SECURITE

Source : Cours suivi à la Haute école de gestion > Formation de concepteur en communication web.

Auteur : Anne-Christine Robert

Date : Décembre 2001

Intro

Machine sans connexion réseau

Risques :

- Panne (perte de données)
- Coupure de courant
- Accès physique à la machine par un tiers
  - Destruction de fichiers (voulu ou non)
  - Destruction de fichiers systèmes
- Virus via disquette, CD-Rom, zip, etc.
- Emplacement (humidité, variation de température, vase)
- Erreurs de manipulation (destruction, modifications de fichiers systèmes, etc.)
- Attention aux macros via disquette, etc.

Mesures :

- Emplacement approprié
- Sécuriser l'emplacement (interdire l'accès)
- Back-up
- Anti-virus (mises à jours des outils !!!)
- Réfléchir à ce que l'on fait (et ne pas aller trop vite, attention aux boutons par défaut)
- Attention aux installations
- Configuration (ex : détection des macros et avertissement)

Machine connectée a un réseau (interne)

Risques :

- Identiques que sur une machine non connectée +
- Destruction/modification de données
- Lecture de données confidentielles
- Virus qui se trouvent sur le réseau
- Observation (mouchard = programme appelé Sniffer)

Mesures :

- Configuration des droits d'accès
- Règles de sécurité données par le responsable sont à respecter.
- Login : choisir un bon mot de passe et ne jamais l'écrire quelque part.
  - Case sensitive, chiffres, caractères spéciaux, pas de mot existant
  - Prendre les premières lettres d'une phrase connue => évite le post it.

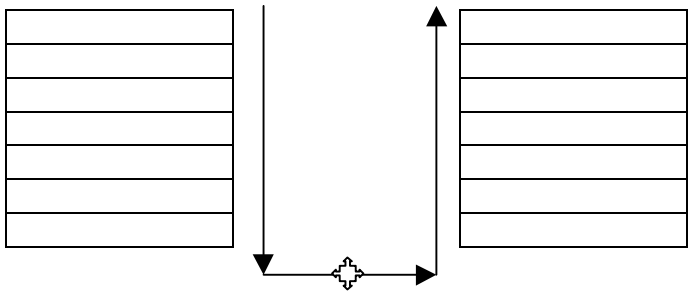
Machine connectée à Internet

Risques :

- Ce sont les mêmes, mais :
  - Le nb d'ennemis est multiplié
  - Pas de responsable du réseau qui puisse le protéger. Il n'y a pas de règles, ni de lois.
  - Les personnes sont difficilement identifiables.

Il faut très clairement identifier les activités "sûres" et celles qui ne le sont pas (ex : paiement à la poste via le net est relativement sûr).

Sniffer



Couches iso

Observation du réseau

Frame ethernet :
Destination | Source | Longueur | Données | PAD | Contrôle

A ce niveau (sur le câble), les données sont illisibles (sous forme de 0 et de 1), il faut un programme pour les lire.

Sur Solaris : Snoop : permet la lecture en claire de ce qui se passe sur le réseau.
Il y a des programmes qui détectent ce type de programmes espions. Si le programme est installé sur une machine, il est "facilement" décelable. Par contre, lorsqu'il est ajouté quelque part (via un portable par exemple) sur le réseau, il devient difficile à détecter.

HTTP

Côté client

A) Navigation (consultation de pages et navigation au moyen de liens)

Mécanisme :
url => requête (serveur DNS)

Théoriquement, le navigateur ne peut voir le poste qui l'héberge. Donc, à priori ne peut obtenir aucune information.
Excepté :

- Les cookies : enregistrement de données sur le disque, renvoi des informations lors des connexions ultérieures.
- L'historique : en fonction de la configuration, ces informations peuvent être lues par un utilisateur du réseau.
- Cache.

Au moment où l'on arrive sur un document de type Word, Excel, etc. en fait tout document que le navigateur ne peut interpréter, que se passe-t-il ?
Grâce à l'extension du fichier, le type-mime, le navigateur sait quel plug-in utiliser.
Type-mime est contenu dans l'en-tête de la réponse, donnée par le serveur.

Le navigateur appelle un programme (plug-in) pour interpréter le document.
Attention, à partir de ce moment, l'interprétation (image, document, programme, etc.) se fait avec les mêmes droits que l'utilisateur connecté.

Risques : programmes nuisibles, destructeurs, macros, virus, etc.

Solutions :

- Configuration du navigateur => ne pas permettre l'interprétation automatique de documents autres que du HTML.
- Choix (ouverture ou non du fichier) qui dépend de :
- Type de document (ex : Word on ne risque rien si les macros sont désactivées).
- Confiance que l'on a envers le site (Attention, même les sites sûrs peuvent être piratés)
- D'où on est => machine de test, Internet café, etc. (moins important que si l'on est chez soi, sur sa machine de travail).

