NOTES DU COURS DE SÉCURITÉ INFORMATIQUE ET CRYPTOGRAPHIE.

Djamba Tunda-Olembe

REPUBLIQUE DEMOCRATIQUE DU CONGO

ENSEIGNEMENT SUPERIEUR ET UNIVERSITAIRE

UNIVERSITE MARIA MALKIA

COURS DE SECURITE INFORMATIQUE ET CRYPTOGRAPHIE

BAC 3 INFORMATIQUE

PAR MASTER ING TUNDA-OLEMBE DJAMBA

NOVEMBRE 2024

PRESENTATION DU COURS

0.1 Liminaire

La sécurité Informatique est l'ensemble des moyens misent en œuvre pour réduire les vulnérabilités d'un système information contre les menaces éventuelles auxquelles il peut être confronté. Les principales solutions de sécurité se basent sur la mise en œuvre des procédures de contrôle d'accès, des surveillances et des techniques de chiffrement et d'isolation de l'environnement.

Avec le développement de l'utilisation d'internet, de plus en plus d'entreprises ouvrent leur système d'information à leurs partenaires ou leurs fournisseurs, il est donc essentiel de connaître les ressources de l'entreprise à protéger et de maîtriser le contrôle d'accès et les droits des utilisateurs du système d'information. Il en va de même lors de l'ouverture de l'accès de l'entreprise sur internet.

Par ailleurs, avec le nomadisme, consistant à permettre aux personnels de se connecter au système d'information à partir de n'importe quel endroit, les personnels sont amenés à « transporter » une partie du système d'information hors de l'infrastructure sécurisé de l'entreprise.

Avec le développement de l'utilisation d'internet, de plus en plus d'entreprises ouvrent leur système d'information à leurs partenaires ou leurs fournisseurs, il est donc essentiel de connaître les ressources de l'entreprise à protéger et de maîtriser le contrôle d'accès et les droits des utilisateurs du système d'information. Il en va de même lors de l'ouverture de l'accès de l'entreprise sur internet.

Par ailleurs, avec le nomadisme, consistant à permettre aux personnels de se connecter au système d'information à partir de n'importe quel endroit, les personnels sont amenés à « transporter » une partie du système d'information hors de l'infrastructure sécurisé de l'entreprise. Hors que toute ordinateur connecté à un réseau ou l'Internet est potentiellement vulnérable à une attaque informatique. Ces attaques sont la plupart lancés automatiquement par des machines infectées par des programmes malveillants (virus, vers, cheval de Troie, Rootskit spams,..), à l'insu. Par conséquent, il faut mettre en œuvre les mesures de sécurité pour protéger le système d'information de l'entreprise.

La sécurité du système d'information d'une entreprise est un **requis** pour la poursuite de ses activités.

La Cryptographie est l'étude des méthodes donnant la possibilité d'envoyer des données de manière confidentielle sur un support donné. Les fonctions principales de la Cryptographie sont le Chiffrement, le Déchiffrement et la Clé.

0.2. Prérequis

1. Cours d'architecture des ordinateurs
2. Cours d' Algorithmique et programmation
3. Cours des réseaux informatiques

0.3. Les objectifs des compétences

Ce cours a les objectifs des compétences suivants :

1. Familiariser les étudiants avec la terminologie et les concepts de la sécurité informatique ;
2. Donner aux étudiants un concentré d'informations sur tout ce qui concerne la sécurité informatique ;
3. Sensibiliser les étudiants risques liés aux attaques informatiques et
4. Amener les étudiants à prendre conscience de l'importance de la sécurité informatique pour la poursuite des activités d'une entreprise informatisée.

0.4. Les objectifs d'apprentissage

A la fin de ce cours l'étudiant ou l'étudiante doit être capable de :

1. Dégager une compréhension globale et cohérente du domaine de la sécurité informatique être au fait des enjeux, des problématiques et des solutions proposés dans la littérature technique ;
2. Elaborer des modèles d' organisation, des procédures, et de bonnes pratiques permettant de sécuriser un système dé information ;
3. Elaborer des procédures pour s'assurer que les mesures de sécurité sont efficaces et que les usagers restent conformes aux procédures et aux bonnes pratiques et

4. Mettre en œuvre la politique de sécurité de l'entreprise, les mécanismes de protection du système d'information de l'entreprise et les outils de l'audit de sécurité informatique.

0.5. Les savoirs sous-jacents

Chapitre 1 : Les fondements de la sécurité informatique

1.1. Généralités sur la sécurité informatique, vulnérabilités informatiques et risques

1.2. Les attaques informatiques

1.3. Les programmes malveillants (Malwares)

1.4. Les logiciels de sécurité informatique (les logiciels anti-programmes malveillants)

1.5. Le pare-feu

1.6. Les protocoles de sécurité

Chapitre 2 : Les ports logiques, les vulnérabilités informatiques et les journaux d'évènements Logs

2.1 Les ports logiques

2.2 Les vulnérabilités informatiques

2.3 Les journaux d'évènements logs

Chapitre 3 : Introduction aux outils de sécurité informatique

3.1 Les scanneurs de ports et l'outil de sécurité informatique Nmap

3.2 Les scanneurs des vulnérabilités informatiques et l'outil de sécurité Informatique NESSUS

3.3 Les tests d'intrusions et l'outil de sécurité informatique SNORT

3.4 La surveillance et l'analyse des journaux logs et les

outils de sécurité Informatique SYSLOG-ng et AWAStats

Chapitre 4 : Les normes et méthodes de sécurité informatiques

4.1 Introduction

4.2 Les normes de sécurité informatiques

4.3 Les principales méthodes de sécurité informatique

Chapitre 5 : Notions de Cryptographie

5.1 Introduction à la Cryptographie

5.2 Cryptographie classique

5.3 Le chiffrement par clé publique

5.4 La gestion de clé

0.6 Bibliographie

A. Livres

1. PILLOU J.F. : « Tout sur la sécurité informatique », Editions DUNOD, Paris, 2016
2. BLOCH L. et WOLFHUGEL C. : « Sécurité informatiques : principes et Méthodes », Editions EYROLLES, Paris, 2023

B. Autres sources

1. DUMAN R. : « Cours de Cryptographie et sécurité informatique », [En ligne], 2010, https : www.fr.Slieshare. [Consulté le01/ 12/ 2024]
3. TUNDA-OLEMBE D. : « Cours de sécurité informatique niveau master » [En ligne], 2018, https : www.fr.scribd.com. [Consulté 01/12/ 2024]

0.7 Méthode de communication

1. Cours théorique orienté vers la pratique ;

2. Travaux et discussion en groupe ;
3. Organisation TD et TP ;
   Autoformation

0.8 Méthode d'évaluation

1. Travaux pratiques ;
2. Interrogations ;
3. Examen

CHAPITRE 1. LES FONDEMENTS DE LA SECURITE INFORMATIQUE

I.1 GENERALITES ,PRINCIPE ET VULNERABILITES

I.1.1 Generalites

Avec le développement de l'utilisation d'internet, de plus en plus d'entreprises ouvrent leur système d'information à leurs partenaires ou leurs fournisseurs, il est donc essentiel de connaître les ressources de l'entreprise à protéger et de maîtriser le contrôle d'accès et les droits des utilisateurs du système d'information. Il en va de même lors de l'ouverture de l'accès de l'entreprise sur internet.

Par ailleurs, avec le nomadisme, consistant à permettre aux personnels de se connecter au système d'information à partir de n'importe quel endroit, les personnels sont amenés à « transporter » une partie du système d'information hors de l'infrastructure sécurisé de l'entreprise.

Avec le développement de l'utilisation d'internet, de plus en plus d'entreprises ouvrent leur système d'information à leurs partenaires ou leurs fournisseurs, il est donc essentiel de connaître les ressources de l'entreprise à protéger et de maîtriser le contrôle d'accès et les droits des utilisateurs du système d'information. Il en va de même lors de l'ouverture de l'accès de l'entreprise sur internet.

Par ailleurs, avec le nomadisme, consistant à permettre aux personnels de se connecter au système d'information à partir de n'importe quel endroit, les personnels sont amenés à « transporter » une partie du système d'information hors de l'infrastructure sécurisé de l'entreprise.

La sécurité Informatique est l'ensemble des moyens misent en œuvre pour réduire les vulnérabilités d'un système information contre les menaces éventuelles auxquelles il peut être confronté. Les principales solutions de sécurité se basent sur la mise en œuvre des procédures de contrôle d'accès, des surveillances et des techniques de chiffrement et d'isolation de l'environnement.

Un risque permet de mesurer les possibilités de l'occurrence d'un événement associé à une situation ou une action. Dans le cas de la sécurité informatique en entreprise, il s'agit de ce que l'on peut perdre en l'absence des moyens adéquat de sécurisation. Lorsqu'on évoque les risques logiques tels qu'intrusion, usurpation d'identité, malveillance, vol d'information, modification d'information..., et les risques physiques tels que les défaillances matérielles, les dommages électriques, les actes de vandalisme..., etc.

Une politique de sécurité peut être vue comme l'ensemble des modèles d'organisation, des procédures et des bonnes pratiques techniques permettant d'assurer la sécurité du système d'information.
Mais qu'est-ce que la sécurité d'un SI ? Elle tourne autour des 6 principaux concepts suivants : l'intégrité des données, la confidentialité de l'information et des échanges, la disponibilité

des services, l'authentification des utilisateurs, la non répudiation des transactions et le respect de la vie privé.
Pour garantir la sécurité, une politique de sécurité est généralement organisée autour de 3 axes majeurs : la sécurité physique des installations, la sécurité logique du système d'information et la sensibilisation des utilisateurs aux contraintes de sécurité.

Un audit de sécurité permet de mettre en évidence les faiblesses de la mise en œuvre d'une politique de sécurité. Le problème peut venir de la politique elle-même : mal conçue ou inadaptée aux besoins de l'entreprise, ou bien d'erreurs quant à sa mise en application.
Des audits sont nécessaires : suite à la mise en place initiale d'une politique de sécurité, puis régulièrement pour s'assurer que les mesures de sécurité sont mises à niveau et que les usages restent conformes aux procédures.

I.1.2 Principe de la sécurité informatique

Le système d'information est généralement défini par l'ensemble des données et des ressources matérielles et logicielles de l'entreprise permettant de les stocker ou de les faire circuler. Le système d'information représente un patrimoine essentiel de l'entreprise, qu'il convient de protéger.

La sécurité informatique, d'une manière générale, consiste à assurer que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre prévu.

La sécurité informatique vise généralement cinq principaux objectifs :

- **L'intégrité de l'information** : c'est-à-dire garantir que seules les personnes autorisées qui peuvent modifier ou détruire l'information ;
- **La confidentialité de l'information et des échanges :** elle consiste à assurer que seules les personnes autorisées aient accès à l'information et aux échanges ;
- **La disponibilité des services et des ressources** : elle consiste à garantir l'accès aux services et aux ressources à des personnes autorisées;
- **La non répudiation des transactions** : elle consiste à garantir que l'accès à l'information et aux ressources ne doit pas être refusé à des personnes autorisées ;
- **L'authentification** : elle consiste à assure que seules les personnes autorisées aient accès aux ressources et à l'information.
- **Le respect de la vie privé** : elle consiste à exiger l' autorisation de l' utilisateur avant d' accéder à ses données privés.

La confidentialité

La **confidentialité** consiste à rendre l'information inintelligible à d'autres personnes que les seuls acteurs de la transaction.

L'intégrité

Vérifier l'intégrité des données consiste à déterminer si les données n'ont pas été altérées durant la communication (de manière fortuite ou intentionnelle).

La disponibilité

L'objectif de la disponibilité est de garantir l'accès à un service ou à des ressources.

La non-répudiation

La **non-répudiation** de l'information est la garantie qu'aucun des correspondants ne pourra nier la transaction.

L'authentification

L'authentification consiste à assurer l'identité d'un utilisateur, c'est-à-dire de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être. Un contrôle d'accès peut permettre (par exemple par le moyen d'un mot de passe qui devra être crypté) l'accès à des ressources uniquement aux personnes autorisées.

I.1.3 VULNERABILITE(FAILLE) ET RISQUES

Vulnérabilité : erreur de conception (bug) dans un produit pouvant altérer la sécurité du système

Où peut-on trouver des vulnérabilités ?

- au niveau du système d'exploitation
- au niveau applicatif
- au niveau du réseau

CAUSES DE VULNERABILITES ET SOLUTION

- Principales causes :
  - les données transitent d'une machine à une autre dans un milieu non sécurisé/sécurisable (internet).
  - failles applicatives de certaines piles de protocoles TCP/IP.
  - mécanisme d'authentification insuffisant (ex : rsh, rlogin).

- Principales solutions mises en oeuvre :
  - utiliser des méthodes de chiffrement.
  - corriger les failles applicatives de piles des protocoles.
  - filtrer les accès aux différents services.

Un risque permet de mesurer les possibilités de l'occurrence d'un événement, associé à une situation ou une activité. De l'autre côté, un enjeu est grossièrement ce que l'on peut gagner ou perdre en posant un acte. Dans le cas de la sécurité informatique en entreprise, il s'agit plutôt de ce que l'on peut perdre, en l'absence de moyens adéquat de sécurisation.[1] Lorsque l'on évoque les risques susceptibles d'engendrer un incident informatique sur le Système d'Information d'une entreprise, on distingue deux grandes catégories :

- Les risques physiques;
- Les risques logiques.

[1]**AMAN VLADIMIR, Concevoir la Sécurité Informatique en Entreprise, p23.**

a) Les risques physiques

Il s'agit de toutes les atteintes physiques directes dont peut être victime un système d'informations au cours de son cycle de vie. On les appelle également risques matériels, parce qu'ils ont trait à l'intégrité du matériel.

Ces risques physiques peuvent être d'origine accidentelle ou malveillante et les conséquences sont aisément identifiables. Ces incidents détériorent les ressources matérielles du système d'information et peuvent avoir un impact direct sur les actifs informationnels que contiennent les systèmes informatiques.

Les risques physiques constituent dans la conception traditionnelle de la sécurité, les premières sources d'inquiétude des responsables d'entreprise en termes de sécurité, même si en pratique, ils ne représentent qu'un faible pourcentage des sinistres informatiques enregistrés en entreprise.

b) Les risques logiques

Avec le développement fulgurant de l'informatique distribuée par opposition à l'informatique centralisée, les données et les applications ont acquis une importance plus grande. En effet, l'on assiste à une migration progressive de la valeur du matériel vers la valeur des données et des applications. La question principale est de savoir comment évaluer et analyser la valeur de ce qui n'est pas physique, donc immatériel c'est-à-dire **la donnée**.

c) Quelques considérations sur les risques

C'est pour répondre à cette question que sont apparues les notions d'accident, d'erreur et de malveillance, directement issues des méthodes d'analyse des risques développées ces dernières années.

1) **L'accident** : Il s'agit là d'un événement perturbant les données ou les flux de données, en l'absence de dommages physiques aux équipements (altération physique du matériel).
2) **L'erreur** : Il peut s'agir d'une erreur de conception, de programmation, de paramétrage ou de manipulation des données et de leurs supports. L'erreur désigne des préjudices consécutifs à l'intervention humaine dans le processus de traitement automatisé des données. Elles constituent les risques les plus fréquents dans le cycle de vie d'un système d'information en entreprise.
3) **La malveillance** : Il s'agit de tous actes traduisant la volonté manifeste de son auteur de faire usage, sans autorisation d'un système d'information, avec des intentions préjudiciables. Le virus informatique et l'acte de malveillance le plus médiatisé, quoiqu'il en existe une très grande diversité (Chevaux de Troie, etc.).
4) **Vol d'informations** : Avec Internet, le vol s'en trouve dématérialisé, étant donné que l'objet dérobé est une valeur virtuelle, voire abstraite. Il s'agit pour l'entreprise de risques tels que : espionnage industriel, vol des listes des clients, vol d'informations comportant des données

personnelles de clients ou du personnel, vol des plans ou les recettes des productions, vol de la comptabilité.

5) **Usurpation d'identité** : Utilisation du compte d'un client ou d'un partenaire, utilisation des identifiants d'une personne de l'entreprise à des fins malveillantes, etc. Une usurpation d'identité peut avoir des conséquences énormes sur l'image de marque d'une entreprise.

I.2 ATTAQUES INFORMATIQUES ET PROTECTION

I.2.1 ATTAQUES INFORMATIQUES

Tout ordinateur connecté à un réseau est potentiellement vulnerable aux attaques.

La sécurité de l'information traite de la prévention de la fraude, ou, à défaut, de sa détection dans des systèmes d'information à l'intérieur desquels l'information elle-même n'a pas d'existence physique significative. On verra dans les transparents suivants une liste d'exemples évidents de tricherie, qui se sont produits dans des cas réels. Ce sont des exemples d'attaques spécifiques qu'une organisation ou un individu peut avoir à affronter. La nature de l'attaque varie considérablement selon les circonstances. Heureusement, il est possible d'approcher le problème en examinant les types génériques d'attaques pouvant être rencontrées. Ce sera le sujet de la prochaine section.

- Obtenir un accès non autorisé à l'information (c'est-à-dire, violer secret ou confidentialité) ;
- Usurper l'identité d'un autre utilisateur pour modifier ses attributs de responsabilité ou pour utiliser les droits de ce dernier dans le but de
  - diffuser une information frauduleuse ;
  - modifier une information légitime ;
  - utiliser une identité frauduleuse pour obtenir un accès non autorisé ;
  - faciliter des transactions frauduleuses ou en tirer partie.
- Refuser la responsabilité d'une information que le fraudeur a diffusée ;
- Prétendre avoir reçu de la part d'un autre utilisateur une information en fait créée par le fraudeur (par exemple, de fausses attributions de responsabilité ou de confiance).

- Prétendre avoir envoyé (à un moment donné) une information qui soit n'a pas été envoyée, soit l'a été à un autre moment ;
- Nier avoir reçu une information ou prétendre qu'elle a été reçue à un autre moment ;
- Étendre des droits d'un fraudeur (pour un accès à des informations) ;
- Modifier (sans autorisation) les droits d'autrui (les inscrire, restreindre ou élargir leurs droits, etc.) ;
- Dissimuler la présence d'information (la communication cachée) dans une autre information (la communication déclarée) ;
- S'insérer dans un lien de communication entre d'autres utilisateurs en tant que point de relai actif (et indétecté) ;
- Apprendre qui a accès à une information donnée (fichiers, etc.) et quand les accès sont réalisés, même si l'information elle-même reste cachée (par exemple, la généralisation de l'analyse de traffic de canaux de communication à des bases de données, des logiciels, etc.) ;

- Mettre en cause l'intégrité d'un protocole en révélant une information que le fraudeur est censé (selon les termes du protocole) garder secrète ;
- Pervertir la fonction d'un logiciel, en général par l'ajout d'une fonction cachée ;
- Faire qu'autrui viole un protocole en intoduisant une informatio incorrecte ;
- Saper la confiance en un protocole en causant des défaillances visibles dans le système ;
- Empêcher la communication entre d'autres utilisateurs, en particulier par des interférences afin que la communication authentique soit rejetée comme non authentique.

Les menaces se multipliant ces derniers temps (virus, vers, spyware, intrusions), il est primordial de savoir quelles attitudes et actions adopter pour se prémunir et réagir aux problèmes de sécurité. A côté de ça un autre problème (le spam) augmente aussi de manière inquiétante...

Une **menace** (*threat*) représente une action susceptible de nuire, tandis qu'une **vulnérabilité** (*vulnerability*, appelée parfois faille ou brèche) représente le niveau d'exposition face à la menace dans un contexte particulier. La **contre-mesure** (ou parade), elle, représente l'ensemble des actions mises en œuvre en prévention de la menace.

Les contre-mesures ne sont généralement pas uniquement des solutions techniques mais également des mesures de formation et de sensibilisation à l'attention des utilisateurs, ainsi qu'un ensemble de règles clairement définies.

Les contre-mesures ne sont généralement pas uniquement des solutions techniques mais également des mesures de formation et de sensibilisation à l'attention des utilisateurs, ainsi qu'un ensemble de règles clairement définies.

Afin de pouvoir sécuriser un système, il est nécessaire d'identifier les menaces potentielles, et donc de connaître et de prévoir la façon de procéder de l' « ennemi ». Le but de cet ouvrage est ainsi de donner un aperçu des menaces, des motivations éventuelles des pirates, de leur façon de procéder, afin de mieux comprendre comment il est possible de limiter les risques.

Une **attaque** est l'exploitation d'une faille d'un système informatique (système d'exploitation, logiciel ou bien même de l'utilisateur) à des fins non connues par l'exploitant du système et généralement préjudiciables.

Sur Internet des attaques ont lieu en permanence, à raison de plusieurs attaques par minute sur chaque machine connectée. Ces attaques sont pour la plupart lancées automatiquement à partir de machines infectées (par des virus, chevaux de Troie, vers, etc.), à l'insu de leur propriétaire. Plus rarement il s'agit de l'action de **pirates informatiques**.

Afin de contrer ces attaques, il est indispensable de connaître les principaux types d'attaques afin de mieux s'y préparer.

Les motivations des attaques sont de différentes sortes :

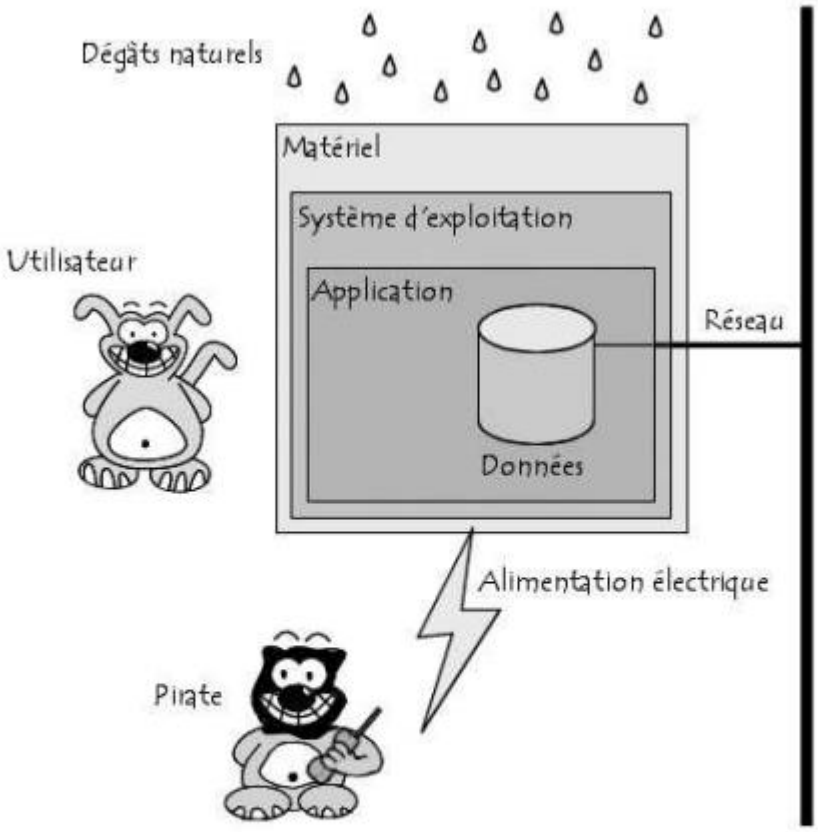
- obtenir un accès au système ;
- voler des informations, tels que des secrets industriels ou des propriétés intellectuelles ;
- glaner des informations personnelles sur un utilisateur ;
- récupérer des données bancaires ;
- s'informer sur l'organisation (entreprise de l'utilisateur, etc.) ;
- troubler le bon fonctionnement d'un service ;
- utiliser le système de l'utilisateur comme « rebond » pour une attaque ;
- utiliser les ressources du système de l'utilisateur, notamment lorsque le réseau sur lequel il est situé possède une bande passante élevée.

