normaliser les niveaux de sécurité auxquels peuvent prétendre les systèmes d'exploitations, la Defense Intelligence Agency (DIA) a créé l'Orange Book. Ce document propose 7 classes de fonctionnalités, ce qui nous donne dans l'ordre croissant D, C1, C2, B1, B2, B3, A1. A chacun de ces niveaux correspondent des fonctionnalités à respecter. Certains concepteurs de systèmes d'exploitation aiment à définir le niveau de sécurité que ceux-ci respectent grâce à ces classes de fonctionnalités. On trouve ainsi que le système NT 3.5 est à la norme C2, ce résultat est à relativiser car ce niveau n'est atteint que grâce à une configuration matérielle bien particulière (système coupé du réseau, pas de lecteur de disquettes) et que le système Solaris B2-compliant de SUN est au niveau B2, sans aucune contrainte.

3.8.1 Présentation de chaque niveau de fonctionnalité

Le niveau de sécurité D

Il correspond à un niveau de sécurité minimal, ce qui veut dire aucune contrainte, on y retrouve le système DOS.

Le niveau de sécurité C1

Il correspond à un niveau de protection discrétionnaire. C'est à dire que l'on assure la séparation des utilisateurs et des données avec la notion de sujet et d'objet, que l'on contrôle l'accès aux informations privées et que les utilisateurs coopèrent sur le même niveau de sensibilité.

Le niveau de sécurité C2

Ce niveau offre un accès contrôlé. C'est à dire que l'on affine le contrôle d'accès, au moyen du login et de l'audit. De plus, on sépare les ressources à protéger. Le passage de la classe C (protection discrétionnaire) à la classe B (protection mandataire) se fait par le biais de la labellisation des données.

Le niveau de sécurité B1

Il offre ainsi une protection par labellisation. On a donc une politique de sécurité associée au marquage des données et au contrôle obligatoire des sujets et des objets.

Le niveau de sécurité B2

Pour atteindre ce niveau dit "Protection Structurée", il faut renforcer le contrôle d'accès, ne pas disposer de canaux cachés, avoir une authentification renforcée, avoir des contraintes accrues de gestion et de contrôle de la configuration ainsi que disposer d'une TCB (Trusted Computing Base) fondée sur un modèle défini et documenté.

Le niveau de sécurité B3

C'est le niveau de " domaine de sécurité ", il faut obligatoirement un administrateur de sécurité, un audit accru et la TCB intervient lors de tout accès de sujet à objet, doit résister à l'intrusion et doit pouvoir être analysée et testée.

Le niveau de sécurité A1

Ce niveau est celui dit de "conception vérifiée". Comme son nom l'indique, il impose d'avoir une FTLS (Formal Top Level Specification) pour la conception de la TCB et du modèle. Il impose en plus des contraintes sur la gestion de la configuration et un administrateur de sécurité et du système distinct.

3.8.2 Réalisation d'un système TRUSTED (à haute sécurité)

Pour réaliser un système TRUSTED, on se base généralement sur la norme F-B1 de l'ITSEC (Critères d'évaluation de la sécurité des systèmes informatiques), norme elle-même dérivée des exigences fonctionnelles de la classe B1 du TCSEC (Trusted Computer System Evaluation Criteria) américain. Cette classe de fonctionnalité impose l'utilisation du contrôle d'accès discrétionnaire, introduisant en plus des fonctions de contrôle d'accès par mandat à tous les sujets et à tous les objets de stockage sous son contrôle. Il est aussi possible d'attribuer un label aux informations exportées.

Les systèmes de classe de fonctionnalité F-B1 sont plus sécurisés que ceux de classe F-C2 qui n'impose que l'identification du sujet, la possibilité de restreindre les actions du sujet à la lecture et à l'observation et à restreindre la transmission des droits à un groupe de sujets.

Pour en savoir plus sur le sujet nous vous suggérons d'aller lire : Le Compartemented Mode Workstation.

3.9 Le "Compartemented Mode Workstation"

Pour respecter les recommandations de la classe de fonctionnalité F-B1, il faut appliquer certains mécanismes comme ceux du CMW.

3.9.1 C.M.W. (Compartemented Mode Workstation)

Ce sigle fait référence à un mode de fonctionnement dans lequel toutes les informations sont labellisées par un niveau de sécurité (system_low, public, restreint, confidentiel, secret, top secret,mais en pratique, cela est paramétrable par l'utilisateur) et un ou plusieurs compartiments d'appartenance (projet A, dossiers X,..., on peut appartenir à plusieurs compartiments). Les spécifications CMW ont été faites en partie par la N.S.A. (National Security Agency) et Sun Microsystems. Ces documents sont difficiles à obtenir (pas de retour de requêtes effectuées chez SUN et la NSA). Toutefois, nous disposons de la documentation de Trusted Solaris qui est B1/CMW.

Dans un système CMW, toutes les informations sont labellisées. Tout objet portant des informations se retrouve donc "teinté" par ses informations et prend, à priori, le niveau de sécurité de celles-ci. Un objet peut créer des informations à son niveau et au-dessus, mais pas en dessous (problème de déclassification). Cette description de fonctionnement s'appelle un "modèle formel de politique de sécurité". Il en existe plusieurs, comme par exemple le modèle "Bell - La Padula"

En pratique, cela signifie que :

– Le système tourne à un certain niveau de sécurité. Ce niveau lui donne :

· Tout pouvoir pour manipuler les informations des utilisateurs, en pratique pour gérer le swap, permettre le scheduling, etc. Cela ne donne cependant pas accès aux informations des utilisateurs, simplement à leur manipulation.

· Une protection contre les modifications intempestives d'utilisateurs ou logiciel, que cela soit le fait de bugs ou de malversation. Exemple : tous les exécutables du système, ainsi que les fichiers de configuration, sont des fichiers SYSTEM_LOW, pour que personne ne puisse écrire dedans (les utilisateurs sont au minimum au niveau PUBLIC : Il ne peut pas "déclassifier" d'information au niveau SYSTEM_LOW). Par contre, les fichiers de logs sont au niveau SYSTEM_HIGH : tout le monde peut écrire dedans, et personne ne peut les lire (information confidentielle)

– Un utilisateur se connecte avec un certain nombre de privilèges. Il choisit à sa connexion :

· Le projet sur lequel il souhaite travailler,

· Son niveau de sécurité auquel il souhaite travailler, s'il dispose de plusieurs niveaux. Les niveaux sont peut-être fonction du projet choisi. Il peut peut-être choisir une fourchette de niveau,

· Aucun utilisateur n'a les pleins pouvoir sur un autre, à fortiori sur tous les autres. En conséquence : l'administrateur système doit gérer sa machine mais pas voir le contenu des fichiers utilisateurs (ou de la RAM qu'ils utilisent). L'officier de sécurité sert à gérer les droits. En pratique, l'administration de la sécurité s'effectue avec ses deux personnes qui disposent de droits spécifiques.