B) Saisie de formulaire

Risques : dépendent des types de données saisies !
- Pas de risque si l'on saisit des banalités
- Risques élevés pour cartes de crédit, informations sensibles ou stratégiques. Observation du réseau et piratage des données, utilisation des informations, stockage des données. (1)
- Risques faibles à moyens pour les adresses postales et mail, téléphone. Spamming, réception de choses non-voulues. (2)

Solutions :
- (1) cryptage des données, utilisation de sites qui utilisent le protocole HTTP**S**.
- (2) créer plusieurs comptes (2^{e} adresse e-mail ou alias qu'on peut détruire si jamais il y a du spam).

C) Téléchargement de fichiers

=> voir risques liés à la disquette + interprétation des fichiers par le navigateur.

Côté serveur

D) Webmaster (côté serveur)

Risques :
- attaque sur le serveur pour obtenir un accès (genre de piratage) : le hacker peut ensuite changer les fichiers, etc.
  - faille d'un programme sur le réseau (ex : ftp, sendmail, application cgi [common gateway interface] => exécution d'un programme avant l'envoi de la réponse)
- Obtention d'un mot de passe
  - "amicalement"
  - par négligence du propriétaire (post-it, etc.)
  - par piratage (observation, écoute)
- Mauvaise configuration du serveur
- Faille du serveur web lui-même

Pourquoi pirater ?
=> par jeu
=> pour utiliser le serveur (on peut ensuite lui faire faire ce que l'on veut !)

Exemple : CODE RED

Utilise une faille des serveurs IIS.
Attaquait les serveurs web et se propageait de serveur en serveur. Exécute la fonction DIR sur C:\
Le processeur a une pile d'instruction qu'il exécute dans l'ordre.

Permet de prendre le contrôle du serveur avec les droits administrateurs (et changer les mots de passe !!!)

Implications financières

Institut de recherche "Computer economics" a calculé et trouvé les résultats suivants (chiffres à prendre avec le recul nécessaire !)

Nb de serveurs infectés : 1 mio
Nb de serveurs contrôlés : 8 mios
Prix :
Nettoyage des serveurs infectés : 1,1 mia $
Installation patch et vérification : 1,5 mia $

Que faire lorsque ce genre d'information circule ?

Update antivirus
Scan du serveur
Back up
Mais surtout, patcher les serveurs (bout de logiciel qui corrige un bug ou un trou de sécurité).

Solutions :
- Configuration
- Politique de sécurité (surveiller les activités du serveur, mise à jour des logiciels, engager une entreprise qui a pour mission de simuler des attaques afin de déceler un maximum des failles du système)

Attaque par déni de service

Risques
Le serveur partage ses ressources en fonction du nombre de demandes (il ne répond par aux requêtes chacune leur tour.)
Si trop de demandes simultanées : **time out**.
Avalanche de requêtes empêchant le serveur de répondre, parce que le protocole http partage les ressources de serveur entre tous les utilisateurs.

Solutions : firewall, politique de sécurité, surveillance. Limitation du nombre de connexions simultanées. Le chiffre maximum n'est pas une valeur calculable de manière mathématique. Il faut vérifier le nombre de requêtes quotidiennes et adapter la politique à la fréquentation du site. S'il y a beaucoup de requêtes, permettre beaucoup d'accès possibles simultanément. Il faut une RAM suffisante pour permettre à la machine de gérer les processus (gestion des connexions).

FTP (file transfert protocol)

Côté client

Webmaster

Occasion : mise à jour de site.
Risques :
- Ecraser un fichier par une mauvaise version
- Erreur dans la transmission qui corrompt le fichier (ex : utilisation du mode binaire pour un fichier qui nécessite le mode ASCII)

- Lors du login, écoute du password (transaction non sécurisée). Réutilisation sauvage du login.
- Lors du transfert, écoute et "vol" du fichier.
- Cf. http

Solutions :

- Sécuriser la connexion et le transfert (non prévus par le protocole au départ).
  - CuteFTP, WSFTP le proposent (SSL)

Utilisateur qui télécharge

Risques :

- Cf. http Téléchargement
- Destruction/modification de fichier

Côté serveur ftp

Utilisé pour mettre à jour un site web et pour proposer le téléchargement de fichiers (plus rapide et plus efficace)

Risques :

- Gestion des utilisateurs (ex : anciens collaborateurs conservent leurs logins et ceux-ci restent valables)
- Trous de sécurité
  - Configuration
  - Bugs du logiciel (ex : celui utilisé par code red)
- Attaques (cf. Serveur web) pour rendre le serveur indisponible
- Intrusion, piratage
- Remplissage de disque
- Abus d'espace

Solutions :

- Politique de sécurité (surveillance, réaction rapide, mis à jour des outils, se tenir informé des dangers, etc.)
- Mise en place d'un serveur ftp sécurisé (connexion et transmission des données sécurisés, même s'il y a écoute, le contenu n'est pas lisible).