La sécurité de l'information traite de la prévention de la fraude, ou, à défaut, de sa détection dans des systèmes d'information à l'intérieur desquels l'information elle-même n'a pas d'existence physique significative. On verra dans les transparents suivants une liste d'exemples évidents de tricherie, qui se sont produits dans des cas réels. Ce sont des exemples d'attaques spécifiques qu'une organisation ou un individu peut avoir à affronter. La nature de l'attaque varie considérablement selon les circonstances. Heureusement, il est possible d'approcher le problème en examinant les types génériques d'attaques pouvant être rencontrées. Ce sera le sujet de la prochaine section.

❑ Types d'attaques

Les systèmes informatiques mettent en œuvre différentes composantes, allant de l'électricité pour alimenter les machines au logiciel exécuté via le système d'exploitation et utilisant le réseau.

Les **attaques** peuvent intervenir à chaque maillon de cette chaîne, pour peu qu'il existe une vulnérabilité exploitable. Le schéma suivant rappelle très sommairement les différents niveaux pour lesquels un risque en matière de sécurité existe.



Il est ainsi possible de catégoriser les risques de la manière suivante :

- **Accès physique** : il s'agit d'un cas où l'attaquant à accès aux locaux, éventuellement même aux machines :
  - coupure de l'électricité ;
  - extinction manuelle de l'ordinateur ;
  - vandalisme ;
  - ouverture du boîtier de l'ordinateur et vol de disque dur ;
  - écoute du trafic sur le réseau ;
  - ajout d'éléments (clé USB, point d'accès WiFi…).

- **Interception de communications** :
  - vol de session (*session hijacking*) ;
  - usurpation d'identité ;
  - détournement ou altération de messages.

- **Dénis de service** : il s'agit d'attaques visant à perturber le bon fonctionnement d'un service. On distingue habituellement les types de déni de service suivant :
  - exploitation de faiblesses des protocoles TCP/IP ;
  - exploitation de vulnérabilité des logiciels serveurs.

- **Intrusions** :
  - balayage de ports ;
  - élévation de privilèges : ce type d'attaque consiste à exploiter une vulnérabilité d'une application en envoyant une requête spécifique, non prévue par son concepteur, ayant pour effet un comportement anormal conduisant parfois à un accès au système avec les droits de l'application. Les attaques par **débordement de tampon** (*buffer overflow*) utilisent ce principe.
  - Maliciels (virus, vers et chevaux de Troie).

- **Ingénierie sociale** : dans la majeure partie des cas le maillon faible est l'utilisateur lui-même ! En effet, c'est souvent lui qui, par méconnaissance ou par duperie, va ouvrir une brèche dans le système, en donnant des informations (mot de passe par exemple) au pirate informatique ou en exécutant une pièce jointe. Ainsi, aucun dispositif de protection ne peut protéger l'utilisateur contre les arnaques, seuls le bon sens, la raison et un peu d'informations sur les différentes pratiques peuvent lui éviter de tomber dans le piège ! La montée en puissance des réseaux sociaux sur le Web a donné encore plus d'importance à ce type d'attaque (voir chap. 3, *Attaque par ingéniérie sociale*).

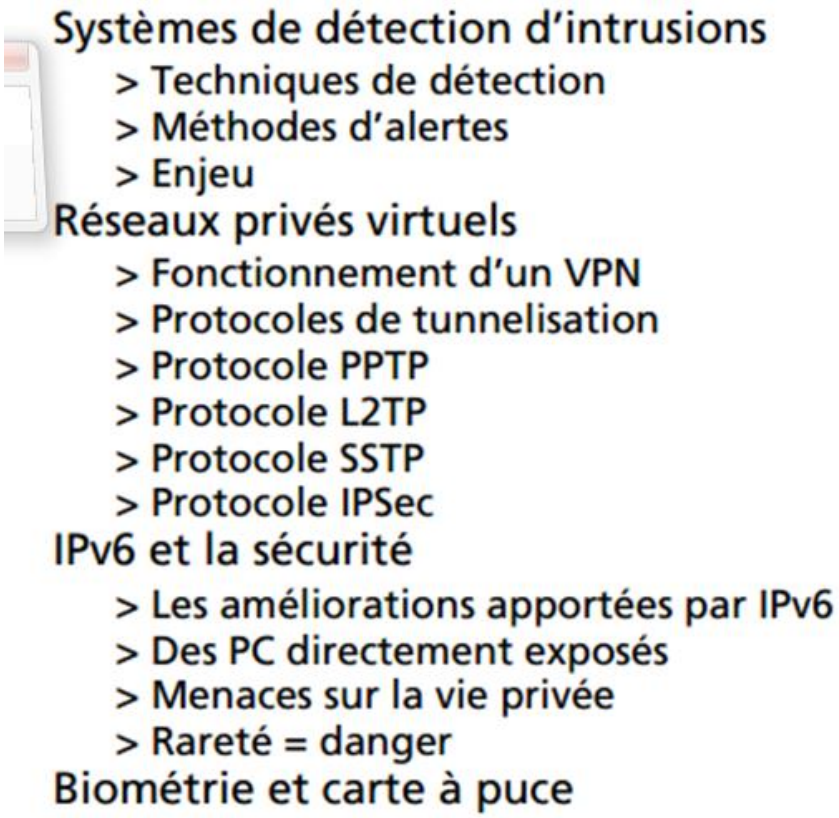
- **Trappes** : il s'agit d'une porte dérobée (*backdoor*) dissimulée dans un logiciel, permettant un accès ultérieur à son concepteur.

I.2.2 PROTECTION

Antivirus
> Principe de fonctionnement
> Détection des *malwares*
> Antivirus mais pas seulement

Système pare-feu (*firewall*)
> Principe de fonctionnement
> Pare-feu personnel
> Zone démilitarisée (DMZ)
> Limites des systèmes pare-feu
> *Honeypots*

Serveurs mandataires (proxy)
> Principe de fonctionnement
> Fonctionnalités d'un serveur proxy
> Translation d'adresses (NAT)
> *Reverse-proxy*



I.3 LES PROGRAMMES MALVEILLANTS (MALWARES)

On appelle **malware** (ou **programme malveillant**, **maliciel**) un programme ou une partie de programme destiné à perturber, altérer ou détruire tout ou partie des éléments logiciels indispensables au bon fonctionnement d'un système informatique.

On distingue principalement sept types de programmes malveillants : les virus informatiques, les bombes logiques, les vers, les chevaux de Troie, les *rootkits*, les *keyloggers* et les *spywares*.

I.3.1 LES VIRUS

I.3.1.1 GENERALITES

Un virus est un petit programme informatique situé dans le corps d'un autre, qui, lorsqu'on le lance, se charge en mémoire et exécute les instructions que son auteur a programmées. La définition d'un virus pourrait être la suivante :

« Tout programme d'ordinateur capable d'infecter un autre programme d'ordinateur en le modifiant de façon à ce qu'il puisse à son tour se reproduire. »

Le véritable nom donné aux virus est **CPA** (soit code auto-propageable), mais par analogie avec le domaine médical, le nom de **virus** leur a été donné.

Les **virus résidents** (appelés **TSR**, *Terminate and Stay Resident*) se chargent dans la mémoire vive de l'ordinateur afin d'infecter les fichiers exécutables lancés par l'utilisateur. Les virus non-résidents infectent les programmes présents sur le disque dur dès leur exécution.

Le champ d'application des virus va de la simple balle de ping-pong qui traverse l'écran au virus destructeur de données, ce dernier étant la forme de virus la plus dangereuse. Avec la « professionnalisation » de la création des virus, ces virus destructeurs ont pratiquement disparu : il ne rapporte rien de détruire des données. Par contre, des virus cryptant les fichiers existent : ces malwares servent à des maîtres chanteurs pour récupérer de l'argent de leur victime en échange de la clé pour décrypter leurs documents.

I.3.2.2 TYPES DE VIRUS

Types de virus

❑ Virus mutants

En réalité, la plupart des virus sont des clones, ou plus exactement des **virus mutants**, c'est-à-dire des virus ayant été réécrits par d'autres utilisateurs afin d'en modifier leur comportement ou leur signature.

Le fait qu'il existe plusieurs versions (on parle de **variantes**) d'un même virus le rend d'autant plus difficile à repérer dans la mesure où

les éditeurs d'antivirus doivent ajouter ces nouvelles signatures à leurs bases de données.

❑ Virus polymorphes

Dans la mesure où les antivirus détectent notamment les virus grâce à leur signature (la succession de bits qui les identifie), certains créateurs de virus ont pensé à leur donner la possibilité de modifier automatiquement leur apparence, tel un caméléon, en dotant les virus de fonction de chiffrement et de déchiffrement de leur signature, de façon à ce que seuls ces virus soient capables de reconnaître leur propre signature. Ce type de virus est appelé **virus polymorphe** (mot provenant du grec signifiant « qui peut prendre plusieurs formes »).

❑ Rétrovirus

On appelle **rétrovirus** ou **virus flibustier** (*bounty hunter*) un virus ayant la capacité de modifier les signatures des antivirus afin de les rendre inopérants.

❑ Virus trans-applicatifs (virus de macros)

Avec la multiplication des programmes utilisant des macros, Microsoft a mis au point un langage de script commun pouvant être inséré dans la plupart des documents pouvant contenir des **macros**, il s'agit de VBScript, un sous-ensemble de Visual Basic. Ces virus ont connu leurs heures de « gloire » avec notamment, les anciennes versions de la suite bureautique de Microsoft. De tel virus pouvaient être situé à l'intérieur d'un banal document Word ou Excel, et exécuter une portion de code à l'ouverture de celui-ci lui permettant d'une part de se propager dans les fichiers, mais aussi d'accéder au système d'exploitation (généralement Windows).

Le début du troisième millénaire a été marqué par l'apparition, à grande fréquence, de scripts Visual Basic diffusés par mail en fichier attaché (repérables grâce à leur extension .VBS) avec un titre de mail poussant à ouvrir le cadeau empoisonné. Ceux-ci avaient la possibilité,

Le début du troisième millénaire a été marqué par l'apparition, à grande fréquence, de scripts Visual Basic diffusés par mail en fichier attaché (repérables grâce à leur extension .VBS) avec un titre de mail poussant à ouvrir le cadeau empoisonné. Ceux-ci avaient la possibilité,

lorsqu'ils étaient lancés à partir d'un client de messagerie Microsoft, d'accéder à l'ensemble du carnet d'adresses et de s'autodiffuser par le réseau. Ce type de virus est appelé **ver** (*worm*).

I.3.2.3 EVITER LES VIRUS

Comme tous les programmes informatiques les virus sont des données interprétables par le système d'exploitation.

Ainsi, tous les fichiers exécutables ou interprétables par le système d'exploitation peuvent potentiellement infecter votre ordinateur. Les fichiers comportant notamment les extensions suivantes sont potentiellement susceptibles d'être infectés : exe, com, bat, pif, vbs, scr, doc, xls, msi, eml.

Attention !

Sous Windows, il est conseillé de désactiver la fonction « *masquer les extensions* », car cette fonction peut tromper l'utilisateur sur la véritable extension d'un fichier. Ainsi un fichier dont l'extension est .jpg.vbs apparaîtra comme un fichier d'extension .jpg !

Cependant, n'importe quel fichier peut être susceptible d'embarquer un virus. Pour cela, il suffit que le programme qui ouvre ces données et les interprètent soit mal conçu. Par exemple, en ne fermant pas correctement des champs de données, il devient possible d'exploiter cette faille par un débordement de tampon (*buffer overflow*) et ajouter un code exécutable par l'ordinateur. Une autre faille peut provenir de l'inclusion d'appels à des éléments infectés se trouvant sur le Web. Les fichiers d'animations Flash (swf), le format d'archive pdf, les fichiers images au format WMF et d'autres ont été la cible de ce type de virus. Il faut donc garder à l'esprit que toutes les applications d'un ordinateur doivent être régulièrement mises à jour.

Ensuite, pour tous les fichiers interprétables directement par le système d'exploitation, seul un antivirus à jour est capable de dire si un programme ne comporte pas d'hôte indésirable. Les éditeurs de ces logiciels proposent des scanners en ligne accessible gratuitement sur http://www.bitdefender.fr/scanner/online/free.html ou encore sur http://www.virustotal.com/fr/ qui sollicite un très grand nombre de moteurs d'antivirus pour analyser un fichier.

I.3.2 LES VERS RESEAU

I.3.2.1 GENERALITES

Un ver est un programme qui peut se reproduire et se déplacer à travers un réseau en utilisant ses mécanismes, sans avoir réellement besoin d'un support physique ou logique (disque dur, programme hôte, fichier…) pour se propager ; un ver est donc un **virus réseau**. La plus célèbre anecdote à propos des vers date de 1988. Un étudiant (Robert T. MORRIS, de Cornell University) avait fabriqué un programme capable de se propager sur un réseau, il le lança et, 8 heures après l'avoir lâché, celui-ci avait déjà infecté plusieurs milliers d'ordinateurs. C'est ainsi que de nombreux ordinateurs sont tombés en pannes en quelques heures car le « ver » (car c'est bien d'un ver dont il s'agissait) se reproduisait trop vite pour qu'il puisse être effacé sur le réseau. De plus, tous ces vers ont créé une saturation au niveau de la bande passante, ce qui a obligé la NSA à arrêter les connexions pendant une journée.

I.3.2.2 PRINCIPE DE FONCTIONNEMENT

Voici la façon dont le ver de Morris se propageait sur le réseau :

- il s'introduisait sur une machine de type Unix ;
- il dressait une liste des machines qui lui étaient connectées ;
- il forçait les mots de passe à partir d'une liste de mots ;
- il se faisait passer pour un utilisateur auprès des autres machines ;
- il créait un petit programme sur la machine pour pouvoir se reproduire ;
- il se dissimulait sur la machine infectée, et ainsi de suite.

Aujourd'hui les vers sont une espèce virale en voie d'extinction du point de vue des systèmes d'exploitation. Sur des machines non maintenues à jour, ils peuvent encore se propager grâce à la messagerie (et notamment par le client de messagerie Outlook) et sous la forme de code directement exécuté par le client de messagerie (JavaScript, VBS…). Mais les vers ont encore un avenir certain dans les applications web : des sites de réseaux sociaux (Facebook, Myspace) et des mondes virtuels (World of Warcraft, Second World) ont vu apparaître des scripts utilisant les langages de ces applications ou JavaScript et pouvant se répandre sur les comptes de tous les utilisateurs de ces services. Là aussi, on peut parler de vers réseau même si leur impact est limité.

I.3.2.3 PARADES

Du point de vue d'un poste client, il est simple de se protéger d'une infection par ver. La meilleure méthode consiste à mettre à jour régulièrement le système d'exploitation de sa machine et ses applications surtout si elles communiquent via le réseau. L'utilisation d'un pare-feu est aussi indispensable pour empêcher l'accession à des services réseaux réservés normalement aux activités intranet. Il convient enfin de ne pas ouvrir « à l'aveugle » des fichiers récupérés par e-mail ou des téléchargements.

Quant à éviter la propagation des vers spécifiques à des applications web, le seul moyen est de ne pas utiliser ces services lors d'une attaque.

Attention !

Lorsqu'on fait la liste des applications à surveiller, il ne faut pas se fier à la fonction principale d'une application pour savoir si elle communique ou non avec le réseau. Ainsi, un traitement de texte comme Microsoft Word, dans ses versions récentes (> 2000) peut avoir une activité réseau. Le logiciel de lecture multimédia Winamp comporte un navigateur web…

I.3.3 Cheval de Troie (informatique)

Un **cheval de Troie** (*Trojan horse* en anglais) est un type de logiciel malveillant, qui ne doit pas être confondu avec les virus ou autres parasites. Le cheval de Troie est un logiciel en apparence légitime, mais qui contient une fonctionnalité malveillante. Le rôle du cheval de Troie est de faire entrer ce parasite sur l'ordinateur et de l'y installer à l'insu de l'utilisateur.

Un cheval de Troie informatique est un programme d'apparence inoffensive, mais qui en contient un autre, malveillant celui-là et qui est installé par l'utilisateur lui-même, ignorant qu'il fait pénétrer un intrus malveillant sur son ordinateur. C'est par analogie, que ce type de programme a été baptisé « cheval de Troie », en référence à la ruse qu'Ulysse utilisa pour contourner les défenses adverses.

En 2014, une étude de l'Association of Internet Security Professionnals centrée sur les dangers du live streaming illégal révèle qu'un ordinateur sur trois est infecté par un logiciel malveillant et que 73 % de ces infections proviennent d'un cheval de Troie[7].

Quelques précisions terminologiques sur le cheval de Troie

Le programme contenu (ou téléchargé par la suite automatiquement) est appelé la "charge utile"[Par qui ?]. Il peut s'agir de n'importe quel type de parasite : virus,

keylogger, logiciel espion... C'est ce parasite qui va exécuter des actions au sein de l'ordinateur victime[8]. Le cheval de Troie n'est rien d'autre que le véhicule, celui qui fait "entrer le loup dans la bergerie". Il n'est pas nuisible en lui-même car il n'exécute aucune action, si ce n'est celle de permettre l'installation du vrai parasite.

Dans le langage courant, par métonymie on nomme souvent "cheval de Troie" le parasite contenu à l'intérieur. Cette confusion est en partie alimentée par les éditeurs d'antivirus, qui utilisent "trojan" comme nom générique pour désigner différents types de programmes malveillants qui n'ont rien à voir avec des trojans[9].

Vecteurs d'infection

Le cheval de Troie prend l'apparence d'un logiciel existant, légitime et parfois même réputé, mais qui aura été modifié pour y dissimuler un parasite. La subtilité avec laquelle l'installation est faite est expliquée par Ken Thompson dans sa conférence Turing. Berné, l'utilisateur va télécharger et installer le programme, pensant avoir affaire à une version saine. En réalité, le logiciel véhicule un parasite qui va pouvoir s'exécuter sur son ordinateur. Les logiciels crackés peuvent être des chevaux de Troie qui vont allécher l'internaute qui cherche à obtenir gratuitement[10] un logiciel normalement payant (Adobe Acrobat pro, Photoshop, Microsoft Office..).

Origines fréquentes des chevaux de Troie

- Téléchargement de versions trafiquées sur des sites non officiels ou des plateformes peu sûres (P2P). Télécharger les logiciels sur le site officiel de l'auteur ou du distributeur évite normalement d'avoir affaire à une version infectée par un cheval de trois. Cela n'est évidemment pas possible pour se procurer des versions crackées, mais faisable pour tous les logiciels gratuits.[pas clair]
- Téléchargement de programmes P2P.
    - Visite de sites Web contenant un exécutable (par exemple les contrôles ActiveX ou des applications Java).
- Exploitation de failles dans des applications obsolètes (navigateurs, lecteurs multimédias, clients de messagerie instantanée) et notamment les Web Exploit.
- Ingénierie sociale (par exemple, un pirate envoie directement le cheval de Troie à la victime par messagerie instantanée).
- Pièces jointes et fichiers envoyés par messagerie instantanée.
- Connexion d'un ordinateur à un périphérique externe infecté.
- Mise à jour de logiciel.
- Absence de logiciel de protection.

Notions voisines du cheval de Troie

Le cheval de Troie ne doit pas être confondu avec d'autres notions proches :

- L'**injecteur** (ou *dropper*, en anglais) est quasiment identique au cheval, car il sert lui aussi de véhicule pour une malveillance. Mais l'injecteur est un programme spécialement fabriqué pour propager des parasites, alors que le cheval est une version modifiée d'un programme existant et légitime.

- La **porte dérobée** (*backdoor*) est un programme qui va s'exécuter discrètement sur l'ordinateur où il est installé pour y créer une faille de sécurité. Le backdoor ouvre un ou plusieurs ports sur la machine, ce qui lui permet d'accéder à internet librement et de télécharger, à l'insu de l'utilisateur, un parasite. Le backdoor n'est donc pas un cheval de Troie : il ne véhicule pas le parasite en lui, il va simplement ouvrir l'accès et récupérer, via internet, le programme malveillant qui se trouve sur un serveur distant.

- Le **RAT (Remote administration tool)** est un logiciel de prise de contrôle à distance d'un ordinateur. Un RAT peut être un outil légitime (par exemple pour le dépannage à distance), mais il peut aussi être utilisé par un pirate pour s'emparer d'une machine. Dans ce cas, l'introduction du RAT sur la machine à contrôler se fait à l'insu de l'utilisateur. Par exemple, par un cheval de Troie qui contient le RAT, mais le RAT n'est pas le cheval. Contrairement à ce qu'on lit parfois, le T de RAT ne signifie pas *Trojan* mais *Tool* (outil).

- Les **bombes de décompression** ne transportent pas de parasite, mais elles peuvent être confondues avec les chevaux de Troie car la notion de conteneur entre aussi en jeu. Il s'agit d'un fichier compressé, par exemple un fichier zip, de taille raisonnable tant qu'il n'est pas ouvert. Mais lorsque l'utilisateur va tenter de la décompresser, elle va générer un fichier d'une taille gigantesque. Cette "explosion" entraîne le ralentissement ou le plantage de l'ordinateur, et sature le disque dur avec des données inutiles. Bien qu'il s'agisse de conteneurs malveillants, le fonctionnement des bombes de décompression n'a donc rien à voir avec celui des chevaux de Troie. En effet, elles ne transportent aucun parasite indépendant, elles saturent la machine de données aléatoires.
- Le Trojan Stealer, plutôt spécialisé dans le vol de données et notamment les comptes en ligne (Mail, Réseaux sociaux ou encore bancaire). Le préfixe utilisé par les antivirus peut alors être Trojan.PWD où PWD signifie password pour mot de passe.