– Les utilisateurs peuvent posséder des droits particuliers nécessaires au fonctionnement correct du système. Ces droits leur permettent d'outrepasser certaines contraintes de la politique de sécurité. Ainsi, un officier de sécurité particulier peut se voir octroyer le droit de déclassifier des informations, c'est à dire passer une information de l'état CONFIDENTIEL à l'état PUBLIC par exemple. Ce genre d'action est bien évidemment audité. Un exécutable particulier peut avoir le droit d'outrepasser la politique de contrôle d'accès mandataire (les niveaux de sécurité) afin de pouvoir effectuer des sauvegardes des informations. Ces sauvegardes doivent contenir les informations de sécurité des fichiers, évidemment. Programmes tar, cpio, etc... à refaire. Certains droits ne doivent pas pouvoir être cumulés sur une seule personne. Exemple, la création et l'autorisation d'un compte utilisateur ne doivent pas pouvoir être donnés à un seul et même compte. L'administrateur créé le compte, met en place les quotas, l'environnement, et l'officier de sécurité lui donne ses droits.
– Il doit exister un chemin de confiance ("Trusted Path") entre l'utilisateur et la machine / le système d'exploitation. C'est une obligation du niveau B1. Ce TP permet à l'utilisateur de donner des informations relatives à la machine, en étant sur qu'il discute bien avec le système et pas avec un programme qui tenterait de se faire passer pour lui. L'accès au TP s'effectue généralement par une combinaison de touche (Ctrl-Alt-Del sous Windows NT par exemple) ou sur certaines zones de l'écran infalsifiables par les utilisateurs (zone de dessin/d'affichage inaccessible aux logiciels par exemple).

– · En ce qui concerne les informations labellisées, il faut tenir compte de tout. Voici une liste, probablement non exhaustive :

· Chaque fichier sur le disque possède ses informations de sécurité (propriétaire, niveau, compartiments),

· Chaque espace mémoire possède ses informations (idem fichier, plus processus propriétaire),

· Tout échange d'information est forcément contrôlé par le noyau : sur disque via le système de fichier, en mémoire (partagée) également, et par des sémaphores et signaux : labels à mettre en place. Un processus ne peut signaler un autre qu'à la condition qu'ils respectent des règles de sécurité définies. ·

– Tout accès réseau implique deux choses

Soit la connexion provient d'une machine de confiance (dialogue sécurisé entre les deux) auquel cas on conserve les informations de sécurité ;

Soit la machine est inconnue auquel cas on lui applique systématiquement un label de sécurité faible (NETWORK, juste au-dessus de SYSTEM_LOW par exemple). Se pose alors le problème du choix du compartiment auquel appartient le paquet. Il peut être décidé qu'un paquet "vierge" prend la teinte du premier processus qui le touche (a priori celui qui a reçu le paquet).

– Tout processus possède ses informations (idem fichier, plus niveau et compartiment courant suivant les fichiers accédés, plus droits d'origine et droits courant). Un logiciel peut se voir attribué par le système des droits particuliers lorsqu'il s'exécute (exemple : programme de sauvegarde qui outrepasse l'accès mandataire). Un tel programme doit provoquer ses droits.

Un point intéressant des systèmes CMW est justement le fonctionnement en compartiments. Un logiciel pour lequel on n'a aucune confiance peut être exécuté sans crainte dans un compartiment qui est propre. Si jamais le logiciel possède une faille, tout ce à quoi il aura accès (ou donnera accès) sera son propre compartiment qui, idéalement, ne contiendra que lui : pas de shell, pas de visibilité sur le système, etc., cela signifie également qu'un programme non prévu pour un système CMW doit, à priori, pouvoir tourner tel quel sur un tel système. Si le logiciel effectue des opérations particulières (écouter sur un port privilégié par exemple), alors il doit pouvoir être possible de lui donner les droits nécessaires à le faire.

Exemples de systèmes CMW :

– Trusted Solaris
– Sco Unix.

3.10 Le modèle "Bell La Padula"

Le modèle Bell La Padula définit l'état d'un système grâce à un "tuple" composé de 4 éléments : (b, M, F, H) (current access set, access permission matrix, level function and hierarchie).

Current Access Set (b)

C'est le mode d'accès de l'objet : execute, read, append, write.

Hierarchy (H)

Hiérarchie de l'objet dans un arbre de groupes, c'est à dire que l'on peut considérer l'ensemble des groupes comme une arborescence, chaque compartiment domine un ou plusieurs autres compartiments et est dominé par un compartiment. Cette organisation n'est pas obligatoire, chaque compartiment peut être totalement indépendant des autres, mais la propriété de dominant-dominé ne rend toutes ses qualités qu'utilisée dans ce contexte d'arborescence.

Access Permission (M)

Matrice associant les objets aux sujets en indiquant les droits qui s'y appliquent (b).

Level Function (f)

Niveau de sécurité de l'objet (Top Secret, Secret, Confidential, Unclassified).

On peut donc trouver un état comme celui-ci :

- Access set : Jean a le droit de lire le fichier du personnel. - Access Permission : la matrice a comme entrée, l'objet fichier du personnel et comme sujet Jean avec le droit en lecture.
- Level Function :Au maximum, jean a (SECRET, {STAFF, FINANCE}), le fichier du personnel a comme classification (CONFIDENTIAL, {STAFF}). Donc, le niveau courant de Jean sera (CONFIDENTIAL, {STAFF}).
- Hierarchy : l'objet est isolé.

Pour ce qui est de la communication entre deux machines, on a donc des propriétés à respecter :

- Il faut que le compartiment de l'information reçue soit un sous-groupe ou le même que celui de l'utilisateur.
- Il faut que le niveau de sécurité de l'information reçue soit inférieur ou égal à celui de l'utilisateur.

En fait, ces deux propriétés définissent bien que pour qu'un utilisateur ait accès à une information, il faut qu'il domine cette information. Ce qui signifie que la couche réseau va effectuer en réception le travail du système pour ce qui est de l'accès aux informations.

3.11 Que faire si vous êtes attaqué ?

3.11.1 Introduction

Vous détectez une activité anormale, une déconnexion, un ralentissement système... Après vérification, vous en êtes sûr, vous êtes attaqué. Voici quelques règles qu'il faut appliquer rapidement. Ces règles dépendent de vous : vous ne réagirez pas de la même façon si vous êtes un particulier qui surfe sur le web, ou un administrateur réseau en entreprise.

3.11.2 Les règles communes

Dans tous les cas, appliquez ces règles :

- La première règle : être rapide ! Une attaque n'est souvent qu'une affaire de secondes, voire de minutes. Le but du hacker est d'arriver à ses fins le plus rapidement possible.
- Ne pas contre-attaquer le hacker. Il réagirait de deux façons différentes si vous contre-attaquez : Il se rend compte qu'il a été repéré, et il quitte rapidement le terrain, réduisant d'autant vos chances de savoir qui il était et ce qu'il voulait. Ou alors, il attaque de plus belle et, s'il est plus fort que vous, vous avez tout à y perdre : un hacker énervé qui pénètre dans votre système risque de faire beaucoup de dégâts...
- Notez l'adresse IP de l'ordinateur victime de l'attaque.
- Notez l'heure de l'attaque.
- Notez le temps de l'attaque.