Messagerie

SMTP (simple mail transfert protocol)

Protocole d'envoi des messages.

Côté client

Risques :

- Ecoute, vol du message au moment de la transmission.
- Perte de mail (envoi du mail, sans savoir que le destinataire n'a pas reçu le message).
- Mail arrive au mauvais destinataire
- Utilisation de "l'identité" d'une autre personne pour envoyer des mails.

Solutions :

- Utiliser un serveur SMTP sûr (compte gratuit est moins sûr qu'un serveur d'entreprise)
- Cryptage des informations confidentielles
- Pour les choses importantes, s'assurer que le destinataire a reçu le message (par téléphone ou demande de confirmation écrite).
- Cacher son adresse (mais la fonction reply ne fonctionne plus !!!)

Côté serveur

Risques :

- Utilisation du serveur comme relais pour faire du spamming. (Un programme utilise le serveur pour envoyer les mails => les utilisateurs spammés prennent le serveur relais comme la source de leurs problèmes. Et l'administrateur peut bloquer les requêtes provenant du serveur [domaine], même les utilisateurs "authentiques". Ceux-ci ne peuvent plus envoyer de mail aux domaines qui leur ont bloqué l'accès).
- Envoi de virus aux utilisateurs de la machine.

Solutions :

- Ne pas accepter l'utilisation du serveur SMTP par des utilisateurs externes au domaine.
  Il faut que l'émetteur OU le destinataire fasse partie du domaine pour que l'opération soit possible.

  Comment savoir qui est à l'intérieur ?
  1) Tester l'adresse IP de l'émetteur, ne permet pas la connexion si l'utilisateur est sur un autre login.
  2) Identification au moyen d'un login et mot de passe (nécessite une configuration sur le serveur et sur le client)

  Comment définir le destinataire ?
  1) Il suffit de tester son adresse électronique (rcpt to:)

- Utiliser MAPS (mail abuse prevention system)
  But : lister les serveurs SMTP par lesquels des personnes font du spam Ces listes devenant trop longues à gérer, un site les tient à jour.
  Il existe 3 listes principales :
  1) Liste des serveurs qui autorisent le relais
  2) Liste des serveurs qui spamment délibérément
  3) Liste des serveurs/providers qui n'identifient pas leurs utilisateurs
  NB: Il faut qu'un délit soit commis et prouvé pour qu'un serveur soit mis dans une liste.

- Surveillance de l'activité du serveur (ex : bloquer si trop de requêtes, limitation du nombre de destinataires, etc.)

- Analyse des messages non désirés. Si les messages ont un point commun, les filtrer.

MAPS : comment cela fonctionne-t-il ?
Solution A : gratuite
Le serveur SMTP fait une demande de connexion au serveur MAPS (DNS). Celui-ci vérifie si l'adresse demandée figure sur une liste et envoie la réponse. Si oui, on ne permet pas la connexion; si non, connexion et utilisation du serveur SMTP.

Solution B : payante
Installation d'une version locale des listes (sur un serveur dns). Elles seront donc accessibles sans délai (plus de dépendance au réseau), mais une mise à jour régulière s'impose.
Problème : SMTP devient dépendant de l'accès au serveur MAPS. Si le serveur ne répond pas, il y a time out (si celui-ci est paramétré trop long, il y a accumulation de demande de connexions en attente). Si le time out arrive au bout, soit :

- Acceptation de la connexion sans vérification via le dns

- Refus de la connexion sans vérification (les serveurs légitimes [qui en figurent pas dans une des listes] ne pourront plus envoyer de messages)

Définition : time out : limite temporelle affectée à une requête, tâche, etc. Permet de ne pas attendre indéfiniment.

POP / IMAP (post office protocol / Internet message Access protocol)

Côté client

Risques :

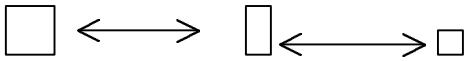
- Virus, programmes malveillants, fichiers attachés infectés, etc. (ne peuvent se trouver dans le message, car il s'agit que de texte)
- Imposture (faux message)
- Ne pas recevoir le message
- Piratage des messages reçus (sur le réseau ou directement sur le serveur)
- Vol du mot de passe (peut supprimer les messages et se faire passer pour le propriétaire de la messagerie)
- "Oublier" sa configuration ou ses messages sur une machine

Solutions :

- Configurer correctement le logiciel client (pas d'exécuter automatique)
- Jamais ouvrir d'attaché douteux
- Demander confirmation au destinataire
- Ne pas accuser trop tôt ! L'émetteur n'est pas une donnée fiable
- Crypter les messages importants
- Connexion sécurisée (mot de passe en passe pas en claire sur le réseau)

WEBMAIL - HOTMAIL

utiliser un compte mail via une interface http



Système interface machine

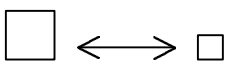
Risques :

- Voir risque http
- Enregistrement des données du compte : que fait l'entreprise avec nos données (si l'interface permet de lire n'importe quel compte).
- Enregistrement des messages : que fait le service avec les messages ?
- Spamming : l'entreprise connaît toutes les adresses mail qui sont lues.