Symptômes possibles d'une infection par le cheval de Troie

- Activité anormale de la carte réseau ou du disque dur (des données sont chargées en l'absence d'activité de la part de l'utilisateur) ou du modem
- Réactions curieuses de la souris
- Ouvertures impromptues de programmes, du lecteur CD/DVD
- Plantages répétés
- Redémarrage répété du système
- Écran ou fenêtres avec des messages inhabituels.

- Un comportement inhabituel dans le fonctionnement de l'ordinateur, tels que: changements d'économiseur d'écran de bureau, modification du rôle des boutons de la souris, modification du volume du lecteur audio.
- Ouverture/Fermeture intempestive de fenêtres.
- Les programmes commencent ou terminent leur exécution de manière inattendue.
- Le navigateur accède tout seul à certains sites Internet.
- Présence d'autres programmes qui n'ont pas été volontairement installés (y compris des logiciels malveillants).
- Vol de renseignements personnels : informations bancaires, mots de passe, codes de sécurité...
- Suppression, modification ou transfert de fichiers (téléchargement ou upload).
- Exécution ou arrêt de processus.
- Arrêt ou redémarrage impromptus de l'ordinateur.
- Surveillance des frappes (voir Enregistreur de frappe)
- Captures d'écran impromptues.
- Espace libre du disque dur occupé par des fichiers inutiles.

Prévention et lutte

- Pour lutter contre ce genre de programme malveillant, l'utilisation d'un antivirus peut s'avérer efficace, mais reste souvent insuffisante. Il est conseillé de faire une analyse complète de son système d'exploitation et un nettoyage profond effectué de préférence par un technicien agréé. Dans certains cas, l'utilisateur peut se retrouver obligé de démarrer sur un autre système d'exploitation, puis de redémarrer en mode sans échec pour pouvoir reprendre la main.

I.3.4 BOMBES LOGIQUES

Sont appelés **bombes logiques** les logiciels dont le déclenchement s'effectue à un moment déterminé en exploitant la date du système, le lancement d'une commande, ou n'importe quel appel au système. Ainsi ce type de virus est capable de s'activer à un moment précis sur un grand nombre de machines (on parle alors de **bombe à retardement** ou de **bombe temporelle**), par exemple le jour de la saint Valentin, ou la date anniversaire d'un événement majeur : la bombe logique Tchernobyl s'est activée le 26 avril 1999, jour du 13e anniversaire de la catastrophe nucléaire.

Les bombes logiques sont généralement utilisées dans le but de créer un déni de service en saturant les connexions réseau d'un site, d'un service en ligne ou d'une entreprise ! Ce type de programme malveillant est devenu très rare de nos jours.

I.3.5 SPYWARES (ESPIOGICIEL)

I.3.5.1 GENERALITES

Un **spyware** (espiogiciel) est un programme chargé de recueillir des informations sur l'utilisateur de l'ordinateur dans lequel il est installé

(on l'appelle donc parfois **mouchard**) afin de les envoyer à la société qui le diffuse pour lui permettre de dresser le profil des internautes (on parle de **profilage**).

Les récoltes d'informations peuvent ainsi être :

- les adresses web (URL) des sites visités,
- les mots-clés saisis dans les moteurs de recherche,
- l'analyse des achats réalisés via Internet, voire les informations de paiement bancaire (numéro de carte bleue/VISA),
- des informations personnelles (numéro de sécurité sociale, etc.).

Les *spywares* s'installent généralement en même temps que d'autres logiciels (la plupart du temps des *freewares* ou *sharewares*). En effet, cela permet aux auteurs desdits logiciels de rentabiliser leur programme, par de la vente d'informations statistiques, et ainsi permettre de distribuer leur logiciel gratuitement. Il s'agit donc d'un modèle économique dans lequel la gratuité est obtenue contre la cession de données à caractère personnel.

Les *spywares* ne sont pas forcément illégaux car la licence d'utilisation du logiciel qu'ils accompagnent précise que ce programme tiers va être installé ! En revanche étant donné que la longue licence d'utilisation est rarement lue en entier par les utilisateurs, ceux-ci savent très rarement qu'un tel logiciel effectue ce profilage dans leur dos.

Par ailleurs, outre le préjudice causé par la divulgation d'informations à caractère personnel, les *spywares* peuvent également être une source de nuisances diverses :

- consommation de mémoire vive,
- utilisation d'espace disque,
- mobilisation des ressources du processeur,
- plantages d'autres applications,
- gêne ergonomique (par exemple l'ouverture d'écrans publicitaires ciblés en fonction des données collectées).

I.3.5.2 TYPES DE SPYWARES

On distingue généralement deux types de *spywares* :

- Les **spywares internes** (ou *spywares intégrés*) comportant directement des lignes de code dédiées aux fonctions de collecte de données.
- Les **spywares externes**, programmes de collectes autonomes installés.

I.3.5.3 PARADES

La principale difficulté avec les *spywares* est de les détecter. La meilleure façon de se protéger est encore de ne pas installer de logiciels dont on n'est pas sûr à 100 % de la provenance et de la fiabilité (notamment les *freewares* et les *sharewares*). Voici quelques exemples (liste non exhaustive) de logiciels connus pour avoir embarqué un ou plusieurs *spywares* : Babylon Translator, GetRight, Go!Zilla, Download Accelerator, Cute FTP, PKZip, KaZaA ou encore iMesh.

Qui plus est, la désinstallation de ce type de logiciels ne supprime que rarement les *spywares* qui l'accompagnent. Pire, elle peut entraîner des dysfonctionnements sur d'autres applications !

Dans la pratique, il est quasiment impossible de ne pas installer de logiciels. Ainsi la présence de processus d'arrière-plan suspects, de fichiers étranges ou d'entrées inquiétantes dans la base de registre peuvent parfois trahir la présence de *spywares* dans le système.

Si vous ne parcourez pas la base de registre à la loupe tous les jours rassurez-vous, il existe des logiciels, nommés **anti-spywares** permettant de détecter et de supprimer les fichiers, processus et entrées de la base de registres créés par des *spywares*.

De plus l'installation d'un **pare-feu personnel** peut permettre d'une part de détecter la présence d'espiogiciels, d'autre part de les empêcher d'accéder à Internet (donc de transmettre les informations collectées).

À savoir

Parmi les anti-spywares les plus connus ou efficaces citons notamment : Ad-Aware de Lavasoft.de, Spybot Search&Destroy et Malware byte's antimalware.

I.3.6 LES RANSOMWARES (RANCOLOGICIEL)

Les **ransomwares** (ou rançongiciels) sont des logiciels malveillants dont le but est de soutirer de l'argent à leurs victimes. Il existe deux types de *ransomwares* :

- Les logiciels malveillants peuvent se contenter de faire peur aux utilisateurs qui les ont lancés (par exemple un virus gendarme qui demande le paiement d'une « amende » à la suite de la détection d'activités illégales sur le poste infecté). Le poste de la victime devient très difficile à utiliser car le logiciel bloque la plupart des applications. Une variante contraint les utilisateurs à cliquer sur des bannières publicitaires pour pouvoir accéder à leur ordinateur.
- Les *ransomwares* « chiffreurs » sont beaucoup plus agressifs car ils chiffrent tout ou partie des stockages de l'ordinateur de la victime. Les données sont cryptées en tâche de fond. Le *malware* affiche ensuite un message demandant le versement d'une rançon contre un moyen de déchiffrer les documents. Certaines variantes désactivent aussi les systèmes de versionning et s'attaquent aux fichiers de sauvegarde.

En 2014 et en 2015, CTB-Locker et Cryptowall ont été les deux crypto-*ransomwares* les plus répandus. L'éditeur Kaspersky a réussi à trouver des clés de déchiffrement pour certaines variantes de ces *malwares* mais, dans la plupart des cas, il n'existe aucun moyen de déchiffrer les données.

Parades

Seul un logiciel antivirus à jour permet d'éviter ce type de *malwares*. On peut en réduire l'impact en procédant à des sauvegardes régulières, en confiant le versionning de fichier à des systèmes déportés (Cloud ou serveur de fichiers). Malheureusement, une fois les fichiers chiffrés, rien ne garantit que les documents pourront être récupérés (pas même le paiement de la rançon qui peut aboutir à télécharger un nouveau *malware* qui va infecter d'autres ordinateurs).

I.3.7 LES KEYLOGGERS (ENREGISTREURS DE TOUCHE)

Un **keylogger** (littéralement enregistreur de touches) est un dispositif chargé d'enregistrer les frappes de touches du clavier et de les enregistrer, à l'insu de l'utilisateur. Il s'agit donc d'un dispositif d'espionnage. Certains *keyloggers* sont capables d'enregistrer les URL visitées, les courriers électroniques consultés ou envoyés, les fichiers ouverts, voire de créer une vidéo retraçant toute l'activité de l'ordinateur !

Dans la mesure où les *keyloggers* enregistrent toutes les frappes de clavier, ils peuvent servir à des personnes mal intentionnées pour récupérer les mots de passe des utilisateurs du poste de travail ! Cela signifie donc qu'il faut être particulièrement vigilant lorsque vous utilisez un ordinateur en lequel vous ne pouvez pas avoir confiance (poste en libre accès dans une entreprise, une école ou un lieu public tel qu'un cybercafé).

Les *keyloggers* peuvent être soit logiciels soient matériels. Dans le premier cas il s'agit d'un processus furtif (ou bien portant un nom ressemblant fortement au nom d'un processus système), écrivant les informations captées dans un fichier caché ! Les *keyloggers* peuvent également être matériels : il s'agit alors d'un dispositif (câble ou *dongle*) intercalé entre la prise clavier de l'ordinateur et le clavier. Des moyens d'interception par écoute radio existent aussi (voir chap. 3, *Attaque par faille matérielle*).

Parades

La meilleure façon de se protéger est la vigilance :

- Ne pas installer de logiciels dont la provenance est douteuse.

- Prudence lors de la connexion à partir d'un ordinateur tiers (à partir d'un cybercafé par exemple) ! S'il s'agit d'un ordinateur en accès libre, il peut être utile d'examiner rapidement les programmes actifs en mémoire, avant de se connecter à des sites demandant un mot de passe. En cas de doute, il est conseillé de ne pas se connecter à des sites sécurisés pour lesquels un enjeu existe (banque en ligne, etc.).

I.3.8 LES SPAMS(Pourriels)

On appelle **spam** ou **pollupostage** (les termes pourriel, courrier indésirable ou *junk mail* sont parfois également utilisés) l'envoi massif de courrier électronique (souvent de nature publicitaire) à des destinataires ne l'ayant pas sollicité.

Les personnes pratiquant l'envoi massif de courrier publicitaire sont appelées **spammers** (en français spammeurs), un mot qui a désormais une connotation péjorative !

Le but premier du spam est de faire de la publicité à moindre prix par « envoi massif de courrier électronique non sollicité » ou par « multipostage abusif ». Les spammeurs prétendent parfois, en toute mauvaise foi, que leurs destinataires se sont inscrits spontanément à leur base de données et que le courrier ainsi reçu est facile à supprimer, ce qui constitue au final un moyen écologique de faire de la publicité.

Les spammeurs collectent des adresses électroniques sur Internet (dans les forums, sur les sites Internet, dans les groupes de discussion, etc.) grâce à des logiciels (appelés **robots**) parcourant les différentes pages et stockant au passage dans une base de données toutes les adresses e-mail y figurant. Il ne reste ensuite au spammeur

L'activité de spam étant combattue mondialement, bon nombre de spammeurs actuels utilisent des réseaux de machines piratées (*botnet*) pour envoyer leur courrier non sollicité. Ceci leur permet de se dissimuler derrière ces milliers de machines et de disposer gratuitement d'une très grande bande passante pour l'envoi de millions de courriers en même temps. Preuve de l'existence de ces *botnets* géants, en novembre 2008, la déconnexion de l'hébergeur américain McColo Corp a fait baisser le spam mondial d'environ 50 % en une seule journée. De 50 spam/seconde, on est passé à moins de 20 spam/s, preuve de la puissance des *botnets*[1] qui étaient contrôlés via ce réseau. En 2010, la mise hors service du *botnet* Waledac par Microsoft a stoppé ce réseau qui envoyait 1,5 milliards de spams par jour. En 2011, Microsoft a aussi mis fin au réseau constitué avec le malware Rustock qui comprenait environ 1 million d'ordinateurs et a généré 47 % du spam mondial, pendant ses quatre années d'existence. Fin 2015, le spam mondial n'a toujours pas retrouvé son niveau de 2010. Sur l'année 2015, Spamcop.net comptabilise en moyen 8 spams par seconde. Les principaux gestionnaires de spam parlent d'environ 60 % d'e-mails indésirables contre 80-90 % à la fin des années 2010. Le ratio coût/bénéfice ne doit plus être assez intéressant pour les spammeurs par rapport aux autres techniques des cybercriminels.

Inconvénients

Les **inconvénients** majeurs du spam sont :

- l'espace qu'il occupe dans les boîtes aux lettres des victimes ;
- la difficile consultation des messages personnels ou professionnels au sein de nombreux messages publicitaires et l'augmentation du risque de suppression erronée ou de non-lecture de messages importants ;
- la perte de temps occasionnée par le tri et la suppression des messages non sollicités ;

- le caractère violent ou dégradant des textes ou images véhiculés par ces messages, pouvant heurter la sensibilité des plus jeunes ;
- la bande passante qu'il gaspille sur le réseau des réseaux.

Le spam induit également des coûts de gestion supplémentaires pour les fournisseurs d'accès à Internet (FAI), se répercutant sur le coût de leurs abonnements. Ce surcoût est notamment lié à :

- la mise en place des systèmes antispam ;
- la sensibilisation des utilisateurs ;
- la formation du personnel ;
- la consommation de ressources supplémentaires (serveurs de filtrage, etc.).

Parades

Les spammeurs utilisent généralement de fausses adresses d'envoi, il est donc totalement inutile de répondre. Qui plus est une réponse peut indiquer au spammeur que l'adresse est active et induire encore plus de spam.

De la même façon, lorsque vous recevez un spam (courrier non sollicité), il peut arriver qu'un lien en bas de page vous propose de ne plus recevoir ce type de message. Si tel est le cas il y a de grandes chances pour que le lien permette au spammeur d'identifier les adresses actives. Il est ainsi conseillé de supprimer le message sans tenter de se désabonner.

La plupart des clients de messagerie actuels permettent de ne pas télécharger les éléments d'un courrier rédigé en HTML. Ne désactivez pas cette fonctionnalité : en effet, si vous décidez de télécharger automatiquement le contenu externe de l'e-mail, vous indiquerez au spammeur que vous avez bien visualisé son spam et que votre adresse est active.

❑ Les antispam

Il existe également des dispositifs antispam permettant de repérer et, le cas échéant, de supprimer les messages indésirables sur la base de règles évoluées. On distingue généralement deux familles de logiciels antispam :

- Les dispositifs antispam **côté client**, situé au niveau du client de messagerie. Il s'agit généralement de systèmes possédant des filtres permettant d'identifier les messages indésirables, sur la base de règles prédéfinies, de liste d'adresse IP

d'envoi référencé comme spammeur, ou d'un apprentissage (filtres bayésiens). Des dispositifs d'authentification existent aussi pour lutter contre le spam (voir chapitre 6).

- Les dispositifs antispam **côté serveur**, permettant un filtrage du courrier avant remise aux destinataires. Ce type de dispositif est de loin le meilleur car il permet de stopper le courrier non sollicité en amont et éviter l'engorgement des réseaux et des boîtes aux lettres. Une solution intermédiaire consiste à configurer le dispositif antispam du serveur de façon à marquer les messages avec un champ d'en-tête spécifique (par exemple X-Spam-Status : Yes). Grâce à ce marquage, il est aisé de filtrer les messages au niveau du client de messagerie.

En cas d'encombrement ou de saturation totale de la boîte aux lettres, la solution ultime consiste à changer de boîte aux lettres. Il est toutefois conseillé de garder l'ancienne boîte aux lettres pendant un laps de temps suffisant afin de récupérer les adresses de vos contacts et d'être en mesure de communiquer la nouvelle adresse aux seules personnes légitimes.

Afin d'éviter le spam, il est nécessaire de divulguer son adresse électronique le moins possible et à ce titre :

- Ne pas relayer les messages (blagues, etc.) invitant l'utilisateur à transmettre le mail au maximum de contacts possibles ou en s'assurant de masquer les adresses des destinataires précédents. De telles listes sont effectivement des aubaines pour les collecteurs d'adresses.

- Éviter au maximum la publication de son adresse e-mail sur des forums ou des sites internet.
- Dans la mesure du possible remplacer son adresse e-mail par une image (moins détectable par les aspirateurs d'adresses).
- Créer une ou plusieurs « adresses poubelles » servant uniquement à s'inscrire ou s'identifier sur les sites jugés non dignes de confiance. Le comble du raffinement, lorsque vous en avez la possibilité, consiste à créer autant d'alias d'adresses que d'inscription en veillant à y inscrire le nom de l'entreprise ou du site. Ainsi, en cas de réception d'un courrier non sollicité il sera aisé d'identifier d'où provient la fuite (qui a « vendu la mèche »).

I.3.9 LES ROOTKITS

I.3.9.1 GENERALITES

Un ***rootkit*** (le nom « **outil de dissimulation d'activité** » est également utilisé[1], ainsi que « **maliciel furtif** »[2] et « **trousse administrateur pirate** »[3]), parfois simplement « **kit** », est un ensemble de techniques mises en œuvre par un ou plusieurs logiciels, dont le but est d'obtenir et de pérenniser un accès (généralement non autorisé) à un ordinateur de la manière la plus furtive possible[4,C 1,L 1], à la différence d'autres logiciels malveillants. Le terme peut désigner la technique de dissimulation ou plus généralement un ensemble particulier d'objets informatiques mettant en œuvre cette technique.

Leur furtivité est assurée par plusieurs mécanismes de dissimulation (voir *infra*) : effacement de traces, masquage de l'activité et des communications, etc. Un *rootkit* peut s'installer dans un autre logiciel, une bibliothèque ou dans le noyau d'un système d'exploitation. Certains peuvent modifier l'hyperviseur fonctionnant en dessous des systèmes ou le micrologiciel intégré dans un matériel. La plupart des *rootkits* servent à installer des logiciels malveillants sur les machines où l'accès est obtenu. Certains fournisseurs de matériels informatiques, tel Sony, les utilisent pour s'assurer du respect des conditions d'utilisation de leurs produits par leurs clients. Certains kits ne jouent pas sur la discrétion mais sur le fait qu'enlever le kit serait une opération ardue[L 2].

Pour l'« attaquant », l'utilité d'un *rootkit* est soit de mettre à disposition des ressources système (temps processeur, connexions réseaux, etc.) sur une, voire plusieurs machines (voir *infra*), parfois en utilisant la « cible » comme intermédiaire pour une autre attaque ; soit d'espionner, d'accéder aux données stockées ou en transit sur la machine cible[L 2].

Ils sont généralement classés parmi les logiciels malveillants, mais pas toujours[L 1] ; ils peuvent utiliser des « techniques virales » pour se transmettre (par exemple, en utilisant un virus ou un cheval de Troie)[5]. Il existe des outils de détection et des méthodes de protection pour les contrer mais elles ne sont pas totalement efficaces.

Moyens de détection

La mise en œuvre d'une détection peut, selon le type de rootkit, demander un examen du système ou d'un périphérique suspect en mode « inactif » (démarrage à partir d'un système de secours ou d'un système réputé sain). Plusieurs méthodes existent.

La *recherche d'objets cachés* (tels que des processus informatiques, des clés de registre, des fichiers, etc.) est essentielle. Des outils comme unhide sous Linux peuvent révéler les processus cachés. Sous Windows, des outils comme RootkitRevealer recherchent les fichiers cachés en listant les fichiers via l'API normale de Windows puis en comparant cette liste à une lecture physique du disque ; les différences entre les deux sont alors repérées comme

suspectes, à l'exception des fichiers légitimes connus de Windows, tels que les fichiers de métadonnées de NTFS comme $MFT ou $Secure[49].

Le calcul régulier des empreintes de fichiers sensibles permet de détecter une modification inattendue. Le contrôle de l'intégrité des fichiers consiste à calculer, pour chaque fichier sensible (bibliothèque, commande système, etc.), une empreinte[13]. Toute modification inattendue de cette empreinte indique une modification du fichier, et donc une contamination potentielle. Cependant, tout système subit des modifications légitimes lors des mises à jour ; idéalement, l'outil de contrôle a la possibilité d'accéder à une base de référence de ces sommes de contrôles, selon la version du système utilisée (rkhunter par exemple).

La *détection de signatures spécifiques* est le procédé classique d'analyse de signature, comme cela se fait pour les virus : on cherche à retrouver dans le système la trace d'une infection, soit directement (signature des objets du rootkit), soit par le vecteur d'infection (virus utilisé par le rootkit)[13].

L'analyse des appels systèmes, des tables d'interruption[50,51], et de manière générale, des tables de travail utilisées par le système, au moyen d'outils spécifiques (des logiciels anti-espion comme HijackThis), permet de voir si ces appels ont été détournés ou non, par exemple en comparant ce qui est chargé en mémoire avec les données brutes de bas niveau (ce qui est écrit sur le disque).

Le hooking consiste à détourner un appel de fonction légitime par un autre qui contient du code malveillant.

On peut aussi s'intéresser à **la charge du système**. Du point de vue du processeur et de l'activité applicative, une surveillance continue peut mettre en évidence une surcharge, à partir du moment de la contamination. Il s'agit essentiellement d'une analyse de la charge habituelle de la machine, comme le nombre de mails sortants ou l'occupation du processeur. Toute modification (en surcharge) sans cause apparente est suspecte, mais elle nécessite une analyse complémentaire pour écarter les causes légitimes (mise à jour du système, installation de logiciels, etc.)