3.11.3 Vous êtes un particulier

Un hacker qui s'attaque à un particulier, ce n'est pas courant. Mais cela arrive : c'est certainement un hacker débutant qui veut se faire la main, sans pour autant prendre le risque de s'attaquer à une société commerciale... Il veut généralement vous déconnecter, faire planter votre ordinateur, ou le ralentir. Néanmoins, ce sont certainement les plus dangereux, car ce sont ceux qui ont le moins d'éthique. Ce sont ces débutants qui prennent plaisir à effacer le contenu de votre disque dur...

L'attitude à prendre dépend de vous :

- Vous n'y connaissez rien en sécurité informatique : Coupez immédiatement votre modem : à la reconnexion, votre adresse IP changera (adresse IP dynamique), car c'est votre fournisseur d'accès qui vous l'attribue. Le hacker aura du mal à vous retrouver. Ne lancez pas ICQ, car

on peut voir si vous êtes online. Pire encore, on peut obtenir votre adresse IP, même si vous avez coché la case "Masquer l'adresse IP". N'allez plus sur IRC, du moins pendant quelques heures.
- Vous connaissez quelques trucs en sécurité informatique : Essayez de comprendre l'attaque : si vous comprenez ce que le hacker fait, vous pourrez peut-être trouver une solution pour ne plus être attaqué de cette façon à l'avenir. Ayez toujours un doigt sur le bouton "Reset" de votre ordinateur, car vous prenez d'énormes risques à laisser faire le hacker...
- Vous êtes un gourou de la sécurité informatique : Vous pouvez appliquez les mêmes règles que celles établies pour les administrateurs réseau.

3.11.4 Vous êtes un administrateur réseau

- Prévenez immédiatement votre supérieur hiérarchique.
- Essayez d'obtenir l'adresse IP du hacker.
- Faites un "tracert" pour connaître la localisation la plus précise possible du hacker.
- Essayez de savoir ce qu'il fait : Est-ce un simple flood ou une tentative d'intrusion de votre réseau ? Scannez vos ports. Utilise-t-il un troyen ? Essayez de trouver s'il n'y en a pas un, à l'aide d'un antivirus, ou autre. Enregistrez les packets d'attaques avec un analyseur de protocole. Notez tout ce qu'il fait.

3.11.5 L'attaque est finie

Il y a quatre possibilités :
- Le hacker n'a pas réussi ce qu'il voulait et vous n'avez pas su ce qu'il faisait (vous avez coupé rapidement votre modem par exemple) : l'incident est clos, mais restez sur vos gardes un moment.
- Le hacker a réussi, mais vous n'avez rien récupéré sur lui et sa méthode : c'est la pire des choses. Faites appel à un consultant, un expert, pour sécuriser au mieux votre réseau.
- Le hacker n'a pas réussi, et vous avez noté plein de renseignements sur lui et sa méthode : faites attention, il pourrait revenir avec une autre méthode. Configurez votre firewall pour ne plus accepter de packets de l'adresse IP de l'attaquant. Prenez un maximum de mesures en prévention.
- Le hacker a réussi, et vous avez noté sa méthode : il est grand temps de trouver une solution pour palier à ce trou de sécurité. Prenez un maximum de mesures en prévention.

3.11.6 Dans tous les cas

Si le hacker :
- A pénétré votre réseau, ou votre ordinateur,
- A dégradé matériellement votre ordinateur (il est possible, par logiciel, de dégrader du matériel),
- A modifié des données,
- A effacé des données,
- A mis en place un cheval de Troie,
- A mis en place un virus ou un ver,
- Vous a porté un préjudice financier,
- Etc (liste non exhaustive).

Portez plainte ! Chacune de ces actions est punissable par la Loi. Il est évident que plus vous avez de renseignements sur le hacker, et plus il sera facilement identifiable.

3.12 Principes de sécurité informatique et évolution du marché.

3.12.1 Introduction

On retrouve actuellement trop souvent des architectures de sécurité axées uniquement sur la prévention et la défense de périmètre. Il y a bien d'autres éléments qui doivent composer une architecture de sécurité. Toute architecture de sécurité (et plus globalement l'approche même de la sécurité) doit selon moi reposer sur un triptyque tel que :

- Prévention
- Détection
- Réaction

Ces trois aspects sont pour le moment très diversement couverts par le marché malgré une nécessité indéniable.

3.12.2 Prévention

La prévention est fondamentale et est généralement bien appréhendée par le plus grand nombre. Le principe : faire tout ce qu'il faut pour se protéger. Elle consiste le plus souvent à adopter la démarche suivante : Analyse des risques Définition d'une politique de sécurité Mise en uvre d'une solution centrée sur un ou plusieurs firewalls. Audit de la solution Mises à jour Le marché à ce jour couvre très bien cette approche : les cabinets de conseils sont très présents sur l'analyse des risques. Les Intégrateurs proposent et mettent place des solutions à tour de bras. Des sociétés se spécialisent dans la réalisation d'audits de sécurité, d'autres effectuent de la veille technologique en sécurité et permettent de déclencher les mises à jour (généralement effectuées par l'intégrateur).

3.12.3 Détection

Le principe est d'être capable de détecter lorsque les mesures de prévention sont prises en défaut. La détection, même si certains outils techniques existent, est encore trop rarement intégrée aux infrastructures. Il est vrai que les intégrateurs proposent souvent ces outils lors de la mise en place d'infrastructures de connexion Internet ; mais leur déploiement reste marginal en dehors de ces projets spécifiques. De plus, à l'heure actuelle un cruel défaut de compétence est à déplorer. Il y à encore trop peu de personnes formées à ce type d'outils. La détection exige un suivi permanent de l'état des système à protéger et des mécanismes de diffusion des alertes générées.

3.12.4 Réaction

S'il est important de savoir qu'une attaque est en cours ou qu'une attaque a réussi il est encore plus important de se donner les moyens de réagir à cet état de fait. C'est l'aspect le plus négligé actuellement même au sein des acteurs majeurs de la sécurité Informatique. Pourtant, il n'est pas possible d'oublier les credo de tous les consultants en analyse de risque : " le risque zéro n'existe pas " ou encore " il n'y a pas de sécurité absolue ". Il faudrait donc toujours prévoir et se préparer au pire. Cela implique la mise en uvre de procédures d'exploitation spécifiques à la réaction en cas d'attaque, la rédaction et le test d'un plan de continuité Informatique à utiliser en cas de sinistre grave. Il est également primordial de se doter des outils permettant d'une part de collecter toutes les informations pouvant être nécessaires en cas de recours juridique. Un cadre doit aussi être prévu au niveau des responsabilités ; de ce fait les contrats d'assurance devront prendre en compte le risque représenté par les pirates. Le marché couvre très mal cet aspect à l'heure actuelle. Il n'existe que très peu de sociétés proposant une offre réelle en investigation d'incidents. Par ailleurs, même si certains cabinets de juristes se spécialisent dans le droit de l'Internet, la couverture du risque Informatique et la définition des " éléments de preuve " dans les affaires de crimes informatiques restent encore floues.