Solutions :

- HTTP**S** : les données sont cryptées entre le navigateur et le serveur.
- Ne pas utiliser ce type de solution pour des informations confidentielles.

Utiliser un compte hotmail, caramail.



Pas de protocole POP

Système machine

Risques :

- Mêmes risques qu'en A)
- rappel : hotmail a été infecté par Code Red. Que se passe-t-il si l'entreprise arrête son activité ?

Solutions :

- HTTP**S** : les données sont cryptées entre le navigateur et le serveur.
- Ne pas utiliser ce type de messagerie pour des informations confidentielles.

LES VIRUS

Intro

Premier virus : "Core War", depuis les années 60. Trois informaticiens de la compagnie Bell. Il s'agissait d'un jeu où deux programmes se combattaient. Ils ont lâché les deux programmes et celui qui se reproduisait le plus et/ou éliminait l'autre avait gagné.
Objectif : création d'un programme qui pouvait occuper tout l'espace RAM. Les programmes étaient écrits en assembleur, passent directement par le CPU, sans passer par l'OS.
La technique de celui qui a gagné se base sur le bombardement de la RAM par des "1". Une fois terminé, le jeu n'est pas conservé. La deuxième étape a été l'enregistrement du virus sur le disque (en texte ou en binaire).

Peu à peu, le potentiel du danger leur est apparu set ils ont arrêté le développement de tels programmes. Ce n'était pas un virus en tant que tel, car il ne se propageait pas.

Le premier vrai virus a été créé par un doctorant, pour simuler la vie artificielle. Le but était de créer un programme parasite se greffant sur d'autres programmes (fichiers). L'idée fut vite reprise par des personnes mal intentionnées.

Quelques dates :

1960 :	Core War
1984 :	Article : "Le premier guide pour fabriquer un virus"
1986 :	Deux pakistanais inventent "brain", le premier virus qui se loge sur le secteur d'amorçage. Le Pakistan ne connaissait pas le droit d'auteur. Le virus s'est donc propagé dans le monde entier via des logiciels piratés. Première diffusion à grande échelle, mais qui n'avait rien à voir avec les "I love you" et "Code Red" actuels. Virus sympathique qui modifiait simplement le nom du lecteur de disquette de a:\ à brain:\
1986 :	VIRDEM (virus de démonstration). Se multiplie en se rattachant à d'autres fichiers. (sur .com : fichier exécutable). Présenté à une conférence du Chaos Computer Club [http://www.ccc.de]. L'idée fait succès, l'auteur écrit un livre => résultat : plein de virus apparaissent.
1987 :	Virus "Cascade" qui crypte son propre code. On ne peut pas le reconnaître (même en langage assembleur)
1988 :	Apparition de plus en plus de virus, souvent très destructeurs. Ex : Lehigh, on pouvait racheter un nouveau PC. Création de petites entreprises d'antivirus (pour chaque virus connu, création de l'antivirus = un programme par virus). A ce moment-là, personne ne s'y est intéressé. Lors de conférences, on remet même en doute l'existence des virus (Norton par exemple). IBM est contaminé par une variante de "Cascade" et contamine ses clients. La décision est prise de prendre les virus au sérieux. IBM mandate un labo de recherche ($$) antivirus et envoie les programmes de décontamination à ses clients (plus personne n'a douté de l'existence des virus depuis là !)
1989 :	"Datacrime", virus non-résidents, mais très agressif, car il formate le cylindre "O" (table d'allocation des fichiers : permet de retrouver les données qui sont enregistrées sur le disque par "morceaux").

1990 : ~18'000 virus différents (contre 18 en 1989).
2000 : "I love you", la rapidité de diffusion prend une nouvelle ampleur.