De la même manière, **l'analyse des flux réseau**[52] permet de détecter une surcharge anormale. Mais il convient également de surveiller une utilisation de ports logiciels inhabituels grâce aux traces issues d'un pare-feu ou grâce à un outil spécialisé. Il est également possible de faire une recherche des ports ouverts et cachés, en comparant ce que connaît le système avec ce qui est effectivement ouvert, grâce à des outils d'investigation comme unhide-tcp. Toute différence peut être considérée comme anormale. Il existe cependant des moyens de dissimulation réseau, comme de la stéganographie ou l'utilisation de canaux cachés, qui rend la détection directe impossible, et nécessite une analyse statistique qui n'est pas forcément déterminante[53].

L'analyse automatisée des logs système[54] s'appuie sur le principe de corrélation, avec des outils de type HIDS qui disposent de règles paramétrables[55] pour repérer les événements anormaux et mettre en relation des événements systèmes distincts, sans rapport apparent ou épars dans le temps.

Le site du Cert-IST propose régulièrement des informations sur les rootkits et les logiciels malicieux en général[56].

Moyens de protection et de prévention

Les moyens de détection peuvent également servir à la prévention, même si celle-ci sera toujours postérieure à la contamination. D'autres mesures en amont peuvent rendre difficile l'installation d'un rootkit[57].

La correction des failles par mise à jour du système d'exploitation permet de réduire la surface d'exposition du système en limitant le temps pendant lequel une faille est présente sur le système[58] et dans les applications[54], afin de prévenir les exploits pouvant être utilisés pour la contamination.

L'utilisation d'un pare-feu, qui fait partie des bonnes pratiques dans le domaine de la sécurité informatique, se révèle efficace dans le cas des rootkits[51,54,58] car cela empêche les communications inattendues (téléchargements de logiciel, dialogue avec un centre de contrôle et de commande d'un botnet, etc.) dont ont besoin les rootkits.

Il est possible de **désactiver le système de chargement de modules en rendant le noyau statique**, ce qui protège contre les rootkits qui s'installent en chargeant un module ; certains rootkits arrivent cependant à contourner cela en reconnaissant l'empreinte du module directement dans la mémoire[C 9].

De même, pour renforcer **la robustesse des bibliothèques** et empêcher le hooking, il est possible de compiler statiquement les bibliothèques[C 10].

La complexité d'un mot de passe augmente exponentiellement lorsque sa taille et le nombre de caractères différents qu'il utilise augmentent. Un mot de passe complexe sera plus long à deviner dans une attaque par force brute.

Des systèmes de prévention d'intrusion[54], sous forme de logiciel ou de matériel, répondent dès qu'une intrusion est détectée, en bloquant des ports ou en interdisant la communication avec une source (adresse IP) douteuse, ou toute autre action appropriée. La détection sera d'autant meilleure que l'outil utilisé sera externe au système examiné, puisque certains rootkits peuvent atteindre des parties de très bas niveau dans le système, jusqu'au BIOS. Un des avantages de ces outils est l'automatisation des tâches de surveillance[13].

Des outils spécialisés de contrôle d'intégrité des fichiers peuvent produire des alertes lors de modifications inattendues. Cependant, ce contrôle à lui seul est insuffisant si

d'autres mesures préventives ne sont pas mises en œuvre, si aucune réponse du système n'est déclenchée ou si ces différences ne sont pas analysées.

Le renforcement de la robustesse des mots de passe est une autre des bonnes pratiques de sécurité informatique qui élimine une des sources principales de contamination. Des éléments d'authentification triviaux sont des portes ouvertes pour tout type d'attaque informatique.

Le démarrage du système à partir d'une image saine, contrôlée et réputée valide du système d'exploitation, via un support fixe (comme un LiveCD, une clé USB) ou par réseau, permet de s'assurer que les éléments logiciels principaux du système ne sont pas compromis, puisqu'à chaque redémarrage de la machine concernée, une version valide de ces objets est chargée. Un système corrompu serait donc remis en état au redémarrage (sauf dans le cas de rootkit ayant infecté un plus bas niveau, comme le BIOS).

Les moyens de protection habituels sont également valables contre les rootkits : « Do everything so that attacker doesn't get into your system »[53]. Durcissement du système[51], filtrages applicatifs (type modsecurity), **utilisation de programmes antivirus**[51,58] pour minimiser la surface d'attaque et surveiller en permanence les anomalies et tentatives de contamination, sont bien sûr à mettre en œuvre pour éviter la contamination du système et l'exposition aux exploits.

Outils et logiciels de détection

À part quelques cas particuliers, l'industrie de la sécurité informatique a tardé à prendre en compte les rootkits, les virus puis les chevaux de Troie accaparant l'attention des éditeurs. Il existe cependant quelques logiciels de détection et de prévention spécifiques à Windows, tels que Sophos Anti-Rootkit ou AVG Anti-Rootkit. Sous Linux, on peut citer rkhunter et chkrootkit ; plusieurs projets open-source existent sur Freshmeat et Sourceforge.net.

Aujourd'hui, il reste difficile de trouver des outils spécifiques de lutte contre les rootkits, mais leur détection et leur prévention sont de plus en plus intégrées dans les systèmes de prévention d'intrusion et même dans les antivirus classiques, lesquels sont de plus en plus obligés de se transformer en suites de sécurité pour faire face à la diversité des menaces ; ils proposent en effet de plus en plus souvent des protections contre les rootkits.

I.4 LES LOGICIELS ANTI-PROGRAMMES MALVEILLANTS (Logiciels de sécurité informatique)

I.4.1 LISTE DE LOGICIELS DE SECURITE INFORMATIQUE

I.4.1.1. Lutte contre les rootkits

Programmes de détection, d'analyse et de prévention de rootkits spécifiques à Windows

- Sophos Anti-rootkit [archive]
- RkU (rootkit unhooker)
- IceSword
- RootkitRevealer de Sysinternals (en), Microsoft[1]

- Blacklight de F-Secure
- Rootkit Hook Analyzer[réf. souhaitée]
- RootAnalyser (Safer Networking)
- HijackThis
- GMER (en)
- DarkSpy
- Trend Micro Rootkit Buster[2]

Programmes de détection et de prévention de rootkits spécifiques à Linux

- chkrootkit de Nelson Murilo et Klaus Steding-Jessen (UNIX/Linux)
- rkhunter de Michael Boelen (UNIX/Linux)
- Zeppoo de ZeppooTeam (UNIX/Linux), renommé kernsh le 15 mai 2007, ce projet est maintenant intégré dans le framework ERESI (18 septembre 2007).
- unhide

I.4.1.2 Outils anti-malware

- Ad-Aware
- AdwCleaner
- Malwarebytes Anti-Exploit
- Malwarebytes Anti-Malware
- Spybot S&D (Safer Networking)
- SpywareBlaster
- SpywareGuard
- Windows Defender

I.4.1.3 Filtres applicatifs

- Modsecurity

I.4.1.4 Recherche de vulnérabilités

- Nikto Web Scanner (en)
- Nessus
- Nmap
- LanGuard de GFI Software (en)[3]
- Elastic Detector [archive]

I.4.1.5 Pare-feux

I.4.1.6 Outils de type suite de sécurité

Il s'agit d'outils ayant plusieurs fonctionnalités de protection, contre les virus mais aussi souvent contre les rootkits, contre les comportements anormaux (par analyse des événements sur le système), etc.

- ThreatFire
- Prevx

I.4.1.7 Anti-virus

Il est également utile de savoir que des outils dits *anti-virus* complètent leurs mécanismes de protection avec la détection de rootkits, le filtrage de courrier, la protection des informations personnelles, les logiciels espions (ou *spywares*), etc (cf p Liste de logiciels antivirus).

I.4.2 LOGICIELS ANTI-VIRUS

I.4.2.1 NOTIONS LOGICIELS ANTI-VIRUS

Les **antivirus** sont des logiciels conçus pour identifier, neutraliser et éliminer des logiciels malveillants (dont les virus informatique[1] ne sont qu'une catégorie).
Ces derniers peuvent se baser sur l'exploitation de failles de sécurité, mais il peut également s'agir de logiciels modifiant ou supprimant des fichiers, que ce soit des documents de l'utilisateur stockés sur l'ordinateur infecté, ou des fichiers nécessaires au bon fonctionnement de l'ordinateur (le plus souvent ceux du système d'exploitation).

Fonctionnement

Un logiciel antivirus vérifie les fichiers et courriers électroniques, les secteurs de démarrage (afin de détecter les virus de boot), mais aussi la mémoire vive de l'ordinateur, les médias amovibles (clefs USB, CD, DVD, etc.), les données qui transitent sur les éventuels réseaux (dont internet), etc.

Différentes méthodes sont possibles :

- Les principaux antivirus du marché se concentrent sur des fichiers et comparent alors la signature virale du virus aux codes à vérifier ;
- La *méthode heuristique* est la méthode la plus puissante, tendant à découvrir un code malveillant par son comportement. Elle essaie de le détecter en analysant le code d'un programme inconnu. Parfois de fausses alertes peuvent être provoquées ;
- L'*analyse de forme* repose sur du filtrage basé entre des règles regexp ou autres, mises dans un fichier junk. Cette dernière méthode peut être très efficace pour les serveurs de messagerie électronique supportant les regexp type postfix puisqu'elle ne repose pas sur un fichier de signatures.

Les antivirus peuvent balayer le contenu d'un disque dur, mais également la mémoire vive de l'ordinateur. Pour les antivirus les plus modernes, ils agissent en amont de la machine en scrutant les échanges de fichiers avec l'extérieur, aussi bien en flux descendant (téléchargement) que montant (téléversement ou upload). Ainsi, les courriels sont examinés, mais aussi les fichiers copiés sur ou à partir de supports amovibles tels que cédéroms, disquettes, connexions réseau, clefs USB…

Approches

On distingue plusieurs types de logiciels antivirus selon leur fonctionnement. La première méthode est celle du dictionnaire.

Dictionnaire

Les créateurs de logiciels antivirus ayant préalablement identifié et enregistré des informations sur des virus, comme le ferait un dictionnaire, le logiciel antivirus peut ainsi détecter et localiser la présence d'un virus.

On appelle ce dictionnaire la base de définition virale qui contient les signatures de virus.

Lorsque cela se produit, l'antivirus dispose de trois options, il peut :

1. Effectuer la suppression du fichier contaminé.
2. Tenter de réparer le fichier endommagé en éliminant le virus ;
3. Déplacer le fichier dans une zone de quarantaine afin qu'il ne puisse être accessible aux autres utilisateurs et logiciels. Ceci permet d'éviter que le virus se répande (par autoréplication), et permet éventuellement de réparer le fichier ultérieurement ;

Afin de maximiser le rendement de l'antivirus, il est essentiel d'effectuer de fréquentes mises à jour en téléchargeant des versions plus récentes. Des internautes consciencieux et possédant de bonnes connaissances en informatique peuvent identifier eux-mêmes des virus et envoyer leurs informations aux créateurs de logiciels antivirus afin que leur base de données soit mise à jour.

Généralement, les antivirus examinent chaque fichier lorsqu'il est créé, ouvert, fermé ou lu. De cette manière, les virus peuvent être identifiés immédiatement. Il est possible de programmer le système d'administration pour qu'il effectue régulièrement un examen de l'ensemble des fichiers sur l'espace de stockage (disque dur, etc).

Même si les logiciels antivirus sont très performants et régulièrement mis à jour, les créateurs de virus font tout aussi souvent preuve d'inventivité. En particulier, les virus « oligomorphiques », « polymorphiques » et plus récemment, « métamorphiques », sont plus difficiles à détecter.

Une autre technique pour contourner ces définitions virales consiste à utiliser des packers/crypters, le fichier malicieux est chiffré et compressé et déchiffré et décompressé lors de son exécution, un peu à la manière des fichiers zip auto-extractible. Le fait que le fichier est chiffré empêche une détection par signature et permet de démultiplier les fichiers malicieux à partir d'une base unique en changeant de packers/crypters.

Liste blanche

La « liste blanche » est une technique de plus en plus utilisée pour lutter contre les logiciels malveillants. Au lieu de rechercher les logiciels connus comme malveillants, on empêche l'exécution de tout logiciel à l'exception de ceux qui sont considérés comme fiables par l'administrateur système. En adoptant cette méthode de blocage par défaut, on évite les problèmes inhérents à la mise à jour du fichier de signatures virales. De plus, elle permet d'empêcher l'exécution de logiciels indésirables.
Étant donné que les entreprises modernes possèdent de nombreuses applications considérées comme fiables, l'efficacité de cette technique dépend de la capacité de l'administrateur à établir et mettre à jour la liste blanche. Cette tâche peut être facilitée par l'utilisation d'outils d'automatisation des processus d'inventaire et de maintenance.

Comportements suspects

Une autre approche pour localiser les virus consiste à détecter les comportements suspects des programmes. Par exemple, si un programme tente d'écrire des données sur un programme exécuté, modifier/supprimer des fichiers système l'antivirus détectera ce comportement suspect et en avisera l'utilisateur qui choisira les mesures à suivre.

Contrairement à l'approche précédente, la méthode du comportement suspect permet d'identifier des virus très récents qui ne seraient pas encore connus dans le dictionnaire de l'antivirus. Toutefois, le fait que les utilisateurs soient constamment avertis de fausses alertes peuvent les rendre insensibles aux véritables menaces. Si les utilisateurs répondent « Accepter » à toutes ces alertes, l'antivirus ne leur procurera aucune protection supplémentaire. Ce problème s'est aggravé depuis 1997, puisque plusieurs programmes inoffensifs ont modifié certains fichiers exécutables sans observer ces fausses alertes. C'est pourquoi, les antivirus les plus modernes utilisent de moins en moins cette méthode.

L'intelligence artificielle des nouveaux antivirus leur permet de choisir la décision à prendre sans en avertir l'utilisateur, ce qui permet d'utiliser à nouveau cette méthode. De plus les filtres se sont considérablement améliorés et les faux positif sont moins nombreux.

Autres approches

L'*analyse heuristique* est utilisée par quelques antivirus. Par exemple, l'antivirus peut analyser le début de chaque code de toutes les nouvelles applications avant de transférer le contrôle à l'usager. Si le programme semble être un virus, alors l'usager en sera averti. Toutefois, cette méthode peut également mener à de fausses alertes. La méthode heuristique permet de détecter des variantes de virus et, en communiquant automatiquement les résultats de l'analyse à l'éditeur, celui-ci peut en vérifier la justesse et mettre à jour sa base de définitions virales.

La *méthode du bac à sable (sandbox en anglais)* consiste à émuler le système d'exploitation et à exécuter le fichier lors de cette simulation. Une fois que le programme prend fin, les logiciels analysent le résultat du bac à sable afin de détecter les changements qui pourraient contenir des virus. En raison des problèmes de performance, ce type de détection a lieu habituellement pendant le balayage sur demande. Cette méthode peut échouer puisque les virus peuvent s'avérer non déterministes et résulter de différentes actions ou même peut-être d'aucune action lorsque exécuté. Il est impossible de le détecter à partir d'une seule exécution.

Les packers[à définir] posent des problèmes d'efficacité aux détections par signature. Une autre limite était le temps assez long entre le moment où l'éditeur mettait à jour ses définitions virales, la disponibilité en ligne et la mise à jour des clients antivirus. Temps pendant lequel les utilisateurs pouvaient être vulnérables.

Pour pallier cela, les éditeurs utilisent des "Antivirus Cloud", où des bases de données gigantesques sont disponibles en ligne, permettant de recouper certaines données. En outre, ces bases de données sont utilisées en temps réel par les antivirus et permettent donc de supprimer le laps de temps entre la mise en ligne et le téléchargement des définitions virales.

Enfin, cela permet aussi d'alléger les clients antivirus où les bases locales étaient de plus en plus volumineuses face à la multiplication des menaces et l'utilisation des packers/crypters[à définir].

I.4.2.2 LISTE DES LOGICIELS ANTI-VIRUS

Logo symbolisant l'antivirus.

Les **antivirus** sont des logiciels conçus pour identifier, neutraliser et éliminer des logiciels malveillants (dont les virus informatique[1] ne sont qu'une catégorie).

Ces derniers peuvent se baser sur l'exploitation de failles de sécurité, mais il peut également s'agir de logiciels modifiant ou supprimant des fichiers, que ce soit des documents de l'utilisateur stockés sur l'ordinateur infecté, ou des fichiers nécessaires au bon fonctionnement de l'ordinateur (le plus souvent ceux du système d'exploitation).

Les sociétés et logiciels anti-virus ci-dessous sont classés par ordre alphabétique.

LISTE DES LOGICIEL ANTIVIRUS

Principales	Agnitum (en) · Avast Software · AVG Technologies · Avira · Bitdefender · Comodo (en) · Dr. Web (en) · ESET · F-Secure · Kaspersky · Intel Security (en) · Microsoft · Panda · Qihoo 360 · Sophos · Symantec · Trend Micro
Secondaires	AhnLab (en) · Cisco · Check Point · ClamWin · Fortinet · FRISK (en) · G Data Software · iolo (en) · Intego · Kingsoft (en) · Lavasoft · Malwarebytes · Quick Heal (en) · TrustPort (en) · VirusBlokAda · Webroot (en) · Zemana (en) · Zone Alarm
PC ET SERVEUR	AhnLab V3 Internet Security (en) · Avast Antivirus · AVG · Avira Internet Security · Bitdefender · ClamWin · ClamAV · Comodo Antivirus (en) · Comodo Internet Security · Dr. Web (en) · NOD32 · F-Secure · F-PROT (en) · Fortinet · G Data Software · Advanced SystemCare (en) · iolo System Shield (en) · Kaspersky Anti-Virus · Kaspersky Internet Security · KingSoft · Mac Internet Security · Malwarebytes' Anti-Malware · McAfee VirusScan · Microsoft Security Essentials · Windows Defender · Panda · 360 Safeguard (en) · Outpost Security Suite (en) · Sophos · Symantec Endpoint Protection (en) · Immunet (en) · Element Anti-Virus (en) · Norton AntiVirus · Norton Internet Security · Spyware Doctor · VirusBarrier · Trend Micro Internet Security (en) · TrustPort (en) · Vba32 AntiVirus (en) · Zone Alarm
MOBILE ET TABLETTE	AhnLab Mobile Security (en) · Avast Antivirus · AVG AntiVirus (en) · Avira Free Android Security · Bitdefender Mobile Security · CM Security · Comodo Mobile Security (en) · Dr. Web Mobile Security Suite (en) · ESET Mobile Security · F-Secure Mobile Security · G Data MobileSecurity · Lookout Mobile Security (en) · McAfee Mobile Security · FireAMP Mobile · Trend Micro Mobile Security · TrustPort Mobile Security (en) · VirusBarrier

I.4.2.3 TABLEAU COMPARATIF DES LOGICIELS ANTI-VIRUS

Logiciel	Windows	Mac OS X	GNU/ Linux	FreeBSD	Unix	Licence	Scan sur demande	Protection en temps réel	Scan d'amorçage
Ad-Aware	Oui	Non	Non	Non	Non	Propriétaire	Oui	Oui	Non
AOL Active Virus Shield	Oui	Non	Non	Non	Non	Freeware	Oui	Oui	Non
Avast! Home Edition	Oui	Oui	Oui	Non	Non	Nagware	Oui	Oui	Oui
Avast! Professional Edition	Oui	Oui	Oui	Non	Non	Propriétaire	Oui	Oui	Oui
Avetix Antivirus Free (plus actif depuis 2015)	Oui	Non	Non	Non	Non	Freeware	Oui	Oui	Non
Avetix Antivirus Pro	Oui	Non	Non	Non	Non	Propriétaire	Oui	Oui	Non
AVG Anti-Virus	Oui	Non	Oui	Oui	Non	Propriétaire	Oui	Oui	Non
AVG Anti-Virus Free	Oui	Non	Oui	Non	Non	Nagware	Oui	Oui	Non
Avira AntiVir Personal - Free Antivirus	Oui	Oui	Non	Non	Non	Nagware	Oui	Oui	Non
Avira AntiVir Premium	Oui	Oui	Oui	Oui	Oui	Propriétaire	Oui	Oui	Non
BitDefender	Oui	Oui	Oui	Oui	Non	Propriétaire	Oui	Oui	Non
BitDefender Free Edition	Oui	Non	Non	Non	Non	Nagware	Oui	Oui (avec Winpooch)	Non
BullGuard	Oui	Non	Non	Non	Non	Propriétaire	Oui	Oui	Non
CA Anti-Virus	Oui	Oui	Oui	Non	Oui	Propriétaire	Oui	Non	Non
Clam AntiVirus	Avec ClamWin	Avec ClamXav	Avec KlamAV et ClamTk	Oui	Oui	GNU GPL	Oui	Oui (en ajoutant ClamWin et Clam Sentinel sous MSWindows)	Non
ClamWin	Oui	Non	Non	Non	Non	GNU GPL	Oui	Avec Winpooch (périmé) ou Clam Sentinel	Non
Comodo AntiVirus	Oui	Non	Oui	Non	Non	Freeware	Oui	Oui	Oui

Logiciel	Windows	Mac OS X	GNU/Linux	FreeBSD	Unix	Licence	Scan sur demande	Protection en temps réel	Scan d'amorçage
Emsisoft Anti-Malware	Oui	Non	Non	Non	Non	Propriétaire	Oui	Oui	Oui
Emsisoft Anti-Malware	Oui	Non	Non	Non	Non	Freeware	Oui	Non	Non
F-Prot	Oui	Non	Oui	Oui	Oui	Propriétaire	Oui	Oui	Non
F-Secure	Oui	Non	Oui	Non	Non	Propriétaire	Oui	Oui	Non
Fortinet FortiClient End Point Security	Oui	Non	Non	Non	Non	Propriétaire	Oui	Oui	Non
G Data Software	Oui	Non	Non	Non	Non	Propriétaire	Oui	Oui	Non
Iantivirus	Oui	Oui	Non	Non	Non	Freeware	Oui	Oui	Non
Kaspersky Anti-Virus	Oui	Oui	Oui (SMB et ENT)	Oui (SMB et ENT)	Oui	Propriétaire	Oui	Oui	Non
McAfee VirusScan	Oui	Oui	Oui	Oui	Oui	Propriétaire	Oui	Oui	Non
Microsoft Security Essentials	Oui	Non	Non	Non	Non	Freeware	Oui	Oui	Non
NOD32	Oui	Oui	Oui	Oui	Oui	Propriétaire	Oui	Oui	Non
Norman	Oui	Non	Oui	Non	Non	Propriétaire	Oui	Oui	Non
Norton AntiVirus (Symantec)[2]	Oui	Oui	Non	Oui	Oui	Propriétaire	Oui	Oui	Oui
Panda Antivirus	Oui	Oui	Oui	Non	Non	Propriétaire	Oui	Oui	Non
PC Tools AntiVirus	Oui	Oui	Non	Non	Non	Propriétaire	Oui	Oui	Non
PC Tools AntiVirus Free Edition (interrompu)	Oui	Oui	Non	Non	Non	Freeware	Oui	Oui	Non
PCSafer Home edition	Oui	Non	Non	Non	Non	Freeware	Oui	Oui	Non
Qihoo 360 Internet Security	Oui	Oui	Non	Non	Non	Freeware	Oui	Oui	Oui
Qihoo 360 Total Security	Oui	Oui	Non	Non	Non	Freeware	Oui	Oui	Oui
Sophos Anti-Virus	Oui	Oui	Oui	Oui	Oui	Propriétaire	Oui	Oui	Non

Logiciel	Windows	Mac OS X	GNU/Linux	FreeBSD	Unix	Licence	Scan sur demande	Protection en temps réel	Scan d'amorçage
TrustPort Antivirus	Oui	Non	Non	Non	Non	Propriétaire	Oui	Oui	
Vba32Antivirus	Oui	Non	Oui	Oui	Non	Propriétaire	Oui	Seulement sur Windows	Non
VIPRE Antivirus + Antispyware (Sunbelt Software)	Oui	Oui	Non	Non	Non	Propriétaire	Oui	Oui	Oui
VirusBarrier Express (Intego)	Oui	Oui	Non	Non	Non	Freeware	Oui	Oui	Oui
VirusBarrier Plus (Intego)	Oui	Oui	Non	Non	Non	Propriétaire	Oui	Oui	Oui
VirusBarrier X5 (Intego)	Oui	Oui	Non	Non	Non	Propriétaire	Oui	Oui	Oui
VirusBuster	Oui	Non	Oui	Oui	Oui	Propriétaire	Oui	Oui	Non
VirusKeeper (AxBx)	Oui	Non	Non	Non	Non	Propriétaire	Oui	Oui	Oui
Windows Defender	Oui	Non	Non	Non	Non	Freeware	Oui	Oui	Oui
Zone Alarm Antivirus	Oui	Non	Non	Non	Non	Propriétaire	Oui	Oui	Oui

I.4.3 AVAST ANTIVIRUS

Avast Antivirus

Développeur Avast Software

Dernière version 18.3.2333 (3 avril 2018)

Environnement Windows, OS X, Android, Linux (serveur)

Langues	Multilingue (46 langues)[N 1]
Type	Antivirus
Politique de distribution	Freemium
Licence	Propriétaire
Site web	www.avast.com [archive]

modifier

Avast Antivirus est un logiciel antivirus développé par la société Avast Software (anciennement *Alwil Software*) située à Prague en République tchèque. C'est un logiciel propriétaire comportant une version gratuite pour une utilisation personnelle et non commerciale. En mars 2015, Avast regroupe 233 millions d'utilisateurs actifs à travers le monde répartis dans 184 pays[1][réf. insuffisante] et est disponible en 46 langues.