3.12.5 Conclusion

La prise en compte des problématiques de sécurité est en cours en France actuellement dans une vaste majorité d'entreprises mais pour l'heure les moyens mis en uvre ne sont pas toujours suffisants. Afin de supporter les entreprises dans leur processus de sécurisation, le marché, en forte croissance, s'est d'abord structuré dans le domaine de la Prévention. Néanmoins, de très nombreuses questions restent encore sans réponse dès lors qu'il s'agit de Détection et de Réaction. Ces deux domaines qui touchent à l'exploitation au quotidien (ou encore opération) des infrastructures de sécurité sont encore pleins de promesses mais aussi sources d'inquiétude pour les différents acteurs de la sécurité informatique.

3.13 Les Sauvegardes

3.13.1 Introduction

Que vous soyez un particulier ou une entreprise, une sauvegarde peut vous tirer d'affaire dans bien des cas. Que vous soyez victime d'une attaque, d'un crash système, d'une défaillance matérielle, etc. ; seule une sauvegarde vous permettra de restaurer entièrement le système dans son état originel. Encore faut-il qu'elles soient bien faîtes ! Simplement, il reste difficile de faire un choix approprié dans la jungle des choix disponibles dans le monde de la sauvegarde. C'est le but de cet article : vous aiguiller dans votre choix.

3.13.2 Les critères de choix

Il est important dans un premier temps de se définir une politique de sauvegarde, un budget et ensuite le choix viendra de lui-même. Une façon d'évaluer le budget à accorder aux sauvegardes est d'estimer les pertes subies en cas d'immobilisation. L'essentiel réside dans le repérage des données à sauvegarder. Tout n'est pas, ni important à sauvegarder, ni modifié à chaque instant. Par exemple, il peut être judicieux d'effectuer une sauvegarde quotidienne des données importantes et une sauvegarde mensuelle (si possible bootable) du système. La sauvegarde bootable du système permettra une restauration automatique de celui-ci. La sauvegarde des données permettra leur restauration à tout instant.

3.13.3 La politique de sauvegarde

Tout dépend du rythme de modification de vos données. Un serveur bancaire national supportant plusieurs milliers d'écritures à l'heure n'aura pas les mêmes besoins en terme de sauvegarde qu'un serveur d'applications modifié une fois par mois... Une fois le rythme des sauvegardes évalué, il ne vous reste plus qu'à déterminer le type de sauvegarde. Il existe principalement 3 type de sauvegardes :

- la sauvegarde totale : l'ensemble des fichiers, répertoires, systèmes de fichiers ou disques sélectionnés est sauvegardé sans restriction.
- la sauvegarde incrémentale : tous les fichiers modifiés depuis la dernière sauvegarde totale sont sauvegardés.
- la sauvegarde différentielle : tous les fichiers modifiés depuis la dernière sauvegarde différentielle sont sauvegardés.

L'utilisation de plusieurs bandes est primordiale ; d'une part, pour en éviter l'usure et, d'autre part, pour supprimer le risque de tout perdre en cas de détérioration de celles-ci. Il est courant de rencontrer la politique suivante (16 bandes) :

- Une sauvegarde totale dans la nuit du vendredi au samedi.
- Une sauvegarde incrémentale les autres nuits.
- Une sauvegarde système une fois par mois.
- La bande du vendredi est conservée 1 mois comme sauvegarde hebdomadaire.
- La bande du dernier vendredi du mois est conservée 1 an comme sauvegarde mensuelle.

– La bande du dernier vendredi de l'année est conservée sans limitation dans la durée comme sauvegarde annuelle.

On aura ainsi besoin de 5 bandes hebdomadaires + 11 bandes supplémentaires pour chaque mois soit 16 bandes.

Vous l'aurez compris, il faudra fixer un lieu de stockage pour ces bandes. Il est de bon ton de les conserver dans un endroit à l'abri du feu et des inondations. L'idéal étant de les conserver dans un coffre ignifugé, ainsi qu'une copie de ces bandes sur un site distant (Ce qui porte à 32 le nombre des bandes).

Outils de planification de sauvegarde

Une fois la politique de sauvegarde choisie, le choix de l'outil dépendra de vos affinités avec tel ou tel éditeur. Tous les logiciels de sauvegarde ont à peu près les mêmes caractéristiques et possibilités.

Un des critères pourra être la plateforme système que vous utilisez. Certains logiciels tournent mieux sur certaines plateformes. Dans le cas d'Unix, n'oublions pas non plus les scripts (par exemple : ufsdump lié à cron) qui bien utilisés vous permettront de développer les mêmes fonctionnnalités.

3.13.4 Les différents supports de sauvegarde

De la simple disquette au disque optique ou aux bibliothèques de sauvegarde, le choix ne manque pas. On s'y perd allègrement d'autant que les technologies évoluent sans cesse... Tout dépendra une fois encore du budget que vous comptez accorder à votre système de sauvegarde Malgré tout, la fiabilité et la capacité des supports restent un choix des plus objectifs.

Voici un tableau non exhaustif qui vous présente les types de supports les plus connus ainsi que leurs caractéristiques principales :

Nom	Capacité	Débit	Technologie	Fiab
Disquette	de 1,44Mo à 2,88Mo	jusqu'à 0,5Mo/s	Magnétique	La p
ZIP	2 formats disponibles : 100Mo ou 250 Mo	de 0,6Mo/s (port parallèle) à 2,4 Mo/s (en ATAPI interne pour le modèle 250)	Magnétique	Peu
JAZZ	2 formats disponibles : 1Go ou 2Go	de 5,5Mo/s à 8Mo/s	Magnétique	Peu
Disque Dur	Plusieurs Go	selon interface (IDE ou SCSI et technologie associée)	Magnétique	Asse
QIC : Quarter Inch Cartridge	de 250Mo à 8Go (compressé)	jusquà 0,8Mo/s	Magnétique	Asse
DAT : Digital Audio Tape	de 2 à 24 Go (selon lecteur)	1,5Mo/s taux de compression de 2 pour 1	Magnétique	Asse assez les t pour
DDS : Digital Data Storage (évolution du DAT)	jusqu'à 40 Go	jusqu'à 4,7Mo/s	Magnétique	Bonn gara cont
DLT : Data Linear Tape	10 Go en natif	3Mo/s taux de compression de 2 pour 1	Magnétique	Bonn moin pour pass
SDLT : Super Data Linear Tape nouvelle technologie	de 110Go à 1,2To (non compressés)	de 11Mo/s à plus de 100 Mo/s	Magnéto-optique (guidage des têtes)	Bonn
CD-ROM	de 600 à 700Mo (nb : une nouvelle technologie permettrait de plus grandes capacités)	multiple de 153,6Ko/s (1x)	Optique	Très
Disque Magnéto-Optiques	de 128Mo à 1,3Go	jusqu'à 5,9Mo/s	Optique	Exce
DVD : Digital Versatile Disk	à partir de 3,2 Go	Multiple de 1,35Mo/s (1x)	Optique	Exce

Ce tableau ne mentionne pas les bibliothèques de sauvegarde pouvant contenir plusieurs centaines de bandes et pilotées par le logiciel de sauvegarde.

Aucune mention n'a été faîte des SANs (Storage Area Network) qui, bien que technologie de sauvegarde, rentrent davantage dans l'architecture réseau de l'entreprise. Mais il ne faut pas ignorer cette technologie pour les plus grosses entreprises ayant besoin de sauvegardes à la volée et ne souhaitant pas surcharger les machines et le réseau.