Description – définition d'un virus

Tout programme capable de se reproduire.
Il existe plusieurs types de virus et leur danger est associé aux facteurs suivants :

- Capacité de destruction (création de faille)
- Capacité de reproduction et diffusion
- Capacité de se cacher, se dissimuler
- Ancienneté / nouveauté
- Facilité de reproduction de variantes

Fonctionnement
Un virus est un programme qui est exécuté :

- Automatiquement lors du démarrage de l'ordinateur (il est alors associé à un endroit qui est toujours exécuté au départ)
- Lorsqu'un programme légitime est lancé alors qu'il était infecté
- Lorsque l'utilisateur fait une action (ouvrir un document par exemple)
- Si le virus est exécuté par l'utilisateur

Le virus ne peut se lancer lui-même.

Méthode de reproduction / diffusion

- En cherchant des cibles sur le disque, la RAM, le réseau
- Document attaché à un mail (attention aux faux virus !!)

Exemple
Un programme est un fichier exécutable.

Démarrage du programme
Code du programme

Exécution du programme [binaire]

Démarrage du virus
Code du programme
Démarrage du programme
Code du virus

Le démarrage du virus indique GoTo démarrage du programme. Ensuite, le code du programme est lancé et à sa suite le code du virus.

La taille du fichier augmente. Un virus ne réduit pas la taille du programme légitime en éliminant des lignes de code, car celui-ci ne fonctionnerait plus et l'utilisateur se rendrait compte immédiatement qu'il y a un problème. Le virus donne lui-même la taille du fichier en faisant croire à l'anti-virus qui celui-ci a sa taille habituelle.

Les types de virus

Les virus de secteur d'amorçage

Le secteur d'amorçage contient les informations nécessaires au démarrage de l'ordinateur. En général, c'est sur le premier secteur d'un disque. Une des informations est justement si le disque est bootable ou non (si on peut démarrer le système d'exploitation ou non).
!! Toujours avoir une disquette de démarrage au cas où !
Dans le BIOS, il y a l'indication des différents disques sur lesquels on exécute le démarrage de l'OS. Le BIOS est accessible même si l'OS ne peut démarrer.

> Rappel : Disques
> Un disque est un ensemble de pistes. On divise le disque en secteurs. Un cylindre est formé de plusieurs pistes de même diamètre et qui se trouvent sur plusieurs disques. Tout cela pour optimiser le temps de lecture et d'écriture.
>
> Il est plus efficace d'enregistrer des données sur différents disques, mais sur la piste et le secteur identique => une tête de lecture par disque.

Le virus se trouve sur le secteur d'amorçage, il est donc exécuter avant l'OS. S'il détruit simplement le secteur, l'OS ne pourra plus démarrer, le virus sera donc tout de suite détecté (la capacité à se cacher est nulle).
La stratégie du virus sera de déplacer la partie bootable sur un autre secteur et d'indiquer à l'ordinateur qu'il faut "maintenant" exécuter le code s'y trouvant afin de démarrer l'OS. Le virus a été exécuté entre temps, il est donc caché de l'anti-virus, celui-ci démarrant avec l'OS.

Les virus parasites

Appelés parasites car ils ne peuvent s'exécuter seuls. Ils se greffent dans un fichier "programme" existant. Ils seront actifs uniquement si l'ordinateur est allumé.

- Les virus résidants en mémoire
  Se trouvent dans un endroit privilégié de la RAM et infectent toutes les applications exécutées. Les cibles de ces virus sont les programmes utilisés.
- Les virus non-résidants
  Contiennent une procédure pour trouver les cibles.

Ces virus sont facilement détectables, car ils modifient la taille des fichiers hôte.
Attention aux virus furtifs, qui sont capables de "mentir" sur la taille réelle d'un fichier.

Les virus polymorphes

Un virus polymorphe modifie son aspect pour échapper aux anti-virus. Il faut se rappeler que ceux-ci utilisent les signatures des virus pour les reconnaître.

Les virus furtifs (intercepteur d'interruption)

Ils prennent le contrôle des interruptions, interceptant ainsi les appels au système. Cela leur permet de falsifier les informations retournées par le système (ex : donne la taille ou la date d'un fichier demandé). Se met entre le système réel et ce que l'utilisateur en voit. Ils sont très efficaces et difficiles à déceler.

Interruption
Sous-programme qui s'exécute chaque fois que l'utilisateur ou le matériel le demande.

La table d'interruption

N°	Adresse du code à exécuter
1	
2	
3	
4	
...	
255	

L'ordinateur est séquentiel, il exécute les instructions les unes après les autres. Si quelque chose d'important survient (ex : insertion CD-Rom) => interruption (ex : pour le signaler dans la fenêtre de l'explorateur).
Les 15 premiers n° sont réservés au matériel.

Peut-on interrompre une interruption ?
Oui, par ex. Crtl – Alt – Delete fonctionne à tout moment. C'est nécessaire. Implémenté avec un ordre de priorité dans le n° d'interruption. La table est prédéfinie par le système. Lorsqu'une interruption est interrompue, la première est mise de côté en attente de traitement pas le CPU (processeur).