I.4.3.1 Historique

À la fin des années 1980[2][réf. insuffisante], deux informaticiens tchèques Eduard Kučera et Pavel Baudiš ont élaboré un programme qu'ils ont appelé « anti-virus advanced set » (AVAST). « Avast » est aussi un terme nautique hollandais (et anglais) qui signifie « stop ». Les deux programmeurs ont développé leur idée et ont fini par produire un antivirus complet du nom d'Avast.

I.4.3.2 Présentation

À la différence de certains antivirus gratuits[réf. nécessaire], ses mises à jour s'effectuent automatiquement dès la connexion à Internet, à l'instar de ses concurrents payants. Avast bénéficie également d'un moteur anti-rootkit (basé sur la technologie GMER) et anti-spyware depuis sa version 4.8.1169, ce qui en fait une suite de sécurité.

Dans le passé, lorsqu'une nouvelle infection apparaissait, Avast était souvent parmi les antivirus les plus lents à l'intégrer dans sa base de données (généralement deux ou trois semaines plus tard, ce qui laissait à l'infection le temps de gagner de nombreux ordinateurs équipés d'Avast). Néanmoins, avec son gain en popularité chez les utilisateurs familiaux, les mises à jour sont maintenant[Quand ?] quotidiennes[3][réf. insuffisante].

I.4.3.3 Version gratuite

La version gratuite est uniquement disponible pour utilisation personnelle et non commerciale. Mais ni pour les institutions, ni pour les associations.

Jusque à la version 11.1.2241 (version de 2016) l'inscription était obligatoire au bout de 12 mois afin de pouvoir continuer à bénéficier de l'antivirus depuis la version gratuite est disponible à vie sans avoir à créer de comptes.

Il existe une version gratuite pour les Très Petites Entreprises et PME en France ainsi que pour les établissements éducatifs aux États-Unis.

Identité visuelle (logo)

- 

  Logo d'Avast avant 2010.

- 

  Logo d'Avast de février à septembre 2010.

- 

  Logo d'Avast de septembre 2010 à septembre 2016.

- 

  Logo d'Avast depuis septembre 2016.

- 

  Logo d'Avast Alternative depuis septembre 2016.

Versions et dates de sorties

Avast! free, la version gratuite du logiciel a évolué depuis la v5.0 en 2007, avant nommée Avast! Home Edition. Quelques repères :

- 18.1.2326 (13/02/2018) dernière version
- 12.1.2272 (21/06/2016)
- 11.1.2253 (04/02/2016)
- 11.1.2241 (04/11/2015) dite 2016
- 10.4.2233 (18/09/2015)
- 10.0.2206 (28/10/2014) dite version 2015
- 9.0.2006 (07/10/2013) dite version 2014
- 8.0.1482 (01/03/2013) dite version 2013
- 7.0.1407 (24/02/2012)
- 6.0.1000 (24/02/2011)
- 5.0.377 (20/01/2010)

I.4.4 WINDOWS DEFENDER

I.4.4.1 PRESENTATION

Windows Defender, appelé officiellement ***Windows Defender Antivirus*** dans *Windows 10 Creators Update*, est un composant antivirus de Microsoft Windows.

Microsoft a d'abord offert le logiciel en téléchargement comme un programme antiespion gratuit pour Windows XP. Microsoft a par la suite livré le logiciel comme un antiespion avec Windows Vista et Windows 7. Finalement, le logiciel a été transformé en un programme antivirus complet remplaçant *Microsoft Security Essentials* dans Windows 8 et les versions ultérieures de Windows.

Ce logiciel était édité par *Giant Software (en)* avant son achat par Microsoft.

Développeur Microsoft

Fichier exécutable MSASCui.exe

Dernière version	4.8.10240.16384 (Windows 10) (Juillet 2015)
Environnement	Microsoft Windows
Langues	Multilingue
Type	Outil de suppression de spyware, puis programme anti-virus
Licence	Gratuiciel
Site web	Windows Defender [archive]

I.4.4.2 Fonctions de base

Avant Windows 8, *Windows Defender* protégeait ses utilisateurs contre les logiciels espions[1]. Il comprenait un certain nombre d'agents de sécurité qui surveillaient en temps réel plusieurs zones de Windows pour détecter des modifications qui auraient pu être causées par des logiciels espions.

Il permettait aussi de supprimer facilement des logiciels ActiveX installés. *Windows Defender* incluait un support intégré pour *Microsoft SpyNet (en)* qui permet aux utilisateurs de signaler à Microsoft ce qu'ils considèrent comme des logiciels espions et quelles applications et pilotes de périphériques ils permettent d'installer sur leur système. La protection contre les virus a été ajoutée dans Windows 8 ; *Windows Defender* dans Windows 8 ressemble à *Microsoft Security Essentials* et utilise les mêmes définitions de virus.

Dans Windows 10, les paramètres de *Windows Defender* sont contrôlés par l'application *Paramètres* qui peut être activée en cliquant sur le bouton *démarrer*, puis sur le bouton *Paramètres*. Depuis la mise à jour anniversaire de Windows 10, une bulle de notification annonce les résultats d'une analyse du système, même si aucun virus n'a été trouvé[2].

I.4.4.2 Histoire

Versions bêta

Windows Defender est basé sur le logiciel *GIANT AntiSpyware*, développé à l'origine par *GIANT Company Software Inc.*. Microsoft a annoncé l'acquisition de cette société le 16 décembre 2004[3,4]. Bien que le logiciel original *GIANT AntiSpyware* prenait en charge les anciennes versions de Windows, le support de la ligne de systèmes Windows 9x a été abandonné.

La première version test de *Microsoft AntiSpyware* a été publiée le 6 janvier 2005 et était essentiellement une copie réemballée de *GILANT AntiSpyware*[3]. De nouvelles versions bêta ont été publiées en 2005 et la dernière version bêta 1 a été publiée le 21 novembre 2005.

Lors de la conférence RSA Security de 2005, le concepteur de systèmes logiciels en chef et cofondateur de Microsoft, Bill Gates, a annoncé que *Windows Defender* (connu sous le nom de *Microsoft AntiSpyware* avant le 4 novembre 2005) serait mis gratuitement à la disposition de tous les utilisateurs de versions autorisées de Windows 2000, Windows XP et Windows Server 2003 pour aider à sécuriser leurs systèmes contre la menace croissante des logiciels malveillants[5].

Windows Defender (bêta 2) a été publié le 13 février 2006. Cette version porte le nouveau nom du programme et inclut une refonte significative de l'interface utilisateur. Le moteur de base a été réécrit en C++, alors que le logiciel original développé par GIANT était écrit en Visual Basic[6]. La conversion à C++ a amélioré la performance de l'application. De plus, depuis la version bêta 2, le programme fonctionne comme un service Windows, contrairement aux versions antérieures, ce qui permet à l'application de protéger l'ordinateur même si un utilisateur n'est pas connecté à l'application. La version bêta 2 nécessite également la validation *Windows Genuine Advantage*.

Cependant, *Windows Defender* (bêta 2) ne contenait pas certains des outils inclus dans *Microsoft AntiSpyware* (bêta 1). Microsoft a supprimé les outils *System Inoculation*, *Secure Shredder* et *System Explorer* inclus dans la version bêta 1 ainsi que l'outil *Tracks Eraser* qui permettait aux utilisateurs de supprimer facilement de nombreux types de fichiers temporaires liés à Internet Explorer 6, incluant les témoins (cookies), les fichiers Internet temporaires et l'historique de lecture du Lecteur Windows Media[3]. Microsoft a, par la suite, publié des versions allemande et japonaise de *Windows Defender* (bêta 2)[7,8].

Disponibilité générale du logiciel antiespion

Le 24 octobre 2006, Microsoft a lancé *Windows Defender* en version régulière (non bêta). Le logiciel prenait en charge Windows XP et Windows Server 2003. Cependant, contrairement aux versions bêta, il ne pouvait pas être exécuté sur Windows 2000[9].

Conversion vers des fonctions antivirus

Windows Defender était inclus dans Windows Vista et Windows 7 en tant que composant antiespion intégré. Dans Windows Vista et Windows 7, *Windows Defender* pouvait être remplacé par *Microsoft Security Essentials*, un produit antivirus de Microsoft qui protégeait l'ordinateur contre une gamme plus large de logiciels malveillants. Lors de l'installation, *Microsoft Security Essentials* désactivait et remplaçait *Windows Defender*[10,11,12].

Dans Windows 8, Microsoft a amélioré *Windows Defender* en en faisant un programme antivirus très similaire à *Microsoft Security Essentials* pour Windows 7 et en utilisant les mêmes mises à jour de définitions de virus que *Microsoft Security Essentials*[13].

Microsoft Security Essentials n'est pas disponible sur les versions de Windows au-delà de Windows 7. Dans Windows 8 et Windows 10, *Windows Defender* est activé par défaut. Il s'interrompt lors de l'installation d'un logiciel antivirus tiers.

À partir de Windows 10, Microsoft a commencé à transférer le contrôle de *Windows Defender* hors de son logiciel original. Initialement, sa boîte de dialogue *Paramètres* a été remplacée par une page dédiée dans l'application *Paramètres* de Windows 10. Dans *Windows 10 Creators Update, Windows Defender* est renommé *Windows Defender Antivirus* pour le distinguer du *Windows Defender Security Center*. Ce dernier est devenu l'avenue privilégiée pour l'interface avec *Windows Defender*[14]. Bien qu'il n'y ait pas de raccourci dans le menu démarrer pour accéder directement au programme *Windows Defender*, il est toujours possible d'y accéder facilement[15,16].

I.4.4.3 Fonctions avancées

Protection en temps réel

Dans les options de *Windows Defender*, l'utilisateur peut configurer des options de protection en temps réel.

Intégration avec le navigateur

L'intégration avec Internet Explorer et Microsoft Edge permet de scanner les fichiers lorsqu'ils sont téléchargés pour détecter les logiciels malveillants téléchargés par inadvertance. Bien qu'il ne s'intègre pas aux navigateurs Web autres que ceux de Microsoft, *Windows Defender* analyse les fichiers téléchargés malveillants dans le cadre de sa protection en temps réel.

I.4.4.4 Windows Defender Offline

Windows Defender Offline (anciennement connu sous le nom de *Standalone System Sweeper Beta*[17]) est un programme antivirus autonome amorçable qui fonctionne à partir d'un disque amorçable et qui est conçu pour analyser des systèmes infectés alors que leurs systèmes d'exploitation sont hors ligne[18]. Depuis la mise à jour anniversaire de Windows 10, cette fonctionnalité hors ligne est intégrée dans le programme régulier de *Windows Defender*[19].

I.4.4.5 Vulnérabilité de sécurité dans *Windows Defender*

Le 5 mai 2017, Tavis Ormandy (en), un chercheur de vulnérabilités travaillant pour Google, a découvert une vulnérabilité dans le module d'analyse JavaScript (NScript) du *Microsft*

Antimalware Engine (MsMpEngine) qui affectait *Windows Defender*, *Microsoft Security Essentials* et *System Center Endpoint Protection (en)*. Le 8 mai 2017, Microsoft avait publié un correctif pour tous les systèmes concernés. *Ars Technica* a félicité Microsoft pour sa vitesse de correction sans précédent et a déclaré qu'une catastrophe avait été évitée[20,21].

I.4.4.6 Protéger mon ordinateur avec Windows Defender

S'applique à : Windows 10

Si vous utilisez Windows 10, vous bénéficierez de la protection antivirus la plus récente grâce à Windows Defender. Lorsque vous démarrez Windows 10 pour la première fois, Windows Defender est activé et protège votre PC en recherchant des logiciels malveillants, des virus et toute menace à la sécurité. Windows Defender utilise la protection en temps réel pour analyser tous les éléments que vous téléchargez ou exécutez sur votre PC.

Windows Update télécharge automatiquement les mises à jour de Windows Defender pour vous aider à sécuriser votre PC et à le protéger contre les menaces.

Si vous disposez d'une version antérieure de Windows et que vous utilisez Microsoft Security Essentials, il est judicieux de passer à Windows Defender.

Remarque

Si vous installez une autre application antivirus, Windows Defender est automatiquement désactivé.

Planifier une analyse dans Windows Defender

Windows Defender analyse régulièrement votre PC pour le maintenir en sécurité. Si vous souhaitez planifier votre propre analyse :

1. Recherchez et ouvrez **Tâches planifiées**.
2. Dans le volet gauche, développez **Bibliothèque du Planificateur de tâches** > **Microsoft** > **Windows**, faites défiler vers le bas, puis double-cliquez sur le dossier **Windows Defender**.
3. Dans le volet en haut au centre, double-cliquez sur **Analyse planifiée de Windows Defender**.
4. Sélectionnez l'onglet **Déclencheurs**, puis sélectionnez **Nouveau**.
5. Définissez la durée et la fréquence, puis sélectionnez **OK**.

Activer ou désactiver la protection en temps réel de Windows Defender

1. Sélectionnez le bouton **Démarrer** , puis **Paramètres** > **Mise à jour et sécurité**. .

Sélectionnez Windows Defender, puis activez ou désactivez la Protection en temps réel.

I.4.5 NORTON ANTIVIRUS

Norton AntiVirus

Développeur	Symantec Corporation
Dernière version	22.10.1.10 (28 août 2017)
Environnement	Windows, Mac OS X
Type	Antivirus
Licence	Propriétaire
Site web	Symantec.com [archive]

modifier

Norton Antivirus est un logiciel antivirus pour les systèmes d'exploitation Windows et Mac édité par la société américaine Symantec.

Le logiciel est décliné en trois versions :

- *Norton Antivirus* : antivirus et antiphishing seulement ;
- *Norton Internet Security* : suite de logiciels offrant une protection plus complète ;
- *Norton 360* : suite offrant une protection optimum et divers outils de sauvegarde de données en ligne et d'optimisation du PC.

Ce logiciel est souvent offert pour une période d'essai de 30 jours à trois mois à l'achat d'un ordinateur, ce qui explique en partie sa part de marché importante chez les particuliers. Cependant, cette pratique est dénoncée par certains comme étant de la vente liée.

Historique

Développé par Peter Tippett, le premier logiciel antivirus "Vaccine" est revendu à Symantec en 1992 et renommé à cette occasion Norton Antivirus.

I.4.6 KASPERSKY ANTI-VIRUS

Kaspersky

Développeur	Kaspersky Lab
Première version	1997
Dernière version	2016 (16.0.0.614) (19 novembre 2015)
Environnements	Windows, Linux, Macintosh
Langues	Multilingue
Type	Antivirus et Anti-spywares
Licence	Logiciel propriétaire
Site web	kaspersky.com [archive]

modifier

Kaspersky Anti-Virus (KAV) (anciennement AntiViral Toolkit Pro ou AVP) est un antivirus créé par la société russe Kaspersky Lab. Cette dernière met aussi à disposition sur son site un système de recherche de virus en ligne. Le code source de Kaspersky Anti-Virus a été diffusé illégalement sur internet en 2008[1], le logiciel dans sa version 8 est écrit en C++ et en Delphi[2].

Présentation

Kaspersky est un antivirus doté de fonctions classiques de protection telles que :

- Mise en mémoire
- Analyse des courriels
- Analyse heuristique
- Analyse des fichiers compressés
- Protection contre les logiciels espions, les virus, autres vers et chevaux de Troie.
- Défense proactive, empêchant notamment l'installation de rootkits.
- La possibilité de lancer des applications en mode virtuel.

Kaspersky offre des licences de 1 à 3 ans.

Récompenses

Selon AV-Comparative, Kaspersky Antivirus se classe parmi les premiers scanneurs de virus en termes de détection en dépit du fait que l'antivirus a échoué à deux tests réalisés en 2007 et 2008 par le magazine Virus Bulletin. Par ailleurs, PC World a récompensé cet antivirus dans sa version 6 par l'octroi du prix Editor's Choice en 2007. Enfin, Ars Technica classe Kaspersky Antivirus parmi les meilleurs logiciels de sécurité destinés aux plateformes Windows.

Kaspersky a été testé par PassMark en 2008.

Limites

Plusieurs options ne se trouvent pas dans cette édition comme : Pare-feu personnel, AntiSpam, AntiBanner et le contrôle parental. Ces outils sont disponibles avec Kaspersky Internet Security.

Kaspersky comme la majorité de ses concurrents, est incompatible avec d'autres logiciels antivirus et antispyware[3].

I.4.7 DARKSPY

DarkSpy est un système de détection d'intrusion à usage individuel. Les auteurs, CardMagic (Mingyan Sun) et *wowocock*, étaient pendant la conception des étudiants-chercheurs diplômés de l'University of Science and Technology of China.

Sommaire

Description

L'essentiel de son développement s'est fait au premier semestre 2006.

Par rapport à d'autres « anti-rootkits », DarkSpy apporte

- une capacité supérieure de détection grâce, entre autres, à ses recherches à la base même du disque dur,
- et des solutions pour désactiver les rootkits en vue de leur éradication, en particulier la possibilité d'intervention sur les ruches et clés de démarrage des rootkits mises en place et cachées dans le Registre de Windows.

Depuis sa version v1.0.5, toujours qualifiée de « version de test » mais déjà stable, il peut fonctionner en complément d'autres logiciels de défense du PC contre les logiciels malveillants.

Précieux complément des moyens de défense

Partant du principe qu'aucun utilitaire actuel ne peut garantir une détection totale de toutes les variétés de rootkits présents et à venir, DarkSpy peut être

- le complément d'outils de prévention d'intrusion comme "Antihook" ou Winpooch ou des versions récentes d'utilitaires commerciaux comme Nod32, "'F-Secure"', "Kaspersky" etc.
- celui d'autres anti-rootkits comme "IceSword"' ou "Gmer".

Pour ceux qui n'ont encore qu'un antivirus et un pare-feu classiques, DarkSpy peut même constituer la base du système de protection contre les rootkits.

Environnement et installation

DarkSpy fonctionne avec les systèmes d'exploitation 32 bits de Microsoft : Windows 2000(SP4 et +) / Windows XP / Windows 2003. Il consomme très peu de ressources et convient donc à pratiquement tous les PCs.

DarkSpy ne requiert pas d'installation particulière. Son processus est lancé en mémoire depuis le répertoire du disque où le fichier ".rar" téléchargé a été décompressé.

Pour son fonctionnement, il place le fichier "DarkSpyKernel.sys" dans "C:\WINDOWS\system32" de manière non définitive. Ce "driver" s'exécute en mode noyau au sein de la mémoire. Après l'arrêt de DarkSpy et au démarrage suivant du système, ce fichier n'est pas automatiquement rechargé.

Les traces laissées dans le Registre de Windows sont peu nombreuses. On peut trouver les "ruches" sans difficulté dans les sections ...

- HKLM\SYSTEM\ControlSetxxx\Enum\Root\
- HKLM\SYSTEM\ControlSetxxx\Services.

Elles ne sont pas dangereuses pour la santé du système. Elles peuvent être laissées ou enlevées grâce à un bon outil de nettoyage du registre (celui de "IceSword" par exemple).

Précautions avant le lancement

Dans certaines conditions d'utilisation, par exemple en présence d'outils actifs de défense du noyau ou du registre de Windows, DarkSpy, ne peut pas fonctionner normalement et parfois même pas être lancé.

Des efforts particuliers ont été faits pour que DarkSpy réduise au mieux les possibilités de conflit. Actuellement, il a été testé en compagnie des suites de Kaspersky / McAfee / Norton / Jetico + Avast! et d'autres, sous différentes versions de Windows (2k/XP/2k3), mais c'est peut-être encore insuffisant. Dans un environnement système complexe, il peut encore y avoir des conflits avec d'autres logiciels. En cas de problème, vous devrez vérifier votre système et essayer de neutraliser ou même désinstaller le logiciel de sécurité qui s'oppose au fonctionnement de DarkSpy.