3.13.5 Conclusion

Les sauvegardes font partie de manière plus globale de la politique de sécurité des données. Que ce soit suite à un crash système, un crash matériel ou une infiltration malveillante (hack), une sauvegarde peut vous faire économiser parfois tout un mois de travail. Il ne faut pas ignorer cet aspect de la sécurité... Il suffit d'une fois...

3.14 La sécurité et l'Open Source

3.14.1 Introduction

Depuis Linus Torvalds et son système Linux, l'Open Source s'est considérablement développé. Mais qu'est-ce que l'Open Source ? C'est le fait de rendre public le code source d'un logiciel. L'Open Source est régie par un ensemble de licences, dont la plus connue est la GNU Public License. Ce code source n'est donc plus la possesion privée d'une personne, d'un groupe de personne, ou d'une société, comme c'était le cas depuis la naissance de l'informatique dans les années 60, jusque dans les années 80/90. Les plus grandes entreprises emboitent actuellement le pas des développeurs indépendants et proposent à leur tour des logiciels de qualité professionnelle en Open Source. Mais derrière cet effervescence intellectuelle, quelles sont les conséquences, en matière de sécurité, pour les projets Open Source ?

3.14.2 Les avantages

Relectures multiples du code

Qu'il soit étudiant, professionel, ou tout simplement amateur et quelque soit son niveau, ses méthodes, sa culture, sa nationalité, le programmeur a accès au code. Il peut donc le relire pour le comprendre et anticiper le debuggage. De ces lectures croisées de nombreux bugs sont décelables. Parmis ces bugs, il y en a certainement qui touchent directement la sécurité du logiciel, comme les buffers overflow. On appelle cela des trous de sécurité applicatif.

Réactivité de l'open source

Un autre avantage de l'open source est le fait que la communauté réagisse plus rapidement dans la correction d'un bug. Cela arrive même fréquemment que le programmeur qui découvre un bug propose aussi le patch qui permette de le corriger, lorsque l'information est rendue publique. Les sociétés traditionnelles de logiciel mettent plus de temps car leur structure est plus hiérarchisée, plus grosse et donc moins réactive.

3.14.3 Les inconvénients

Relectures multiples du code

La relecture multiple du code permet de détecter un plus grand nombre de trous de sécurité dans un logiciel. Par contre, il serait naif de penser que tous les trous de sécurité sont vus ! Les logiciels sont de plus en plus complexes et certains dépassent même la vision que peut avoir un programmeur de l'ensemble du logiciel. C'est le cas de Linux : Les programmeurs se cantonnent à écrire des patchs, au mieux des modules, et rares sont ceux qui ont une vision globale de tous les morceaux de code que composent le kernel Linux. Si on ajoute à cela qu'il existe un noyau Linux pour chaque OS, et que chaque OS a un comportement différent en matière de sécurité dans la programmation... De surcroit, à l'échelle de tous les programmeurs dans le monde, il existe peu de personnes qualifiées pour faire une relecture de code open source, à la recherche de trous de sécurité. Il est donc tout à fait concevable qu'un trou de sécurité existe dans un logiciel open source et que personne ne le découvre avant des mois, voir des années.

L'open source est... Open !

Le fait de mettre le code accessible à tout le monde est risqué : Si une personne découvre un trou de sécurité, rien ne l'empêche de le garder pour lui en vue d'en tirer un profit quelconque. Un trou de sécurité ne peut être corrigé que s'il est connu. Donc tant que des hackers gardent leurs informations pour eux, le logiciel cible ne sera pas corrigé. Et cela peut prendre des mois.

3.14.4 Conclusion

Tout le monde sait que baser la sécurité sur un programme propriétaire n'est pas sécurisant : N'importe quel hacker peut désassembler le code jusqu'à comprendre comment la protection est faite. C'est un fait. C'est pourquoi l'open source est généralement considéré comme plus sécurisant qu'un code propriétaire. Comme nous l'avons vu, il n'en est rien. Le seul fait qui aille dans le sens de l'open source, c'est qu'un bug de sécurité est en général plus rapidement découvert, et donc plus rapidement corrigé.

3.15 Méthode d'Audit de Sécurité Informatique dans l'Entreprise : La méthode FEROS.

3.15.1 Introduction

La plupart des entreprises a encore des difficultés à concevoir que leur sécurité peut être défaillante. Ceci est particulièrement vrai dans un contexte de PME-PMI. Chaque structure préfère croire que les pertes de données n'arrivent "qu'aux autres" et argue souvent que les audits effectués par des sociétés spécialisées type SSII sont trop onéreux pour leur budget. Malheureusement, les dirigeants ou responsables informatiques oublient de se poser une question cruciale : "combien cela me coutera-t-il si l'informatique de mon entreprise, ma base de donnée client ou ma comptabilité par exemple, disparaissait ?". En effet, cette question devrait pouvoir sensibiliser n'importe quel décideur responsable, et l'amener à faire valider la cohérence de ses systèmes d'informations et de sauvegardes. Il existe aujourd'hui sur le marché différentes méthodes permettant d'auditer de manière fiable et autonome une entreprise. Cependant, il faut garder en tête que faire appel à des professionnels serait plus judicieux. Dans tous les cas, voici une synthèse des quatres principales méthodes disponibles sur le marché :

- La méthode Feros
- La méthode Méhari
- La méthode Marion
- La méthode Melissa.

3.15.2 La méthode FEROS

La méthode FEROS signifie : Fiche d'Expression Rationnelle des Objectifs de Sécurité des Systèmes d'Information. Elle part du constat simple que les Responsables Informatiques sont généralement chargés de veiller sur :

- le bon fonctionnement des matériels et réseaux de communication de l'entreprise,
- l'intégrité des données qui circulent, sont sauvegardéses et archivées.

Partant de ce constat, il est donc primordial de défionir des objectifs de sécurité à mettre en oeuvre dans l'entreprise. Pour ce faire, il faut commencer par déterminer les besoins de la structure auditée :

- identification des données cruciales,
- détermination d'un seuil de tolérance de disponibilités ou innaccessibilité des dites données
- identification des impératifs légaux incontournables qu'il faut respecter.

En parallèle de ces objectifs doivent se trouver les contraintes incontournables que rencontre la société :

- contraintes matérielles,

- contraitres de limitation de savoir faire en interne,
- contrainte de disponibilités des ressources hommes etc.

C'est en confrontant les besoins de l'entreprise en matière de sécurité avef les contraintes auxquelles elle doit faire face qu'il faudra déterminer la politique de sécurité à mettre instaurer. Une fois les grands axes de cette politique définis, il est important selon la méthode FEROS de s'interroger sur les menaces que l'entreprise peut rencontrer :

- piratage amateur à vocation ludique,
- piratage professionnel à la solde de la concurrence,
- piratage d'un personnel interne en colère contre l'entreprise
- etc .....

La méthode FEROS s'articule autour de quatre axes principaux :

- un guide permettant à chaque strucutre auditée de rédiger un questionnnaire qui lui sera propre,
- le questionnaire structuré qui permettra de mettre en évidence les particularités de l'entreprise,
- un glossaire qui défénira précisement le sens de chaque terme technique employé pour le vulgariser auprès des décideurs non techniques,
- une synthèse des menaces potentielles qui permettra d'en prévoir les parades.