Exemple de réallocation d'interruption par un virus

Un bon virus ne remplace pas simplement l'adresse d'un code à effectuer, sinon il sera automatiquement détecté (puisque la fonction qu'il remplace n'est pas exécutée).

Processus :

1. Le virus lit et stocke l'adresse qu'il va remplacer.
2. Il indique l'adresse du virus à la place de l'autre (sera donc exécuté en premier)
3. A la fin de son propre code, il va signaler au CPU l'adresse qu'il a remplacé comme étant l'adresse suivante à exécuter.

Dans une situation saine :
L'interruption n est appelée. Le CPU lit l'adresse de l'interruption et charge le code s'y trouvant et exécute pas à pas le code.

Dans une situation contaminée :
L'interruption n est appelée. Le CPU lit l'adresse du virus et charge le code s'y trouvant et l'exécute. La dernière instruction du code sera de charger le code de l'adresse de l'interruption n. La CPU continue d'exécuter pas à pas ce qui s'y trouve, l'interruption paraît donc être traitée normalement.

Remarque : c'est ainsi que travaillent les anti-virus. Ex : sauvegarder un fichier avant de l'ouvrir. Au moment de la sauvegarde, l'anti-virus vérifie la validité du fichier. Ensuite, l'ouverture sera sécurisée.

Les virus flibustiers (Bounty hunter)

Virus spécialisé dans l'attaque contre les anti-virus. Leur but est de les rendre inopérants.

Les virus macro

Infectent le fichier normal.dot, qui est le modèle par défaut pour les documents Word. Si un autre document est défini comme document par défaut, il infectera celui-ci !

Les vers

Virus se transmettant et infectant le réseau :

- Réseaux (Novell, Windows, etc.)
- Messagerie (ex : I love you)
- HTTP (ex : Code Red)

Nimda essaye tous les moyens !

Les virus VB

Programmés en VB et contenus dans la suite MSOffice. VBS interprété par IE. Beaucoup moins fins et s'appuient en général sur l'ignorance de l'utilisateur.

Les Troyens (cheval de Troie !! Exécutables !)

L'exécution d'un programme anodin, installe en cachette un programme nuisible ou une porte dérobée (back door). Permet de prendre le contrôle de la machine à distance !

Les faux virus (HOAX)

Exemple typique : vous recevez un message vous avertissant que IBM, Microsoft, etc. ont découvert un nouveau virus, en général très destructeur, que les anti-virus n'ont pas encore eu le temps de réagir, etc.
Avertissez toutes vos connaissances le plus vite possible. Travail sur la bêtise des gens.
Se propagent très bien, encore mieux que les vrais virus.
Il faut chercher des informations (permet de savoir si le virus est un vrai ou non !) sur les sites d'anti-virus, des grosses entreprises, etc.
Attention : ne jamais supprimer un fichier suite à ce genre de message ! Virus belge.

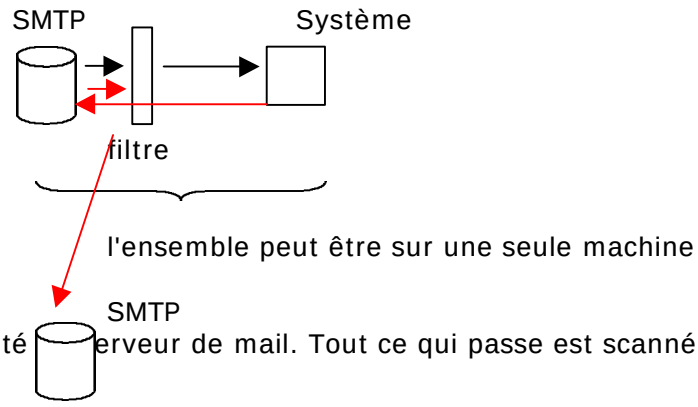
Création de virus : http://www.pipo.com/guillermito/darkweb/

EICAR : European Institute for Computer Anti-Virus

Création d'un "faux" virus qui permet de tester la validité de l'anti-virus (virus test). Tous les anti-virus le connaissent, donc cela permet de tester sa configuration sans risque.

Les anti-virus

Travaillent sur les signatures. Se mettent dans les interruptions (DOS par exemple). Permet d'être averti si un fichier est contaminé au moment de l'enregistrement. Scanner les disques, notamment les serveurs la nuit (pour ne pas gêner l'utilisation des machines).
Filtre pour la messagerie.



Filtrage

Fonctionnement :

1. Voir si message contient un fichier attaché.
2. Décodage (attaché en type MIME)
3. Analyse
4. Décision si infecté : QUI avertir ? L'émetteur ou le récepteur (paramétrable).