Sous peine de risquer un écran bleu de la mort, prenez garde de ...

- ne jamais essayer de déboguer DarkSpy avec des outils comme "Softice" / "Windbg" / "Syser debugger", car, naturellement, des fonctionnalités anti-debugg sont incluses.
- ne pas appliquer à DarkSpy un outil de rétro-ingénierie.

Modules de détection des intrusions

Process

DarkSpy établit une liste de tous les processus en activité et dispose essentiellement de trois fonctions :

- Détection des processus cachés visualisés en rouge,
- Interruption d'un processus sélectionné,
- Interruption forcée d'un processus.

Driver Module

DarkSpy procède à une recherche poussée des extensions d'application : services des processus - bibliothèques de fonctions) et pilotes cachés. Il en établit une liste complète avec indication de leur adresse en mémoire et de leur implantation sur le disque.

Les éléments cachés sont visualisés en rouge

Port

DarkSpy recherche et affiche de tous les ports ouverts, dont les ports cachés par un rootkit ayant des fonctions de cheval de Troie.

Modules de suppression des intrus

Le parasite peut être directement neutralisé s'il utilise un processus. Dans le cas d'un service, DLL ou pilote détecté dans le "Driver Module", il faudra intervenir soi-même aux endroits d'implantation.

Registry

DarkSpy inclut un très puissant module de gestion du Registre de Windows où se trouvent les ruches et clés de lancement d'une majorité des intrus, que ceux-ci utilisent un processus ou (c'est de plus en plus souvent le cas) un (ou plusieurs) service, dll et/ou driver.

Grâce à cet outil, ces clés pourront être modifiées ou supprimées afin que le rootkit ne se relance pas après un arrêt-redémarrage du système d'exploitation.

File

DarkSpy offre la possibilité de forcer la suppression des fichiers cachés et indésirables. Les fonctionnalités particulières de DarkSpy lui permettent d'outrepasser les protections de fichiers mises en place par certains logiciels.

Si les ruches et clés de lancement du rootkit ont été détectées, il est cependant conseillé de les désactiver et de redémarrer le système avant de supprimer les fichiers actifs. Une fois le rootkit neutralisé, ses fichiers peuvent être plus facilement supprimés.

I.5 PARE-FEU

I.5.1 Terminologie

Un pare-feu est parfois appelé *coupe-feu*, *garde-barrière*, *barrière de sécurité*, ou encore *firewall*. Traduction littérale : mur de feu[réf. souhaitée].

Dans un environnement Unix BSD (Berkeley Software Distribution), un pare-feu est aussi appelé *packet filter*.

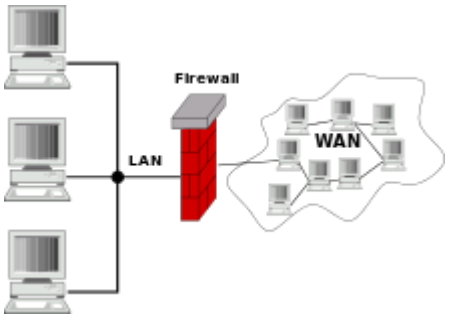
I.5.2 Origine du terme

Selon le contexte, le terme peut revêtir différentes significations :

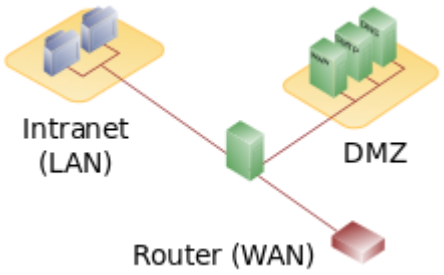
- dans le domaine de la lutte contre les incendies de forêt, il se réfère aux allées pare-feu destinées à contenir l'extension des feux de forêts ;

- au théâtre, le déclenchement d'un mécanisme « *pare-feu* » (ou « *coupe-feu* ») permet d'éviter la propagation du feu de la salle vers la scène ;
- dans le domaine de l'architecture, il fait référence aux portes coupe-feu ou à tout autre dispositif constructif destiné à contenir l'extension d'un incendie ;
- en informatique, l'usage du terme « pare-feu » est donc métaphorique. Sa dénomination, reprend au sens figuré l'intention de "brûler par un mur de feu virtuel" tout ce qui tente d'entrer avec l'intention de nuire dans une machine ou un réseau. Il établit une barrière de protection contre les intrusions et les contaminations venant de l'extérieur.

I.5.3 Fonctionnement général



Pare-feu passerelle entre LAN et WAN.



Pare-feu routeur, avec une zone DMZ.

Le pare-feu est jusqu'à ces dernières années considéré comme une des pierres angulaires de la sécurité d'un réseau informatique (il perd en importance au fur et à mesure que les communications basculent vers le HTTP sur SSL, court-circuitant tout filtrage). Il permet d'appliquer une politique d'accès aux ressources réseau (serveurs).

Il a pour principale tâche de contrôler le trafic entre différentes zones de confiance, en filtrant les flux de données qui y transitent. Généralement, les zones de confiance incluent Internet (une zone dont la confiance est nulle) et au moins un réseau interne (une zone dont la confiance est plus importante).

Le but est de fournir une connectivité contrôlée et maîtrisée entre des zones de différents niveaux de confiance, grâce à l'application de la politique de sécurité et d'un modèle de connexion basé sur le principe du moindre privilège.

Le filtrage se fait selon divers critères. Les plus courants sont :

- l'origine ou la destination des paquets (adresse IP, ports TCP ou UDP, interface réseau, etc.) ;
- les options contenues dans les données (fragmentation, validité, etc.) ;
- les données elles-mêmes (taille, correspondance à un motif, etc.) ;
- les utilisateurs pour les plus récents.

Un pare-feu fait souvent office de routeur et permet ainsi d'isoler le réseau en plusieurs zones de sécurité appelées zones démilitarisées ou DMZ. Ces zones sont séparées suivant le niveau de confiance qu'on leur porte.

Enfin, le pare-feu est également souvent situé à l'extrémité de tunnel IPsec ou SSL. L'intégration du filtrage de flux et de la gestion du tunnel est en effet nécessaire pour pouvoir à la fois protéger le trafic en confidentialité et intégrité et filtrer ce qui passe dans le tunnel. C'est le cas notamment de plusieurs produits du commerce nommés dans la liste ci-dessous.

I.5.4 Technologies utilisées

Les pare-feux récents embarquent de plus en plus de fonctionnalités, parmi lesquelles on peut citer :

- Filtrage sur adresses IP / protocole,
- Inspection *stateful*[2] et applicative,
- Intelligence artificielle pour détecter le trafic anormal,
- Filtrage applicatif :
    - HTTP (restriction des URL accessibles),
    - Courriel (Anti-pourriel),
    - Logiciel antivirus, anti-logiciel malveillant
- Traduction d'adresse réseau,
- Tunnels IPsec, PPTP, L2TP,
- Identification des connexions,
- Serveurs de protocoles de connexion (telnet, SSH), de protocoles de transfert de fichier (SCP),
- Clients de protocoles de transfert de fichier (TFTP),
- Serveur Web pour offrir une interface de configuration agréable,
- Serveur mandataire (« *proxy* » en anglais),
- Système de détection d'intrusion (« IDS » en anglais)
- Système de prévention d'intrusion (« IPS » en anglais)

I.5.5 Catégories de pare-feu

Les pare-feux sont un des plus vieux équipements de sécurité informatique et, en tant que tel, ils ont été soumis à de nombreuses évolutions. Suivant la génération du pare-feu ou son rôle précis, on peut les classer en différentes catégories.

Pare-feu sans état (*stateless firewall*)

C'est le plus vieux dispositif de filtrage réseau, introduit sur les routeurs. Il regarde chaque paquet indépendamment des autres et le compare à une liste de règles préconfigurées. Ces règles peuvent avoir des noms très différents en fonction du pare-feu :

- « ACL » pour *Access Control List* (certains pare-feux Cisco),
- politique ou *policy* (pare-feu Juniper/Netscreen),
- filtres,
- règles ou *rules*,
- etc.

La configuration de ces dispositifs est souvent complexe et l'absence de prise en compte des machines à états des protocoles réseaux ne permet pas d'obtenir une finesse du filtrage très évoluée. Ces pare-feux ont donc tendance à tomber en désuétude mais restent présents sur certains routeurs ou systèmes d'exploitation.

Article connexe : Serveur sans état.

Pare-feu à états (*stateful firewall*)

Certains protocoles dits « à états » comme TCP introduisent une notion de connexion. Les pare-feux à états vérifient la conformité des paquets à une connexion en cours. C'est-à-dire qu'ils vérifient que chaque paquet d'une connexion est bien la suite du précédent paquet et la réponse à un paquet dans l'autre sens. Ils savent aussi filtrer intelligemment les paquets ICMP qui servent à la signalisation des flux IP.

Enfin, si les ACL autorisent un paquet UDP caractérisé par un quadruplet (ip_src, port_src, ip_dst, port_dst) à passer, un tel pare-feu autorisera la réponse caractérisée par un quadruplet inversé, sans avoir à écrire une ACL inverse. Ceci est fondamental pour le bon fonctionnement de tous les protocoles fondés sur l'UDP, comme DNS par exemple. Ce mécanisme apporte en fiabilité puisqu'il est plus sélectif quant à la nature du trafic autorisé. Cependant dans le cas d'UDP, cette caractéristique peut être utilisée pour établir des connexions directes (P2P) entre deux machines (comme le fait Skype par exemple).

Article connexe : Pare-feu à états.

Pare-feu applicatif

Dernière génération de pare-feu, ils vérifient la complète conformité du paquet à un protocole attendu. Par exemple, ce type de pare-feu permet de vérifier que seul le protocole HTTP passe par le port TCP 80. Ce traitement est très gourmand en temps de calcul dès que le débit devient très important. Il est justifié par le fait que de plus en plus de protocoles réseaux utilisent un tunnel TCP afin de contourner le filtrage par ports.

Une autre raison de l'inspection applicative est l'ouverture de ports dynamique. Certains protocoles comme FTP, en mode passif, échangent entre le client et le serveur des adresses IP

ou des ports TCP/UDP. Ces protocoles sont dits « **à contenu sale** » ou « passant difficilement les pare-feux » car ils échangent au niveau applicatif (FTP) des informations du niveau IP (échange d'adresses) ou du niveau TCP (échange de ports). Ce qui transgresse le principe de la séparation des couches réseaux. Pour cette raison, les protocoles « à contenu sale » passent difficilement voire pas du tout les règles de NAT ...dynamiques, à moins qu'une inspection applicative ne soit faite sur ce protocole.

Chaque type de pare-feu sait inspecter un nombre limité d'applications. Chaque application est gérée par un module différent pour pouvoir les activer ou les désactiver. La terminologie pour le concept de module est différente pour chaque type de pare-feu : par exemple : Le protocole HTTP permet d'accéder en lecture sur un serveur par une commande GET, et en écriture par une commande PUT. Un pare-feu applicatif va être en mesure d'analyser une connexion HTTP et de n'autoriser les commandes PUT qu'à un nombre restreint de machines.

- **Pare-feu applicatif** sur Bee Ware [archive]; DenyAll [archive]
- **Firewall as a Service** (filtrage en fonction de l'origine et de la destination de chaque paquet) sur UPPERSAFE [archive]
- **Conntrack** (suivi de connexion) et **l7 Filter** (filtrage applicatif) sur Linux Netfilter
- **CBAC** sur Cisco IOS
- **Fixup** puis **inspect** sur Cisco PIX
- **ApplicationLayerGateway** sur **Proventia M**,
- **Predefined Services** sur Juniper ScreenOS
- **Stateful Inspection** sur Check Point FireWall-1
- **Deep Packet Inspection** sur Qosmos [archive]
- **Web Application Firewall** sur BinarySEC [archive]

Pare-feu identifiant

Un pare-feu réalise l'identification des connexions passant à travers le filtre IP. L'administrateur peut ainsi définir les règles de filtrage par utilisateur et non plus par adresse IP ou adresse MAC, et ainsi suivre l'activité réseau par utilisateur.

Plusieurs méthodes différentes existent qui reposent sur des associations entre IP et utilisateurs réalisées par des moyens variés. On peut par exemple citer authpf [archive] (sous OpenBSD) qui utilise ssh pour faire l'association. Une autre méthode est l'identification connexion par connexion (sans avoir cette association IP = utilisateur et donc sans compromis sur la sécurité), réalisée par exemple par la suite NuFW, qui permet d'identifier également sur des machines multi-utilisateurs.
On pourra également citer Cyberoam qui fournit un pare-feu entièrement basé sur l'identité (en réalité en réalisant des associations adresse MAC = utilisateur) ou Check Point avec l'option NAC Blade qui permet de créer des règles dynamiques basée sur l'authentification Kerberos d'un utilisateur, l'identité de son poste ainsi que son niveau de sécurité (présence d'antivirus, de patchs particuliers).

Pare-feu personnel

Les pare-feux personnels, généralement installés sur une machine de travail, agissent comme un pare-feu à états. Bien souvent, ils vérifient aussi quel programme est à l'origine des données. Le but est de lutter contre les virus informatiques et les logiciels espions.

Portail captif

Les portails captifs sont des pare-feux dont le but est d'intercepter les usagers d'un réseau de consultation afin de leur présenter une page web spéciale (par exemple : avertissement, charte d'utilisation, demande d'authentification, etc.) avant de les laisser accéder à Internet. Ils sont utilisés pour assurer la traçabilité des connexions et/ou limiter l'utilisation abusive des moyens d'accès. On les déploie essentiellement dans le cadre de réseaux de consultation Internet mutualisés filaires ou Wi-Fi.

LISTE DE PARE-FEU

Versions libres

- Linux Netfilter/Iptables, pare-feu libre des noyaux Linux 2.4, 2.6, 3.0 et suivants.
- Linux Ipchains, pare-feu libre de l'ancien noyau Linux 2.2.
- Packet Filter ou **PF**, pare-feu libre de OpenBSD, importé depuis sur les autres BSD.
- IPFilter ou **IPF**, pare-feu libre de BSD et Solaris 10 et 11.
- Ipfirewall ou **IPFW**, pare-feu libre de FreeBSD.
- iSafer, pare-feu libre pour Windows.

Distributions Linux

- SmoothWall : distribution linux packageant Netfilter et d'autres outils de sécurité pour transformer un PC en pare-feu dédié et complet.
- IPCop : distribution linux packageant Netfilter et d'autres outils de sécurité pour transformer un PC en pare-feu dédié et complet.
- Ipfire : Dérivé de IPCop, mais avec des idées très nouvelles et de design.
- Pfsense, distribution firewall open source très avancée basée sur FreeBSD et dérivée de m0n0wall qui utilise entre autres OpenBSD packet Filter.
- Zeroshell
- Amon, module du projet EOLE, distribution GNU/Linux se basant sur Ubuntu et proposant des outils d'administration.

Pare-feu identifiants

- NuFW : Un pare-feu identifiant (authentifiant). La partie serveur et les clients pour OS libres sont sous licence GPL. Le client Windows est sous licence propriétaire.

Portails captifs

- ALCASAR : Solution complète et intégrée pour contrôler et imputer les accès Internet. Sous licence GPL et gratuit.

Boîtiers pare-feu

- Arkoon Network Security
  - **Appliances UTM FAST360**, certifiées Critères communs niveau EAL3+

- Astaro Security Gateway
  - **Appliances Astaro UTM**, certifiées Critères communs niveau EAL4+ [1]

- Check Point
  - **Check Point FireWall-1**

- Cisco Systems
  - **Cisco PIX**, **Cisco ASA** et **Cisco FWSM**, boîtier pare-feu
  - **Cisco VPN3000**, boîtier pare-feu orienté RPV

- EdenWall Technologies
  - **EdenWall**, boîtier pare-feu dont la technologie est basée sur NuFW, apportant la notion d'identité des utilisateurs, en cours de qualification Critères communs niveau EAL3+[réf. nécessaire]

- Fortinet
  - **Appliances UTM FortiGate**, certifiées Critères communs niveau EAL4+, FIPS 140-2, multiple ICSA Labs Certifications dont SSL-TLS (VPN), IPSec, Network IPS, Antivirus, et Firewall

- Juniper Networks
  - **Juniper Screen OS**, boîtier pare-feu

- NetASQ
  - **Appliances UTM NetASQ**, certifiées Critères Communs niveau EAL4+(fr)) et IPv6 ready ipv6forum [archive]

- Nortel
  - Famille **Nortel VPN Router**
  - Famille **Nortel Switched Firewall**

- Stonesoft
  - **StoneGate**, pare-feu professionnel commercial centralisé pour gérer ses grappes d'appliances Pare-feu VPN et NIDS

- ZyXEL
    - **ZyWALL**, pare-feu professionnel UTM (Antivirus, filtrage applicatif, IDP, filtrage de contenu, antispam) avec Tunnel VPN IPSEC et SSL

Pare-feu personnels

- Comodo Firewall : gratuit pour usage personnel, compatible avec d'autres logiciels de protection, très complet et paramétrable, choix du tout automatique au tout manuel suivant le niveau de l'utilisateur.
- Zone Alarm : le pare-feu personnel de ZoneLabs repris par Check Point, maintenant incompatible avec tout autre logiciel de protection sauf MS Windows Defender.
- NetBarrier : le pare-feu personnel pour Mac OS X d'Intego
- Microsoft
    - Pare-feu de connexion Internet de Windows XP
    - Windows Firewall
    - Microsoft Internet Security and Acceleration Server, le pare-feu proxy-cache de Microsoft

I.6 LES PROTOCOLES DE SECURITE

PRESENTATION

Un **protocole** est une méthode standard qui permet la communication entre des processus (s'exécutant éventuellement sur différentes machines), c'est-à-dire un ensemble de règles et de procédures à respecter pour émettre et recevoir des données sur un réseau. Il en existe plusieurs selon ce que l'on attend de la communication. Certains protocoles sont par exemple spécialisés dans l'échange de fichiers (le FTP), d'autres servent à gérer simplement l'état de la transmission et des erreurs (c'est le cas du protocole ICMP).

Sur Internet, les protocoles utilisés font partie d'un ensemble de protocoles, appellés **protocoles TCP/IP**. Cette suite de protocoles contient entre autres les protocoles TCP pour le transport des données, HTTP pour le web, FTP pour le transfert de fichiers, ICMP pour le contrôle des erreurs, etc.

La plupart des protocoles de la suite TCP/IP ne sont pas sécurisés, c'est-à-dire que les données transitent en clair sur le réseau. Ainsi, des protocoles de plus haut niveau, dits « protocoles sécurisés » ont été mis au point afin d'encapsuler les messages dans des paquets de données chiffrés.

Protocole SSL

SSL (*Secure Sockets Layers*, que l'on pourrait traduire par couche de sockets sécurisée) est un procédé de sécurisation des transactions effectuées via Internet. Le standard SSL a été mis au point par Netscape, en collaboration avec Mastercard, Bank of America, MCI et Silicon Graphics. Il repose sur un procédé de cryptographie par clé publique afin de garantir la sécurité de la transmission de données sur Internet. Son principe consiste à établir un canal de communication sécurisé (chiffré) entre deux machines (un client et un serveur) après une étape d'authentification.

Le système SSL est indépendant du protocole utilisé, ce qui signifie qu'il peut aussi bien sécuriser des transactions faites sur le Web par le protocole HTTP que des connexions *via* les protocoles FTP, POP ou IMAP. En effet, SSL agit telle une couche supplémentaire permettant d'assurer la sécurité des données, située entre la couche application et la couche transport (protocole TCP par exemple).

De cette manière, SSL est transparent pour l'utilisateur (entendez par là qu'il peut ignorer qu'il utilise SSL). Par exemple un client qui recourt à un navigateur Internet pour se connecter à un site de commerce électronique sécurisé par SSL enverra des données chiffrées sans aucune manipulation nécessaire de sa part.

La totalité des navigateurs supportent aujourd'hui le protocole SSL. Sur les sites qui l'utilisent, un cadenas apparaît sur la même ligne que son adresse web.

La totalité des navigateurs supportent aujourd'hui le protocole SSL. Sur les sites qui l'utilisent, un cadenas apparaît sur la même ligne que son adresse web.

- Sous Internet Explorer :

- Sous Mozilla :

Un serveur web sécurisé par SSL possède une URL commençant par **https**://, où le « s » signifie bien évidemment *secured* (sécurisé).

Protocole SSH

Internet permet de réaliser un grand nombre d'opérations à distance, notamment l'administration de serveurs ou bien le transfert de fichiers. Le protocole telnet et les *r-commandes BSD* (*rsh*, *rlogin* et *rexec*) permettant d'effectuer ces tâches distantes possèdent l'inconvénient majeur de faire circuler en clair sur le réseau les informations échangées, notamment l'identifiant (*login*) et le mot de passe pour l'accès à la machine distante. Ainsi, un pirate situé sur un réseau entre l'utilisateur et la machine distante a la possibilité d'écouter le trafic, c'est-à-dire d'utiliser un outil appelé *sniffer* capable de capturer les trames circulant sur le réseau et ainsi d'obtenir l'identifiant et le mot de passe d'accès à la machine distante.

Même si les informations échangées ne possèdent pas un grand niveau de sécurité, le pirate obtient un accès à un compte sur la machine distante et peut éventuellement étendre ses privilèges sur la machine afin d'obtenir un accès administrateur (*root*).

Étant donné qu'il est impossible de maîtriser l'ensemble des infrastructures physiques situées entre l'utilisateur et la machine distante (Internet étant par définition un réseau ouvert), la seule solution est de recourir à une sécurité au niveau logique (au niveau des données).

Le protocole **SSH** (*Secure Shell*) répond à cette problématique en permettant à des utilisateurs (ou bien à des services TCP/IP) d'accéder à une machine à travers une communication chiffrée (appelée tunnel).

Protocole Secure HTTP

S-HTTP (*Secure HTTP*, ce qui signifie Protocole HTTP sécurisé) est un procédé de sécurisation des transactions HTTP reposant sur une amélioration du protocole HTTP mise au point en 1994 par l'EIT (*Enterprise Integration Technologies*). Il permet de fournir une sécurisation des échanges lors de transactions de commerce électronique en cryptant les messages afin de garantir aux clients la confidentialité de leur numéro de carte bancaire ou de toute autre information personnelle. Une implémentation de S-HTTP a été développée par la société Terisa Systems afin d'inclure une sécurisation au niveau des serveurs web et des navigateurs.