Cette méthode émane du SCSSI et vous la retrouverez intégralement sur leur site.

3.15.3 Notre opinion

Cette méthode possède le gros avantage d'être simple d'appréhension pour un non initié à la technique. Fonctionnnelle et structurée cette méthode permet de réagir rapidement et pourquoi pas de préparer le travail d'une société extérieure qui sera chargée d'approfondir chaque point soulevé par l'application de cette méthode. En effet, elle nous semble êter un bon moyen de "préparer le terrain" et donc réaliser des économies substantielles en ne faisant appel à des spécialistes qu'une fois le terrain déblayé.

3.16 La translation d'adresse (NAT)

3.16.1 Overview

La technique de translation d'adresses (NAT en anglais, RFC 3022) est une pratique courante qui est apparue à l'origine pour palier au manque croissant d'adresses IPv4 libres. En effet, ces adresses sont codées sur 4 octets et sont du type 0.0.0.0 à 255.255.255.255 (certaines valeurs étant réservées et par conséquence inutilisables) ; il y a donc peu d'adresses disponibles en comparaison du nombre croissant de machines sur Internet. Il fut donc décidé de réserver des intervalles d'adresses à des usages privés uniquement (RFC 1918). Ce sont les adresses :

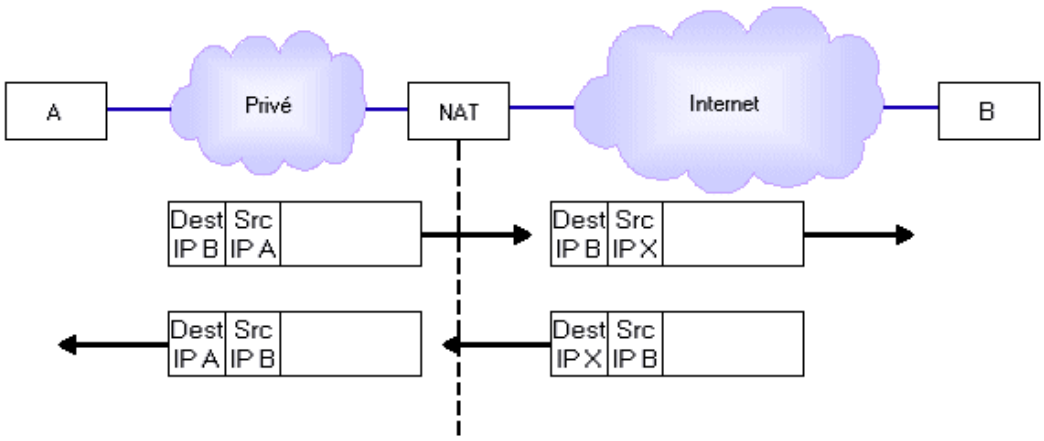
10.0.0.0 - 10.255.255.255 (10/8 prefix)

172.16.0.0 - 172.31.255.255 (172.16/12 prefix)

192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

En conséquence, ces adresses ne sont pas routables sur Internet et ne doivent pas être utilisées par des machines de ce réseau. Par contre, tous les réseaux privés peuvent utiliser ces adresses sans restrictions.

Comme ces adresses ne sont pas routables sur le réseau public, la translation d'adresse est utilisée pour permettre aux machines du réseau privé d'accéder à Internet, et de façon générale à d'autres réseaux. Le principe de base est simple puisqu'il s'agit de remplacer à la volée les champs d'adresses dans les paquets qui sont destinés à un autre réseau (ce qui implique que le NAT soit effectué entre les 2 interfaces réseau, entre le réseau privé et les autres).



Comme le montre le schéma, le NAT va effectuer le remplacement de l'IP source de A par son IP X puis il va router le paquet vers le réseau extérieur. La réponse lui parviendra, et suivant la technique utilisée que nous allons détailler plus loin, il va cette fois-ci modifier l'adresse de destination X pour celle de A sur son réseau privé.

3.16.2 Techniques de translation

Il existe plusieurs variantes de NAT suivant la topologie du réseau privé, le nombre de machines privées, le nombre d'adresses IP publiques et les besoins en termes de services, d'accessibilité et de visibilité de l'extérieur.

- Le NAT de base est statique et attribue de façon automatique une adresse IP à une autre. Aucune information liée à la connexion n'est nécessaire, il suffit de modifier le paquet suivant la règle prédéfinie de translation. L'idéal dans ce cas-ci est d'avoir le même nombre d'IP extérieures que d'IP privées.
- Le NAT dynamique ne considère aucune association prédéfinie entre l'IP publique et l'IP privée de la requête qu'il reçoit. Il peut d'ailleurs y avoir plusieurs IP extérieures tout comme il y a plusieurs IP privées. Cela entraine nécessairement un suivi de la connexion car le NAT attribue l'IP extérieure lors de la requête initiale qui provient de son réseau privé ; il doit ensuite pouvoir discriminer les paquets entrants de façon à pouvoir leur attribuer à chacun l'IP correspondante sur le réseau privé (celle de la connexion). Le but étant de rester transparent vis-à-vis de l'ordinateur source ou distant ; un problème se pose si l'on ne dispose pas du même nombre d'adresses IP externes que d'adresses privées, car si toutes les adresses externes sont déjà en cours d'utilisation, aucune machine supplémentaire ne pourra accéder au réseau extérieur.
- Le NAPT MASQ (Network Address and Port Translation) permet de résoudre le problème cité précédement et s'avère donc particulièrement utile si le nombre d'adresses externes est limité ; c'est le cas typique d'une connexion Internet simple où plusieurs machines vont devoir partager la même adresse IP publique (externe). Le problème technique derrière cette méthode est bien de savoir à quelle machine privée les paquets entrants sont destinés, puisqu'ils ont tous -à priori- la même adresse IP de destination (celle de la passerelle). Pour permettre leur différenciation, le NAT va devoir conserver une trace plus complète des paramètres de chaque connexion de façon établir un véritable contexte pour chacune de ces dernières. Parmi ces critères de séparation, citons :

l'adresse source est le premier élément qui est regardé ; chaque machine du réseau privé aura tendance dans la majorité des cas à communiquer avec une machine extérieure différente. Donc les paquets entrants seront porteurs de cette information et permettront au NAT d'identifier la

machine à l'origine de chaque échange. Mais cela ne fonctionnera pas si les machines extérieures ne sont pas toutes différentes.

le protocole supérieur peut également être régardé par le NAT pour pouvoir identifier le contexte. Ce sera par exemple de l'UDP ou du TCP, et si une machine utilise le premier et une autre utilise TCP, alors le NAT saura retrouver la machine initiale de la connexion.

le port et d'autres informations liées aux protocoles supérieurs peuvent être utilisés pour identifier chaque contexte. Ainsi le NAT pourra faire la différence entre des paquets entrants qui présentent la même IP source, le même protocole de transport mais un port de destination différent.