Exemple

15 juillet : mise en place de la dernière version d'un anti-virus
17 juillet : Virus SIRCAM. McAffee fait une nouvelle liste de virus
18 juillet : SIRCAM. McAffee annonce qu'il faut absolument ajouter l'extension du virus dans la liste des extensions à contrôler.
19 juillet : SIRCAM. McAffee annonce en grande urgence qu'il faut ajouter Trash dans les répertoires à parcourir.
Attention !! On se croit protégé, mais une mise à jour régulière des anti-virus n'est pas une garantie. Se tenir au courant sur les sites spécialisés.

Conclusion sur les virus

Nous avons vu différents types de virus et surtout qu'il était relativement simple de les créer ou du moins de les copier.
Bien que l'on soit dépendant des logiciels et systèmes d'exploitation, une politique (= comportement) conséquente limite efficacement les risques courus.
Les virus peuvent être néfastes, ils engendrent au mieux une perte de temps et au pire une perte de :

- Données
- Clients (site inaccessible, etc.)
- Matériels
- Argent

Ils concernent les webmasters au niveau de :

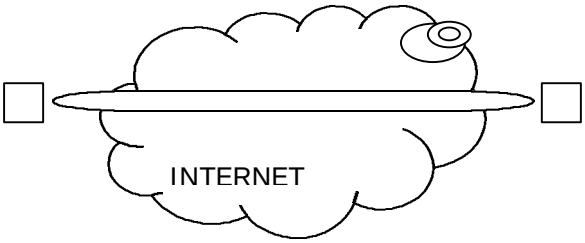
- Utilisation (PC, messagerie, programme, …)
- Responsable du contenu (validité des informations à disposition)
- Responsable serveur web (exigences aux ingénieurs système)
- Responsable serveur (patch, etc.)
- Personne de contact (le webmaster doit répondre à beaucoup de personnes concernant de l'information, quel que soit le type de question)

LA CRYPTOGRAPHIE

Nous allons voir quelques éléments de cryptographie, utiles au webmaster.

Introduction

La nature même du réseau IP (ou autre…), où les informations transitent en clair justifie à elle seule l'utilisation du cryptage. Notamment pour les données sensibles.



L'idée est de créer un canal virtuel, hermétique aux attaques. (HTTP**S** : ce canal existe et "protège" toutes les informations qui transitent).

Les attaques récentes sur les serveurs HTTP, telles qu'I love you, Nimda, etc., montrent que la transmission des informations n'est pas le seul risque auquel est exposé un serveur. Même si la transmission est sécurisée, le fait de stocker des informations en clair met aussi en péril la sécurité du site. De plus, on n'est pas à l'abri d'une personne maladroite ou malveillante travaillant dans l'entreprise.

Donc, nous allons voir les deux aspects :

1. Stockage des données
2. Transmission sécurisée

Généralités sur les algorithmes de cryptage à clés

Algorithme symétrique

- Clé identique pour crypter et décrypter
- Avantage : très rapide
- Désavantage : il faut envoyer/transmettre la clé.

Algorithme asymétrique

- Systèmes à clés publiques, clés privées.
  On utilise une des deux suivants ce que l'on veut faire.

Algorithme hybride

On crypte la clé symétrique avec la clé asymétrique. Utilise la rapidité des clés symétriques. On élimine le problème de la transmission de la clé symétrique.

A **B**

A envoie les informations cryptées avec la clé symétrique unique. La clé symétrique est cryptée avec la clé publique de B. Les deux informations peuvent se trouver dans le même message.
B décrypte la clé symétrique avec sa clé privée. Ensuite, il peut décrypter le message, au moyen de la clé symétrique.

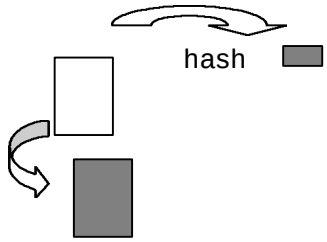
Le hashage

Les fonctions de hashage sont utilisées dans plusieurs contextes. La cryptographie en est un, la programmation un autre.
Caractéristiques : le paramètre (input) est de longueur non définie. Le résultat (output) est lui de taille fixe. Facilement calculable. Pas de fonction inverse possible. Pas de collision.

Exemple :

- La famille "MD" (MD5, MD9). Entreprise RSA.
- SHA-1 (Secure hash algorithme). Définit dans le cadre d'un projet du gouvernement américain.
- RIPE-160. Successeur des "MD". (160 car la clé est codée sur 160 bits)

Utilisation



A crypte le message avec la clé publique de B. B décrypte le message, obtenant le message ainsi que le hash. Il crée un hash avec le message et compare les deux hashs. Si ceux-ci sont identiques, le message reçu est identique au message envoyé.

A et B ont le même algorithme de hashage. Se trouve généralement dans le logiciel (ex. MD5). Permet de s'assurer que le message est intact.