❑ **Complémentarité avec SSL**

Alors que SSL et S-HTTP étaient concurrents, un grand nombre de personnes a réalisé que les deux protocoles de sécurisation étaient complémentaires, étant donné qu'ils ne travaillaient pas au même niveau. De cette façon, SSL permet de sécuriser la connexion Internet tandis que S-HTTP permet de fournir des échanges HTTP sécurisés.

C'est pourquoi, la compagnie Terisa Systems, spécialisée dans la sécurisation des réseaux, formée par RSA Data Security et l'EIT, a mis au point un kit de développement permettant de développer des serveurs web implémentant SSL et S-HTTP (*SecureWeb Server Toolkit*), ainsi que des clients web supportant ces protocoles (*SecureWeb Client Toolkit*).

Protocole SET

SET (*Secure Electronic Transaction*) est un protocole de sécurisation des transactions électroniques mis au point par Visa et MasterCard, et s'appuyant sur le standard SSL.

SET est basé sur l'utilisation d'une signature électronique au niveau de l'acheteur et une transaction mettant en jeu non seulement l'acheteur et le vendeur, mais aussi leurs banques respectives.

Lors d'une transaction sécurisée avec SET, les données sont envoyées par le client au serveur du vendeur, mais ce dernier ne récupère que la commande. En effet, le numéro de carte bleue est envoyé directement à la banque du commerçant, qui va être en mesure de lire les coordonnées bancaires de l'acheteur, et donc de contacter sa banque afin de les vérifier en temps réel.

Ce type de méthode nécessite une signature électronique au niveau de l'utilisateur de la carte afin de certifier qu'il s'agit bien du possesseur de cette carte.

Protocole S/MIME

S/MIME (Secure MIME, soit *Secure Multipurpose Mail Extension*, que l'on pourrait traduire par extensions du courrier électronique à buts multiples et sécurisées) est un procédé de sécurisation des échanges par courrier électronique permettant de garantir la confidentialité et la non-répudiation des messages électroniques.

S/MIME est basé sur le standard MIME, dont le but est de permettre d'inclure dans les messages électroniques des fichiers attachés autres que des fichiers texte (ASCII).

C'est ainsi grâce au standard MIME qu'il est possible d'ajouter des pièces jointes de tous types aux courriers électroniques.

Protocole DNSsec

Les serveurs DNS sont des ordinateurs qui structurent le Web mondial. Les acteurs de ce secteur ont donc décidé de sécuriser les transmissions de ces serveurs entre eux. C'est ce que tente de faire le **protocole sécurisé DNSsec** (*Domain Name System Security Extensions*). DNSsec constitue une des extensions du protocole DNS et est censé assurer l'authentification et l'intégrité des enregistrements du DNS. Son fonctionnement est basé sur des vérifications de signatures numériques et d'échange de données cryptées. DNSsec permet ainsi de constituer une chaîne d'identification sécurisée. L'utilisateur pourrait donc mieux identifier si tel ou tel sous-domaine d'un site web est bien celui auquel il veut s'adresser. DNSsec pourrait être mis en place de façon systématique pour contrer les failles du protocole DNS et le phénomène de *phishing*.

CHAP.II LES PORTS LOGIQUES, LES VULNERABILITES INFORMATIQUES ET LES JOURNAUX D'EVENEMENTS LOGS

II.1 LES PORTS LOGIQUES

Dans la suite des protocoles Internet et correspondant à la couche de transport du modèle OSI, la notion de **port** logiciel permet, sur un ordinateur donné, de distinguer différents interlocuteurs. Ces interlocuteurs sont des programmes informatiques qui, selon les cas, écoutent ou émettent des informations sur ces ports. Un port est distingué par son numéro.

Le terme **port** est aussi parfois utilisé pour désigner les **interfaces de connexion**, un concept sensiblement différent.

○

Origine du mot

Port, en informatique, est une traduction erronée de l'anglais ***port*** (en) [archive] ; l'étymologie du mot au sens informatique est le latin *porta (→ porte)*, et non *portus (→ port)*[1].

Explication métaphorique

Pour simplifier, on peut considérer les ports comme des *portes donnant accès au système d'exploitation* : (Microsoft Windows, Mac OS, GNU/Linux, Solaris…). Pour fonctionner, un programme (par exemple un jeu à accélération 3D/2D, ou un logiciel de retouche photo) ouvre des portes pour entrer dans le système d'exploitation, mais lorsque l'on quitte le programme, la porte n'a plus besoin d'être ouverte.

Utilité

Grâce à cette abstraction, on peut exécuter plusieurs logiciels serveurs sur une même machine, et même simultanément des logiciels clients et des serveurs, ce qui est fréquent sur les systèmes d'exploitation multitâches et multiutilisateurs.

Attribution des ports

Un numéro de port est codé sur 16 bits, ce qui fait qu'il existe un maximum de soit 65 536 ports distincts par machine. Ces ports sont classés en 3 catégories en fonction de leur numéro:

- les numéros de port de 0 à 1 023 correspondent aux ports "bien-connus" (well-known ports), utilisés pour les services réseaux les plus courants.
- les numéros de ports de 1 024 à 49 151 correspondent aux ports enregistrés (registered ports), assignés par l'IANA
- les numéros de ports de 49 152 à 65 535 correspondent aux ports dynamiques, utilisables pour tout type de requêtes TCP ou UDP autres que celle citées précédemment.

Lorsqu'un logiciel client veut dialoguer avec un logiciel serveur, aussi appelé service, il a besoin de connaître le port écouté par ce dernier. Les ports utilisés par les services devant être connus par les clients, les principaux types de services utilisent des ports qui sont dits réservés. Par convention, ce sont tous ceux

compris entre 0 et 1 023[2] inclus et leur utilisation par un logiciel serveur nécessite souvent que celui-ci s'exécute avec des droits d'accès particuliers. Les services utilisant ces ports sont appelés les *services bien connus*("Well-Known Services").

Le fichier *services* indique la liste de ces services dits well-known. Sous UNIX, ce fichier est directement dans /etc ; sous Windows, ce fichier est par défaut dans C:\Windows\System32\drivers\etc. Les services les plus utilisés sont :

- 9, pour le WoL, Wake-on-LAN, c'est-à-dire le démarrage à distance par un câble réseau ethernet. Wake-on-LAN
- 20/21, pour l'échange de fichiers via FTP
- 22, pour l'accès à un shell sécurisé Secure SHell, également utilisé pour l'échange de fichiers sécurisés SFTP
- 23, pour le port telnet
- 25, pour l'envoi d'un courrier électronique via un serveur dédié SMTP
- 53, pour la résolution de noms de domaine en adresses IP : DNS
- 67/68, pour DHCP et bootpc
- 80, pour la consultation d'un serveur HTTP par le biais d'un navigateur web
- 110, pour la récupération de son courrier électronique via POP
- 123, pour la synchronisation de l'horloge : Network Time Protocol (NTP)
- 143, pour la récupération de son courrier électronique via IMAP
- 389, pour la connexion à un LDAP
- 443, pour les connexions HTTP utilisant une surcouche de sécurité de type SSL : HTTPS
- 465, pour l'envoi d'un courrier électronique via un serveur dédié utilisant une surcouche de sécurité de type SSL : SMTPS
- 500, port utilisé pour le canal d'échange de clés IPsec
- 554, port utilisé pour accepter les connexions client RTSP entrantes et pour fournir des paquets de données aux clients qui diffusent en utilisant RTSPT.
- 636, pour l'utilisation d'une connexion à un LDAP sécurisé par une couche SSL/TLS
- 1352, pour le protocole Lotus Notes Domino
- 1433, serveur de base de données MS SQL
- 1521, serveur de base de données Oracle Database
- 1723, pour l'utilisation du protocole de VPN PPTP
- 3306, serveur de base de données MySQL
- 3389, pour la prise de contrôle à distance RDP
- 5432, serveur de base de données PostgreSQL
- 6667, pour la connexion aux serveurs IRC
- 25565, port par défaut des serveurs Minecraft[3,4,5]

III.2 LES VULNERABILITES INFORMATIQUES

III.2.1 PRESENTATION

Dans le domaine de la sécurité informatique, une **vulnérabilité** ou **faille** est une faiblesse dans un système informatique permettant à un attaquant de porter atteinte à l'intégrité de ce système, c'est-à-dire à son fonctionnement normal, à la confidentialité ou à l'intégrité des données qu'il contient.

Ces vulnérabilités sont la conséquence de faiblesses dans la conception, la mise en œuvre ou l'utilisation d'un composant matériel ou logiciel du système, mais il s'agit souvent d'anomalies logicielles liées à des erreurs de programmation ou à de mauvaises pratiques. Ces dysfonctionnements logiciels sont en général corrigés à mesure de leurs découvertes, mais l'utilisateur reste exposé à une éventuelle exploitation tant que le correctif (temporaire ou définitif) n'est pas publié et installé. C'est pourquoi il est important de maintenir les logiciels à jour avec les correctifs fournis par les éditeurs de logiciels. La procédure d'exploitation d'une vulnérabilité logicielle est appelée exploit.

II.2.2 Causes

Les vulnérabilités informatiques proviennent souvent de la négligence ou de l'inexpérience d'un programmeur. Il peut y avoir d'autres causes liées au contexte comme l'évolution des technologies, notamment en cryptographie. Une vulnérabilité permet généralement à l'attaquant de duper l'application, par exemple en outrepassant les vérifications de contrôle d'accès ou en exécutant des commandes sur le système hébergeant l'application.

Quelques vulnérabilités surviennent lorsque l'entrée d'un utilisateur n'est pas contrôlée, permettant l'exécution de commandes ou de requêtes SQL (connues sous le nom d'injection SQL). D'autres proviennent d'erreurs d'un programmeur lors de la vérification des buffers de données (qui peuvent alors être dépassés), causant ainsi une corruption de la pile mémoire (et ainsi permettre l'exécution de code fourni par l'attaquant).

II.2.3 Publication d'une vulnérabilité

Méthode de publication

La méthode de publication des vulnérabilités est un sujet qui fait débat au sein de la communauté de la sécurité des systèmes d'information. Certains affirment qu'il est nécessaire de publier immédiatement toutes les informations à propos d'une vulnérabilité dès qu'elle a été découverte *(full disclosure)*. D'autres prétendent qu'il est préférable de limiter en premier lieu

la publication uniquement aux utilisateurs qui en ont un besoin important (divulgation responsable, voire coordonnée), puis après un certain délai, de publier en détail, s'il y a besoin.

Ces délais peuvent permettre de laisser le temps aux développeurs de corriger la vulnérabilité et à ces utilisateurs d'appliquer les patchs de sécurité nécessaires, mais peuvent aussi accroître les risques pour ceux qui n'ont pas ces informations. Les éditeurs de logiciels appellent cette méthode de publication la divulgation responsable et encouragent les chercheurs en sécurité à l'utiliser. En théorie, ces délais permettent aux éditeurs de publier les correctifs nécessaires pour protéger leurs logiciels et leurs utilisateurs, mais en pratique, cela ne les contraint pas à corriger les vulnérabilités. Il a ainsi pu arriver que certaines vulnérabilités soient restées non corrigées pendant des mois voire des années, tant qu'aucun exploit n'a été publié. Devant cette situation, certains ont décidé de laisser un délai - considéré raisonnable - aux éditeurs pour corriger les vulnérabilités avant de les divulguer. Ainsi la société TippingPoint laisse un délai de 6 mois avant de divulguer les détails d'une vulnérabilité[1].

Date et source de publication

La date de publication est la première date à laquelle une vulnérabilité est décrite sur un média. L'information révélée suit les conditions suivantes :

– l'information est disponible librement et publiquement ;
– l'information sur la vulnérabilité est publiée par une source indépendante et de confiance ;
– la vulnérabilité a fait l'objet d'analyse par des experts, notamment sur l'estimation du risque de la révélation.

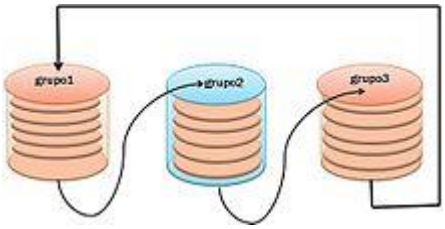
D'un point de vue sécurité, seule une publication libre d'accès et complète peut assurer que toutes les parties intéressées obtiennent l'information appropriée. La sécurité par l'obscurité est un concept qui n'a jamais fonctionné.

La source de la publication doit être indépendante d'un éditeur, d'un vendeur, ou d'un gouvernement. Elle doit être impartiale pour permettre une diffusion de l'information juste et critique. Un média est considéré comme « de confiance » lorsqu'il est une source de la sécurité des systèmes d'information largement acceptée dans l'industrie (par exemple : CERT, CESTI, Securityfocus, Secunia).

L'analyse et l'estimation du risque assurent la qualité de l'information révélée. Une unique discussion sur une faille potentielle dans une liste de diffusion ou une vague information d'un vendeur ne permettent donc pas de qualifier une vulnérabilité. L'analyse doit inclure assez de détails pour permettre à un utilisateur concerné d'évaluer lui-même son risque individuel, ou de prendre une mesure immédiate pour se protéger.

II.3 JOURNAUX DES EVENEMENTS (FICHIERS LOGS) ET SYSLOG

II.3.1 JOURNAUX DES EVENEMENTS (FICHIERS LOGS)



Structure de journal informatique "circulaire"

En informatique, le concept d'**historique des événements** ou de **journalisation** désigne l'enregistrement séquentiel dans un fichier ou une base de données de tous les événements affectant un processus particulier (application, activité d'un réseau informatique…). Le **journal** (en anglais *log file* ou plus simplement *log*), désigne alors le fichier contenant ces enregistrements. Généralement datés et classés par ordre chronologique, ces derniers permettent d'analyser pas à pas l'activité interne du processus et ses interactions avec son environnement.

Un fichier texte peut être la structure sous-jacente d'un fichier journal

En français, le *journal* est le livre où sont inscrits des événements. Il s'applique initialement dans le domaine des organisations (journal de bord), puis à partir du XIX^e^ siècle dans le domaine littéraire (journal intime) et dans le domaine de la presse d'information (journal d'information)[2]. Le journal est aussi le terme utilisé en comptabilité d'entreprise et a diffusé dans les système informatiques bancaires pour la traçabilité des transactions et des erreurs, puis dans le domaine des télécommunications, enfin en informatique générale[2]. Le terme se différencie des registres et main courante par son séquencement temporel[2].

En anglais et en argot franglais, le terme *log file* est la traduction de *journal* ou de *main-courante*, tandis que le terme *inscription* est traduit en anglais par *log*.

Fonctionnalités offertes

L'enregistrement d'un historique permet de mettre en œuvre des fonctionnalités telles que les « derniers fichiers ouverts », les « dernières commandes tapées » ou les « dernières pages web consultées ».

Applications

La journalisation est une technique importante, utilisé entre autres en sécurité informatique et dans les systèmes de comptabilité et paiement.

Dans le cadre de la sécurité, les événements enregistrés seront les accès au système, les modifications de fichiers, etc. On consacre typiquement une ligne par événement, en commençant par le moment exact (date, heure, minute, seconde) où il a eu lieu.

La journalisation permet d'effectuer des analyses diverses, généralement statistiques ; de faire des hypothèses sur les dysfonctionnements ou les pertes de performance d'un système.

L'accès aux journaux peut contrevenir à certaines exigences de confidentialité, voire de sécurité.

La journalisation permet aussi d'enregistrer les événements dans deux places différentes.

Les enregistrements d'événements peuvent également avoir une importance légale. Par exemple un fournisseur d'accès à Internet est tenu de fournir un historique des connexions de ses clients (Loi du 15 novembre 2001 "LSQ", Loi du 21 juin 2004 "LCEN").

Journalisation applicative

La journalisation applicative désigne l'enregistrement chronologique des opérations de la logique métier pendant le fonctionnement de l'application. Un journal applicatif est lui-même une exigence du métier. Il est donc défini comme une fonctionnalité faisant partie de la logique applicative. Par conséquent, il ne devrait pas être arrêté pendant le fonctionnement de l'application.

Journalisation système

La journalisation système désigne l'enregistrement chronologique des événements survenant au niveau des composants du système. Le niveau de cette journalisation peut être paramétré, afin de filtrer les différents événements selon leur catégorie de gravité. Les catégories généralement utilisées sont, par ordre croissant de gravité : information, débogage, avertissement, erreur.

Pour exemple, les systèmes Unix mettent en œuvre cette journalisation système à l'aide du protocole Syslog.

II.3.2 Syslog

Syslog	
Fonction	Transmission de journaux
Port	UDP 514
RFC	RFC 3164[1] RFC 3195[2] RFC 5424[3] RFC 5426[4]
modifier	

Syslog est un protocole définissant un service de journaux d'événements d'un système informatique. C'est aussi le nom du format qui permet ces échanges.

Historique

Syslog a été développé dans les années 1980 par Eric Allman dans le cadre du projet Sendmail[5], et n'était initialement prévue que pour Sendmail. Il s'est avéré si utile que d'autres applications ont commencé à l'utiliser. Syslog est depuis devenu la solution de journalisation standard sur les systèmes Unix et Linux[6], il y a également une variété d'implémentations syslog sur d'autres systèmes d'exploitation (Windows notamment[7]) et est généralement trouvé dans les périphériques réseau tels que les commutateurs ou routeurs.

Le protocole Syslog

Présentation générale

En tant que protocole, Syslog se compose d'une partie cliente et d'une partie serveur. La partie cliente émet les informations sur le réseau, via le port UDP 514. Les serveurs collectent l'information et se chargent de créer les journaux.

L'intérêt de Syslog est donc de centraliser les journaux d'événements, permettant de repérer plus rapidement et efficacement les défaillances d'ordinateurs présents sur un réseau.

Il existe aussi un logiciel appelé *Syslog*, qui est responsable de la prise en charge des fichiers de journalisation du système. Ceci inclut aussi le démon klogd, responsable des messages émis par le noyau Linux.

Positionnement système[modifier | modifier le code]

Le protocole syslog utilise un socket afin de transmettre ses messages. Suivant les systèmes, celui-ci est différent:

Plate-forme	**Méthode**
Linux	Un SOCK_STREAM unix nommé /dev/log; certaines distributions utilisent SOCK_DGRAM
BSD	Un SOCK_DGRAM unix appelé /var/run/log.
Solaris (2.5 et inférieurs)	Un flux SVR4 appelé /dev/log.
Solaris (2.6 et supérieurs)	En plus du flux habituel, une porte multithreaded appelée /etc/.syslog_door est utilisée.
HP-UX 11 et supérieur	HP-UX utilise le Tube Unix nommé /dev/log de taille 2048 bytes
AIX 5.2 and 5.3	Un SOCK_STREAM ou un SOCK_DGRAM unix appelé /dev/log.

Une problématique nait de ce choix architectural, l'utilisation d'un point d'entrée unique crée des saturations système qui ont incité nombre de logiciels à utiliser leur propre système d'enregistrement.

CHAP.III : INTRODUCTION AUX OUTILS DE SECURITE INFORMATIQUE

III.1 LES SCANNEURS DES PORTS ET L'OUTIL DE SECURITE INFORMATIQUE NMAP

III.1.1 NOTIONS SUR LE BALAYAGE DE PORT

En informatique, le **balayage de ports** (*port scanning* en anglais) est une technique servant à rechercher les ports ouverts sur un serveur de réseau. Cette technique est utilisée par les administrateurs des systèmes informatiques pour contrôler la sécurité des serveurs de leurs réseaux. La même technique est aussi utilisée par les pirates informatiques pour tenter de trouver des failles dans des systèmes informatiques. Un balayage de ports (*port scan* ou *portscan* en anglais) effectué sur un système tiers est généralement considéré comme une tentative d'intrusion, car un balayage de ports sert souvent à préparer une intrusion.

Le balayage de ports est une des activités considérées comme suspectes par un système de détection d'intrusion. Un système de détection d'intrusion peut être réglé à différents niveaux de sensibilité. Un niveau de sensibilité élevé génèrera plus de fausses alertes, un niveau de sensibilité bas risque de laisser passer les balayages effectués par des systèmes sophistiqués comme Nmap qui disposent de diverses options pour camoufler leurs balayages.

Pour tromper la vigilance des systèmes de détection et des pare-feu, les balayages peuvent se faire dans un ordre aléatoire, avec une vitesse excessivement lente (par exemple sur plusieurs jours), ou à partir de plusieurs adresses IP.

Les balayages de ports se font habituellement sur le protocole TCP ; néanmoins, certains logiciels permettent aussi d'effectuer des balayages UDP. Cette dernière fonctionnalité est beaucoup moins fiable, UDP étant orienté sans connexion, le service ne répondra que si la requête correspond à un modèle précis variant selon le logiciel serveur utilisé.

III.3.2 Techniques de balayage des ports

1. TCP

Un balayage de ports vise typiquement le protocole TCP, car c'est celui qui est utilisé par la majorité des applications. L'objectif du balayage est de savoir si un logiciel est en écoute sur un numéro de port donné. Si un logiciel écoute, on dit que le port est *ouvert*, sinon on dit qu'il est *fermé*. Le balayage d'un port se passe en deux étapes :

1. l'envoi d'un paquet sur le port testé ;
2. l'analyse de la réponse.

Il existe de nombreuses variantes pour le paquet émis. Il y a le paquet valide selon la norme TCP, le paquet « TCP SYN », et les paquets invalides. L'utilisation des paquets invalides vise à tromper les systèmes de détection d'intrusion. La liste des paquets invalides utilisés est :

- ACK ;
- FIN ;
- Maimon[1] (FIN/ACK) ;
- NULL (aucun) ;
- Xmas[2] (tous) ;
- *Window* (ACK).

Le serveur peut répondre de différentes manières :

- ouverture de connexion acceptée : envoi d'un paquet TCP SYN/ACK ;
- fermeture de la connexion : envoi d'un paquet TCP RST ;
- absence de réponse : on dit que le paquet est *droppé*.

La réponse *ouverture de connexion acceptée* indique clairement que le port est ouvert. La réponse *fermeture de la connexion* indique que le port est fermé. L'*absence de réponse* est souvent due à un pare-feu qui vise à contrer le balayage de ports. Le pare-feu peut détecter un trafic anormal et décider d'ignorer pendant un certain temps tous les paquets provenant de la machine générant le trafic anormal. En absence de réponse, on ne peut donc pas savoir avec certitude si le port est ouvert ou fermé.