Il reste un dernier cas dans lequel tout cela ne suffira pas, c'est celui où les 2 contextes basés sur ces informations sont identiques, c'est-à-dire quand les paquets entrants présentent la même IP source, le même protocole de transport et le même port de destination. Dans ce cas-là, le NAT effectue une translation de port en même temps que d'adresse pour pouvoir identifier les flux de façon certaine. Cela consiste à modifier les paramètres de connexion avec la machine distante de façon à utiliser le port voulu sur la passerelle où se situe le NAT. Cette opération reste transparente pour la machine locale (privée) puisque cela est effectué au niveau du NAT qui rétablit ensuite les paramètres initiaux pour cette machine.

Comme il existe 65535 ports disponibles (moins les 1024 réservés), cela laisse une grande marge de sécurité. La dénomination MASQ provient du fait que cette opération est comparable à une attaque du type man-in-the-middle sauf qu'elle ne vise pas à obtenir quelqu'information que ce soit (la législation à ce niveau est stricte, voir les recommandations de la CNIL).

- Le NAPT Redirect/Port Forwarding est identique au précédent sauf qu'il présente des services additionnels de redirection des flux entrants ou sortants. Ainsi, le Port Forwarding permet à l'extérieur d'accéder à un service (serveur WEB ou autre) qui est en fait basé sur une machine de réseau privé : la machine distante pense communiquer avec la machine hébergeant le NAT alors qu'en fait celui-ci redirige le flux vers la machine correspondant réellement à ce service. Le Redirect permet quant à lui de rediriger les flux sortants vers des services particuliers commes des proxies, firewalls, etc...
- Le Bi-directional NAT diffère des précédents puisqu'il permet à des machines distantes d'accéder à des machines du réseau privé, et ce directement contrairement au Port forwarding. Le principe fait appel au service DNS pour interpréter les requêtes ; celles-ci sont initiées par la machine distante et reçues par le NAT. La passerelle répond par sa propre adresse IP tout en gardant en mémoire l'association entre l'IP distante et l'IP requise pour le service. Ainsi, les paquets provenant de la machine distante seront transférés vers la machine correspondante. Le problème de cette technique est l'utilisation du service DNS qui peut être couteux dans le cas d'un utilisateur de base accédant au réseau public Internet. Par contre, cette solution pourra se révéler utile dans le cas d'une entreprise interconnectant plusieurs réseaux privés car les serveurs DNS sont alors mieux maitrisés.
- Le Twice-NAT est, comme son nom l'indique, une technique de double translation d'adresses et de ports. A la fois les paramètres de destination et ceux de la source seront modifiés. Concrètement, on peut dire que le NAT cache les adresses internes vis-à-vis de l'extérieur ainsi que les adresses externes vis-à-vis du réseau privé. L'utilité de cette technique apparait quand plusieurs réseaux privés sont interconnectés : comme nous l'avons expliqué précédement, les machines doivent utiliser des plages d'adressage bien précises ce qui peut créer des conflits et des collisions entre plusieurs réseaux privés (c'est-à-dire plusieurs machines utilisant la même adresse IP privée). Le Twice-NAT permet de résoudre ces problèmes de collisions en modifiant les 2 adresses du paquet.
- Le NAT avec Serveurs Virtuels/Load Balancing est une évolution des techniques de NAT qui permet d'optimiser leurs implémentations. L'utilisation de serveurs virtuels est actuellement très répandue ; cela correspond à une machine inexistante représentée uniquement par son adresse IP et prise en charge par une ou plusieurs machines réelles qui ont également leurs propres adresses (différentes). Ainsi, les requêtes des machines distantes sont adressées à une ou plusieurs adresses virtuelles correspondant à la passerelle où est implémenté le démon effectuant le NAT. Celui-ci remplace alors l'adresse virtuelle par une des adresses réelles

appartenant aux machines implémentant le service NAT, puis leur transmet la requete et la connexion associée. La sélection de l'adresse réelle peut se faire sur la base de la charge de travail de la machine correspondante : si le serveur NAT est surchargé, on choisira un autre serveur NAT moins chargé. Cette technique est à la base du load balancing et il existe de nombreux algorithmes de sélection et de répartition de la charge. Enfin, comme le service NAT est habituellement placé sur la machine chargée du routage, une évolution possible est l'utilisation de routes virtuelles tout comme nous avons vu les adresses virtuelles. Dans ce dernier cas, la passerelle possède plusieurs interfaces vers le réseau externe et peut choisir laquelle utiliser en fonction de la charge de traffic sur chaque brin.

3.16.3 Avantages et inconvénients

N'oublions pas que l'utilité principale du NAT est d'économiser les adresses IP nécessaires pour connecter un réseau à Internet par exemple. Cela s'avère particulièrement utile pour tout particulier possédant une connexion Internet simple (modem, ADSL, cable) avec allocation d'une adresse dynamique. Si ce particulier possède plusieurs machines sur son réseau privé, il pourra utiliser la fonctionnalité de NAT pour partager l'adresse IP de sa machine principale.

D'autre part, les fonctionnalités avancées du NAT permettent d'interconnecter plusieurs réseaux privés de façon transparente même s'il existe des conflits d'adressage entre eux.

Par contre, dans la majorité des techniques citées précédement, la connexion est nécessairement initiée à partir d'une machine locale. Les machines externes ne verront que l'adresse de la passerelle et ne pourront pas se connecter directement aux machines locales ; cela est bien sûr résolu avec les techniques plus évoluées de translation, mais celles-ci restent couteuses et peu accessibles.

Enfin, l'opération même de translation peut poser un certain nombres de problèmes que nous allons aborder dans le paragraphe suivant.

3.16.4 Sécurité et NAT

Le NAT présente à la fois des inconvénients et des avantages au niveau de la sécurité pour les machines du réseau privé.

Tout d'abord, comme nous l'avons vu précédement, le NAT n'est pas une opération anodine et ce bien qu'il ait pour vocation d'être transparent. En effet, le NAT modifie les paquets IP et cela a pour conséquence directe de casser tout contôle d'intégrité au niveau IP et même aux niveaux supérieurs puisque TCP par exemple inclue les adresses dans ses checksums ! Concrètement, on se rend compte qu'un protocole de sécurisation des datagrammes comme IPSec est totalement incompatible avec le NAT, que ce soit en mode tunneling ou transport (voir fiche IPSec).

Une autre raison simple est qu'un NAT évolué a tendance à remonter les couches pour étudier les protocoles de transport afin de rassembler assez d'informations pour chaque contexte. Tout chiffrement à ce niveau empêcherait donc le NAT de fonctionner, puisque les informations seraient alors cryptées.

Un des avantages du NAT est de protéger les machines du réseau privé d'attaques directes puiqu'elles ne sont en fait pas accessibles de l'extérieur. De plus dans la majorité des cas, les requêtes de connexion ne peuvent provenir que de ces machines privées. Cela permet également de se prémunir contre un monitoring du traffic qui viserait à scruter les communications entre 2 machines particulières, un serveur sur Internet par exemple et une machine du réseau privé. Comme cette dernière n'est plus identifiable, l'opération devient impossible à moins de remonter au niveau applicatif (d'où l'utilité d'utiliser une protection/chiffrement à ce niveau également).