Autre utilisation

La fonction de hashage donne un index unique pour chaque élément.

N°	...
1	
2	
3	
4	
5	
6	
n	

Hash(...) = x, chiffre unique
Du coup (...) est positionné au niveau x de la table. Quand je cherche la position de (...), je fais hash(...) et j'obtiens sa position.

Algorithmes pour crypter et décrypter

Code de César

Décalage des lettres, chiffres utilisés. La structure du message reste identique (fréquence d'apparition des lettres !). "Facile" à trouver. Plusieurs versions jusqu'à celle d'Auguste, qui elle, est indécryptable. Il suffit d'utiliser une clé aussi longue que le message.

```
      Messageacrypter..
      eticilacleeticila...
+     message crypté
```

Les problèmes sont le temps de traitement, la longueur de la clé, la transmission de la clé, il faut en trouver chaque fois une autre.

Clés symétriques

A) DES : Data encryption standard (IBM ~1976)
   Traitement de l'information
   1. Découpage en bloc de 64 bits
   2. Permutation des blocs
   3. Découpage des blocs en deux parties
   4. Permutations et substitutions répétées dans les deux blocs
   5. Groupement des deux parties

   Les blocs font 64 bits
   8 bits : réservés pour le check
   56 bits : pour la clé
   Aujourd'hui, cette méthode est trop facile à décrypter. On utilise actuellement de 3DES (logique identique, mais la procédure est appliquée 3 fois).

B) AES : advanced encryption standard (successeur de DES)
   Bloc de 128 bits, clé codée sur 128, 192 et 256 bits.
   Concours de la meilleure clé :
   1. Rijndael (belges)
   2. Mars (IBM)
   3. RC6 (RSA)
   4. Serpent
   5. TwoFish

Clés publiques

RSA (1978)
Idée de base : la multiplication de deux nombres premiers est rapide, par contre, la factorisation en deux nombres premiers d'un grand nombre est très longue.
Factorisation : recherche des nombres qui divisent un autre nombre.
Ex : 16 {1;2;4;8;18}

RSA travaille avec des nombres premiers très grands (100 digits).
La diffusion de leur méthode favorise la confiance en l'algorithme.

M : message en claire
C : message crypté
(e, n) : clé publique
(d, n) : clé privée

Cryptage : C = m^e mod n
Décryptage : M= c^d mod n

Il faut choisir {d,e,n}

Soit p,q, deux nombres entiers de taille égale.

1) n = pq
2) chercher e, tel qu'aucun facteur commun avec (p-1)(q-1)
3) calculer d, tel que ed mod(p-1)(q-1) = 1

Prenons p=29, q=37

1. n = 1073
2. (p-1)(q-1) = 28 x 36 = 1008
   e = 71
3. d tel que 71d mod 1008 = 1
   d = 1079

Clé publique : (71,1073)
Clé privée : (1079,1073)

Exemple :

M = 'hello'
Il faut transformer les lettres en chiffres pour pouvoir effectuer des calculs (en code ASCII).
M devient : 72'69'76'76'79

Il faut ensuite diviser le message M en blocs plus petits que le nombre de digits de n (sinon, possibilité de trouver des répétitions)

> 726'976'767'900 [on complète le dernier bloc avec des 0].

Note : [en ASP]

Ord("A") = code ASCII de "A"
Asc(76) = lettre correspondant au code ASCII

Effectue cela caractère après caractère.

Mid("string", start, longueur) : permet de découper un string en caractères.

M^e mod n

726 => 436
976 => 822
767 => 825
900 => 552

C = 436822825552

M = C^d mod n

436 => 726
822 => 976
825 => 767
552 => 900

On connaît le nombre de digit des blocs : n-1

Stéganographie – Watermarking

L'idée est de cacher dans un support "anodin" des informations secrètes. Utilisé pour transmettre des informations ou pour signer (watermarking) un document multimédia, afin de prouver que l'on en est l'auteur.
Impossible de retrouver l'information si on ne sait pas comment la chercher.

Procédure :
1. Interprétation des fréquences
2. Transformation mathématique
3. Ajouter l'information cachée à la transformée
4. Faire la transformation inverse
5. On obtient ainsi l'image, le son, … contenant l'information de manière invisible.

Le logiciel utilisé doit être le même pour l'insertion et l'extraction de l'information cachée. Le fait que l'information principale soit banale est une protection supplémentaire (ne montre pas le fait que le vrai message est caché).

Utilisation pratique de la cryptographie pour webmaster :

1. SSL : secure socket layer (serveur HTTPS)
2. Développement d'applications liées à une BDD. Ne jamais stocker des mots de passe en claire. Penser à trouver une solution simple pour crypter le mot de passe.