3.16.5 Conclusion

Le NAT est aujourd'hui incontournable dans la plupart des topologies réseau, à partir du moment où l'on souhaite connecter le réseau à d'autres. Comme nous l'avons vu, les techniques correspondant au service NAT ont évolué pour répondre aux besoins croissants de transparence, connectivité, disponibilité, etc... Quoiqu'il en soit, l'utilisation d'une telle technique ne doit pas

être prise à la légère car elle implique autant d'inconvénients que d'avantages. Enfin, on peut s'interroger sur la pérennité du NAT sachant que cette technique n'était à l'origine destinée qu'à palier les lacunes d'IPv4. Or, il y a fort à parier qu'elle sera toujours effective avec les nouvelles adresses IPv6, autant à cause de ses qualités de sécurisation que du fait de la lenteur prévisible de la migration des terminaux d'un système d'adressage à l'autre.

Chapitre 4

Divers

4.1 Qui se cache derrière Securiteinfo.com ?

Nous sommes une petite équipe de bénévoles. Nous prenons sur notre temps libre pour développer ce site web.

4.1.1 scrap aka Arnaud Jacques

Signes particuliers : Créateur de Securiteinfo.com. Ingénieur Sécurité et Réseaux
Articles publiés sur le site web :
- Hacking : Qu'est-ce que c'est ?
- Types d'attaque
- Challenges de hacking
- Ath0
- Boink
- Bonk
- BrKill
- Cisco® 7161
- Click/Winnewk
- Coke
- FTP Bounce
- Land
- Out Of Band
- NT inetinfo
- NT stop
- Oshare
- Ping
- Flooding
- Ping Of Death
- Plantage de site web
- Pong
- Smack/Bloop
- Snork
- Smurf
- SMTPd Overflow
- Sping
- Teardrop
- Trous de sécurité applicatifs
- UDP 0
- WinArp

- Wins 53
- Wins 137
- Cracking par dictionnaire
- Cracking par force brute
- Tempest
- Social Engeeriering
- Stéganographie
- Vers
- Virus
- Les mots de passe à usage unique (OTP : One Time Password)
- Les mots de passe
- Que faire lors d'une attaque ?
- Liste des ports utilisés par les troyens
- Les traces que vous laissez derrière vous...
- Open Source et sécurité

4.1.2 Eyrill aka Véronique Sainson

Signes particuliers : Chargée de recrutement en SSII. Notre doyenne :) C'est la seule fille de l'équipe

Articles publiés sur le site web :

- Mail Bombing
- La méthode FEROS
- Chevaux de Troie
- Espiogiciels (Spywares)
- Key Loggers
- Instrument de paiement électronique
- Projet de Loi sur la Société de l'Information
- Réglementation des télécommunications
- Protection des personnes physiques à l'égard du traitement des données
- A propos du Projet de Loi sur la Société de l'Information
- Vers qui vous orienter ?
- Directive 1999/93/CE
- Loi n° 88-19 du 5 janvier 1988
- Loi n° 78-17 du 6 janvier 1978
- Loi n° 2000-230 du 13 mars 2000
- Loi n° 2000-494 du 6 juin 2000 portant création d'une Commission nationale de déontologie de la sécurité
- Projet de Convention sur la cybercriminalité
- Projet de rapport explicatif du Projet de Convention sur la cybercriminalité Liberté de Communication
- Protection juridique des programmes d'ordinateur L'O.C.L.C.T.I.C.
- Décret de création de l'O.C.L.C.T.I.C.
- Directive concernant la protection juridique des bases de données
- Directive sur le commerce électronique
- La C.N.I.L.
- Catégorie de Moyens et de Prestations de Cryptologie pour lesquelles la procédure de déclaration préalable est substituée à celle d'autorisation
- Normes et Règlementations Techniques
- Catégorie de Déclaration des Moyens de Cryptologie
- Catégorie de Moyens et de Prestations de Cryptologie

4.1.3 SoGoodToBe

Signes particuliers : Cryptopathe
Articles publiés sur le site web :
- Introduction à la sécurité informatique
- BufferOverflow
- L'authentification .htaccess / .htpasswd
- Initiation à la cryptographie
- La cryptographie à algorithmes symétriques
- Le chiffrement AES
- Les fonctions de hachage
- 802.11b ou le WEP remis en cause
- La biométrie
- Deni de Service
- Déni de Service Distribué (DDoS)
- ARP redirect
- SSL
- Translation d'adresse (NAT)

4.1.4 Valgasu

Signes particuliers : Spécialiste en sécurité informatique
Articles publiés sur le site web :
- Cartes magnétiques
- DNS Spoofing
- IP Spoofing
- Cross Site Scripting

4.1.5 Jis aka Mustapha Benjada(*)

Signes particuliers : Ingénieur sécurité
Articles publiés sur le site web :
- Les PKI
- Les systèmes de détection d'intrusions gratuits
- Les firewalls
- Quelle détection d'intrusion adopter ?
- Classes de Fonctionnalité
- Le Compartemented Mode Workstation
- Le modèle "Bell La Padula"
- SSL

4.1.6 Secunix(*)

Signes particuliers : Chef de projet
Articles publiés sur le site web :
- Les sauvegardes

4.1.7 Mavric

Signes particuliers : Etudiant en informatique industrielle
Articles publiés sur le site web :
- PGP : Chiffrement de données

4.1.8 JB700(*)

Signes particuliers : Ingénieur sécurité
Articles publiés sur le site web :
- Les contrats de licence et la Sécurité Informatique
- Principes de sécurité informatique et évolution du marché

4.2 Nos actions

Securiteinfo.com prend part à des actions d'envergures internationales.

4.2.1 Le SETI

Le SETI est un projet qui consiste à décoder les signaux électromagnétiques en provenance de l'univers, à des fins de recherche d'une vie extraterrestre. Nos statistiques concernant ce projet se trouvent ici

4.2.2 Distributed.net

Distributed.net est une association de développement du calcul partagé. Grace à cette association, Securiteinfo.com participe au calcul du crackage de clé RC5 64 bits et au calcul mathématique des Règles de Golomb Optimales 24 et Règles de Golomb Optimales 25. Le principe de crackage de la clé RC5 64 bits est expliqué ici et sur le site officiel de distributed.net. Les règles de Golomb Optimales sont explicitées sur le site officiel de distributed.net. Gagnez un t-shirt Securiteinfo.com en participant à ce projet. Conditions et détails se trouvent sur cette page

4.2.3 La recherche contre le cancer

Intel, United Devices, la fondation nationale pour la recherche contre le cancer (américain) et l'université d'Oxford proposent un client de calcul distribué permettant de faire de la recherche dans le domaine de la chimie organique. Le but visé étant de calculer l'intéraction de molécules avec une protéine qui a une chance potentielle de combattre le cancer. Le client est disponible à cette adresse. Attention, le calcul est compliqué, il est donc conseillé d'avoir un ordinateur puissant (Pentium minimum !). Exemple : 1 heure pour réaliser 1% de calcul sur un Pentium MMX 200 MHz. Pour information, en Amérique, plus de 1500 personnes meurent par jour d'un cancer. N'hésitez pas à donner vos cycles CPU inutiles pour contrer cette terrible maladie. Les statistiques de notre équipe sont disponibles