La technique *Window* envoie un paquet TCP ACK et observe la taille de la fenêtre TCP du paquet de réponse (TCP RST). Si le port est fermé, la taille de la fenêtre de la réponse est nulle.

La technique *Mainon* est utilisée sur les systèmes BSD. Uriel Maimon a constaté que ces systèmes ignorent un paquet TCP FIN/ACK (invalide) si le port est ouvert au lieu d'envoyer la réponse TCP RST.

2. Autres techniques pour TCP

Une autre technique consiste à passer par un serveur FTP. On utilise la fonctionnalité de serveur mandataire des serveurs FTP pour balayer les ports.

Enfin, la technique Idle scanning utilise l'identifiant de fragmentation du protocole IP. Un système de détection d'intrusion pense que l'analyse provient d'un ordinateur zombi. Consultez l'article (en) Idle Scanning and Related IPID Games [archive] pour plus de renseignements.

3. UDP et IP

Pour le protocole UDP, on envoie un paquet UDP vide (de longueur nulle). Si le port est fermé, un message ICMP de type 3 (*destinataire inaccessible*) et code 3 est envoyé.

Il est également possible de lister les protocoles IP pris en charge par un hôte. On appelle cette technique *IP protocol*

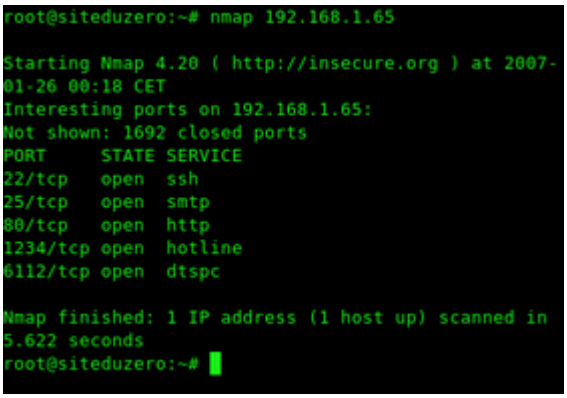
4. **Version du scanner des ports**

On peut détecter le système d'exploitation et sa version par la prise d'empreinte de la pile TCP/IP. Un logiciel tel que Nmap permet également de détecter le nom du logiciel écoutant sur un port, voire sa version.

III.3.2 LE NMAP (Scanneurs des ports)

III.3.2.1 PRESENTATION

Nmap



Développeur	Fyodor
Première version	1er septembre 1997[1]
Dernière version	7.70 (20 mars 2018)[2]

Écrit en	C++, Python, C, Lua et Java
Environnements	Multiplate-forme
Type	Sécurité informatique
Licence	GNU GPL
Site web	nmap.org [archive]

modifier

Nmap est un scanneur de ports libre créé par Fyodor et distribué par Insecure.org. Il est conçu pour détecter les ports ouverts, identifier les services hébergés et obtenir des informations sur le système d'exploitation d'un ordinateur distant. Ce logiciel est devenu une référence pour les administrateurs réseaux car l'audit des résultats de Nmap fournit des indications sur la sécurité d'un réseau. Il est disponible sous Windows, Mac OS X, Linux, BSD et Solaris.

Le code source de Nmap est disponible sous la licence GNU GPL.

III.3.2.2 Les bases du scan de ports

Même si le nombre de fonctionnalités de Nmap a considérablement augmenté au fil des ans, il reste un scanner de ports efficace, et cela reste sa fonction principale. La commande de base **nmap** <***target***> scanne plus de 1 660 ports TCP de l'hôte <*target*>. Alors que de nombreux autres scanners de ports ont partitionné les états des ports en ouverts ou fermés, Nmap a une granularité bien plus fine. Il divise les ports selon six états: ouvert (open), fermé (closed), filtré (filtered), non-filtré (unfiltered), ouvert|filtré (open|filtered), et fermé|filtré (closed|filtered).

Ces états ne font pas partie des propriétés intrinsèques des ports eux-mêmes, mais décrivent comment Nmap les perçoit. Par exemple, un scan Nmap depuis le même réseau que la cible pourrait voir le port 135/tcp comme ouvert alors qu'un scan au même instant avec les mêmes options au travers d'Internet pourrait voir ce même port comme filtré.

III.3.2.3 Les six états de port reconnus par Nmap

ouvert (open)

> Une application accepte des connexions TCP ou des paquets UDP sur ce port. Trouver de tels ports est souvent le but principal du scan de ports. Les gens soucieux de la sécurité savent pertinemment que chaque port ouvert est un boulevard pour une attaque. Les attaquants et les pen-testers veulent exploiter ces ports ouverts, tandis que les administrateurs essaient de les fermer ou de les protéger avec des pare-feux sans gêner leurs utilisateurs légitimes. Les ports ouverts sont également intéressants pour

des scans autres que ceux orientés vers la sécurité car ils indiquent les services disponibles sur le réseau.

fermé (closed)

Un port fermé est accessible (il reçoit et répond aux paquets émis par Nmap), mais il n'y a pas d'application en écoute. Ceci peut s'avérer utile pour montrer qu'un hôte est actif (découverte d'hôtes ou scan ping), ou pour la détection de l'OS. Comme un port fermé est accessible, il peut être intéressant de le scanner de nouveau plus tard au cas où il s'ouvrirait. Les administrateurs pourraient désirer bloquer de tels ports avec un pare-feu, mais ils apparaîtraient alors dans l'état filtré décrit dans la section suivante.

filtré (filtered)

Nmap ne peut pas toujours déterminer si un port est ouvert car les dispositifs de filtrage des paquets empêchent les paquets de tests (probes) d'atteindre leur port cible. Le dispositif de filtrage peut être un pare-feu dédié, des règles de routeurs filtrants ou un pare-feu logiciel. Ces ports ennuient les attaquants car ils ne fournissent que très peu d'informations. Quelques fois ils répondent avec un message d'erreur ICMP de type 3 code 13 (« destination unreachable: communication administratively prohibited »), mais les dispositifs de filtrage qui rejettent les paquets sans rien répondre sont bien plus courants. Ceci oblige Nmap à essayer plusieurs fois au cas où ces paquets de tests seraient rejetés à cause d'une surcharge du réseau et pas du filtrage. Ceci ralenti terriblement les choses.

non-filtré (unfiltered)

L'état non-filtré signifie qu'un port est accessible, mais que Nmap est incapable de déterminer s'il est ouvert ou fermé. Seul le scan ACK, qui est utilisé pour déterminer les règles des pare-feux, catégorise les ports dans cet état. Scanner des ports non-filtrés avec un autre type de scan, comme le scan Windows, SYN ou FIN peut aider à savoir si un port est ouvert ou pas.

ouvert|filtré (open|filtered)

Nmap met dans cet état les ports dont il est incapable de déterminer l'état entre ouvert et filtré. Ceci arrive pour les types de scans où les ports ouverts ne renvoient pas de réponse. L'absence de réponse peut aussi signifier qu'un dispositif de filtrage des paquets a rejeté le test ou les réponses attendues. Ainsi, Nmap ne peut s'assurer ni que le port est ouvert, ni qu'il est filtré. Les scans UDP, protocole IP, FIN, Null et Xmas catégorisent les ports ainsi.

fermé|filtré (closed|filtered)

Cet état est utilisé quand Nmap est incapable de déterminer si un port est fermé ou filtré. Cet état est seulement utilisé par le scan Idle basé sur les identifiants de paquets IP.

III.2 SCANNEUR DE VULNERABILITES ET L'OUTIL DE SECURITE INFORMATIQUE NESSUS

III.2.1 NOTIONS SUR LES SCANNEURS DE VULNERABILITES

En sécurité informatique, un **scanner (ou scanneur) de vulnérabilités** est un programme conçu pour identifier des vulnérabilités dans une application, un système d'exploitation, ou un réseau.

1. Utilisation

Les scanneurs de vulnérabilités peuvent être utilisés dans des objectifs licites ou illicites :

- objectifs licites : les experts en sécurité informatique ou les entreprises utilisent les scanneurs de vulnérabilités pour trouver les failles de sécurité des systèmes informatiques et des systèmes de communications de leurs entreprises dans le but de les corriger avant que les pirates informatiques ne les exploitent ;
- objectifs illicites : les pirates informatiques utilisent les mêmes équipements pour trouver les failles dans les systèmes des entreprises pour les exploiter à leur avantage.

Cet outil peut être une brique intégrée d'une solution de sécurité plus large: un SIEM ou un SOC par exemple.

2. Principes de fonctionnement

2.1 Généralités

Les scanners de vulnérabilités se présentent sous plusieurs formes: logiciel à installer sur son système, machine virtuelle pré-configurée (*virtual appliance*) ou encore en SaaS dans le *cloud*.

Un scanner de vulnérabilités se "lance" sur une ou plusieurs cibles, dans un réseau interne ou sur Internet. Ces cibles (URL, adresse IP, sous-réseau) sont renseignées par l'utilisateur lorsqu'il désire mener son scan. La plupart des outils suivent le schéma de scan suivant:

- détection des cibles actives (attente d'une réponse ICMP, ARP, TCP, etc. pour déterminer si la cible répondra au scanner)
- détection des ports TCP et UDP accessibles sur la cible (scan de ports)
- détection des services actifs (SSH, HTTP, etc.) sur chacun de ces ports et de leurs versions (phase de "*fingerprint*")

- éventuellement: utilisation d'un compte fourni pour se connecter sur la machine et lister les programmes non visibles depuis le réseau (navigateur, liseuse, suite bureautique, etc.)
- éventuellement: reconnaissance des applications Web accessibles et construction de l'arbre de chaque site Web (phase dite de "*crawling*")
- sélection des modules de sécurité à lancer sur la cible selon les services précédemment reconnus
- lancement des modules de sécurité
- génération du rapport de sécurité

Un scanner de vulnérabilités est donc un outil complexe qui peut faire appel à de nombreux programmes spécifiques pour chacune des tâches pré-citées.

2.2 Cibles

Un scanner de vulnérabilités est théoriquement capable de tester tout élément joignable par une adresse IP :

- Ordinateur
- Serveur
- Routeur, commutateur, pare-feu
- Smartphone
- Objet connecté
- Site Web
- Automate, robot, machine
- Caméra IP
- IPBX, poste téléphonique sur IP
- etc.

Le fait de pouvoir joindre un élément n'implique cependant pas forcément que son niveau de sécurité puisse être audité correctement. Pour cela, le scanner doit embarquer les modules de sécurité idoines dans son catalogue. Par exemple, si une cible ne possède que le port UDP 161 ouvert avec le service SNMP, un scanner pourra reconnaître que cette cible est active mais ne pourra juger son niveau de sécurité qu'en incorporant des modules d'attaque contre SNMP.

a) Restitution des résultats

La visualisation et la restitution des résultats se fait traditionnellement via deux paradigmes. Premièrement, une vue "par vulnérabilité" permettant de lister toutes les vulnérabilités identifiées dans le scan et de donner pour chacune d'elle la liste des machines affectées. Deuxièmement, une vue "par machine" listant les cibles de l'audit associées à la liste de leur vulnérabilités respectives.

Traditionnellement, en sécurité informatique, les vulnérabilités sont restituées par ordre de criticité suivant une échelle à 4 niveaux:

- Critiques: les vulnérabilités permettant généralement une prise de contrôle ou une exécution de commande à distance dont l'exploitation est relativement simple
- Majeures: les vulnérabilités permettant une prise de contrôle ou une exécution de commande à distance dont l'exploitation demeure complexe
- Moyennes: les vulnérabilités ayant des impacts limités ou dont l'exploitation nécessite des conditions initiales non triviales
- Mineures: les vulnérabilités ayant des impacts faibles ou nuls à moins d'être combinées à d'autres vulnérabilités plus importantes

L'état de l'art associe à chaque vulnérabilité un score entre 0 et 10 appelé CVSS (*Common Vulnerability Scoring System*) qui dépend des caractéristiques de cette vulnérabilité. La version 3 du CVSS prend en compte, au minimum, les éléments suivants :

- Le vecteur d'attaque : est-ce que l'attaquant peut venir de n'importe où ou bien doit-il avoir une position de départ privilégiée ?
- Complexité : exploiter la vulnérabilité est-il trivial (par exemple si un exploit existe) ou bien hautement techniques ?
- Privilèges requis : l'attaquant doit-l disposer d'accès préalables (un compte utilisateur par exemple) pour pouvoir mener son action ?
- Interaction avec un utilisateur: l'attaquant doit-il amener la victime à effectuer une action pour que son attaque réussisse (comme l'inciter à cliquer sur un lien) ?
- Périmètre : est-ce qu'une exploitation permet à l'attaquant d'avoir accès à de nouvelles cibles ?
- Impacts : une exploitation réussie entraîne-t-elle des pertes de confidentialité/disponibilité/intégrité ?

Le score CVSS est régulièrement utilisé par les scanners de vulnérabilités pour pondérer les risques associés à une cible.

Les scanners doivent donner à l'utilisateur tous les éléments pertinents pour sa compréhension de la vulnérabilité. Traditionnellement, une description de vulnérabilité comporte les éléments suivants :

- Le nom de la vulnérabilité
- Sa criticité
- La cible touchée
- Une brève description de sa nature
- Une référence à une base de connaissance type CVE, OSVDB, DSA...
- Une mention de la simplicité de l'exploitation
- Une description de l'impact en cas d'exploitation réussie
- Une ou plusieurs préconisations pour la résoudre

Parfois, d'autres éléments y sont ajoutés:

- Le niveau de confiance à accorder à la vulnérabilité : quantification du risque qu'il s'agisse ou non d'un faux positif
- S'il existe ou non un exploit automatique

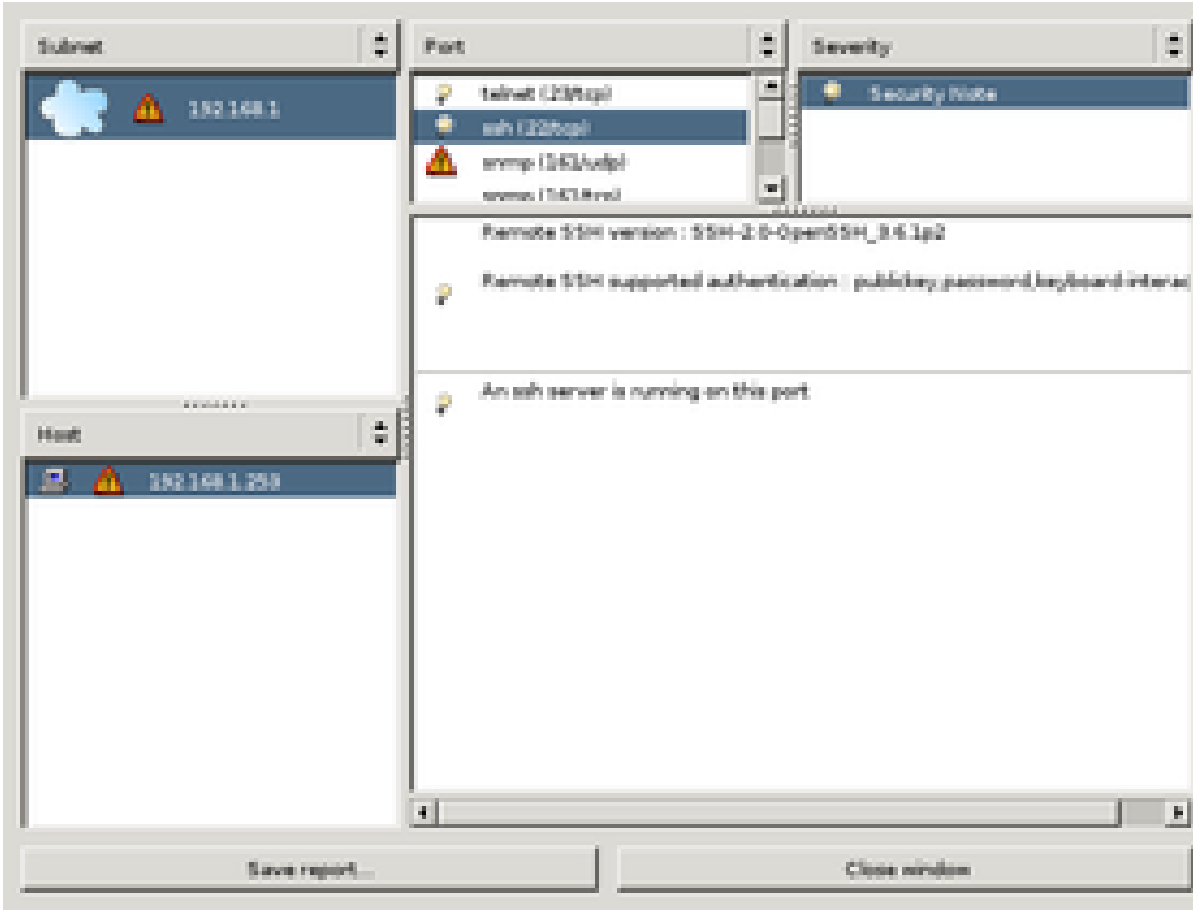
- Un extrait des données ayant permis au module de sécurité de conclure à la présence de cette faille
- La famille de vulnérabilité (authentification, mise à jour, etc.)
- Des liens pour en savoir plus (notamment pour des explications plus détaillées du fonctionnement technique de la vulnérabilité)

Les rapports sont souvent exportables aux formats PDF, CSV, HTML, etc.

LE NESSUS

1. Généralités

Nessus



Développeur	Tenable Network Security
Dernière version	5.2.6 (24 mars 2014)[1]
Environnement	Multiplate-forme
Type	Scanner de vulnérabilité
Licence	GNU GPL, puis propriétaire

Site web	nessus.org [archive]

Nessus est un outil de sécurité informatique. Il signale les faiblesses potentielles ou avérées sur les machines testées. Ceci inclut, entre autres :

- les services vulnérables à des attaques permettant la prise de contrôle de la machine, l'accès à des informations sensibles (lecture de fichiers confidentiels par exemple), des dénis de service...
- les fautes de configuration (relais de messagerie ouvert par exemple)
- les patchs de sécurité non appliqués, que les failles corrigées soient exploitables ou non dans la configuration testée
- les mots de passe par défaut, quelques mots de passe communs, et l'absence de mots de passe sur certains comptes systèmes. Nessus peut aussi appeler le programme externe Hydra (de) pour attaquer les mots de passe à l'aide d'un dictionnaire.
- les services jugés *faibles* (on suggère par exemple de remplacer Telnet par SSH)
- les dénis de service contre la pile TCP/IP

III.3 TESTS D'INTRUSION AVEC L'OUTILL DE SECURITE INFORMATIQUE SNORT

Présentation

Snort est un système de détection d'intrusion à partir d'analyse du trafic en temps réel. Pour cela, il effectue des analyses de protocole, recherche/correspondance de contenu et peut être utilisé pour détecter une grande variété d'attaques et de sondes comme des dépassements de buffers, scans, attaques sur des CGI, sondes SMB, essai d'OS fingerprintings et d'autres. Lorsque Snort détecte une intrusion, il peut soit prévenir l'administrateur système par un message sms/mail/vocal, soit interdire tout trafic provenant de l'attaquant, soit interdire tout trafic d'une classe d'adresse.

Toutefois Snort nécessite d'être maintenu à jour régulièrement pour rester efficace. Pour obtenir une représentation visuelle, on pourra utiliser des paquets tels que SnortSnarf, ACID, sguil ou BASE (Basic Analysis and Security Engine).

Snort a trois fonctionnalités :

- Sniffer le réseau : permet d'observer les paquets qui circulent sur le réseau.
- Générer des fichiers logs : génère des fichiers de logs au lieu d'afficher à l'écran les paquets.
- Détecter les intrusions : permet de détecter des intrusions suivant des règles contenues dans des fichiers. Ces fichiers sont téléchargeables sur http://www.snort.org/pub-bin/downloads.cgi

Snort s'utilise en ligne de commande et possède de nombreuses options…

Sniffer le réseau

Afficher les en-têtes de paquets TCP/IP
snort –v

Afficher les en-têtes de paquets IP, TCP/UDP/ICMP et les données
snort –vd

Pour afficher tous les paquets en transit sur l'interface eth0 :
snort -dvi eth0

```
03/30-23:01:37.323403 192.168.0.113:22 -> 82.242.68.85:1468
TCP TTL:64 TOS:0x10 ID:43060 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0xC5184FA2  Ack: 0xF09B7CDB  Win: 0x53C0  TcpLen: 20
4B 64 58 7C 2D C7 7D 74 22 C9 F2 A0 AF F0 1C 5E  KdX|-.}t"......^
10 7F 52 B8 C1 31 75 DF 35 E4 6F 62 C5 D1 48 E1  ..R..1u.5.ob..H.
E3 A0 0C 53 6C C4 8F 2D 55 EA 20 26 7C 6E 90 65  ...Sl..-U. &|n.e
A7 F4 C6 32 3A 35 AB 9B 12 26 81 88 AA 32 E6 31  ...2:5...&...2.1
F1 88 54 69 5C EB 6C 6C 9D 0E 97 F2 A7 A7 93 68  ..Ti\.ll.......h
…
…
78 DE 01 E9 BB EC 5D 97 18 1D 1E 37 53 66 66 C9  x.....]....7Sff.
BD 7E 83 F1 37 97 83 DA 75 39 4E C1 F1 0A 93 58  .~..7...u9N....X
93 72 DC 2C F0 77 B0 29 D8 45 D2 A6 AB E7 54 76  .r.,.w.).E....Tv
D2 59 F1 CE                                      .Y..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

```
*** Caught Int-Signal


===============================================================================

Snort received 339 packets
    Analyzed: 63(18.584%)
    Dropped: 120(35.398%)
    Outstanding: 156(46.018%)
===============================================================================
Breakdown by protocol:
    TCP: 63         (100.000%)
    UDP: 0          (0.000%)
   ICMP: 0          (0.000%)
    ARP: 0          (0.000%)
  EAPOL: 0          (0.000%)
   IPv6: 0          (0.000%)
ETHLOOP: 0          (0.000%)
    IPX: 0          (0.000%)
   FRAG: 0          (0.000%)
  OTHER: 0          (0.000%)
DISCARD: 0          (0.000%)
===============================================================================
Action Stats:
ALERTS: 0
LOGGED: 0
PASSED: 0
===============================================================================
Snort exiting
```

Générer des logs

Renvoie la sortie de la commande dans un fichier log dans le répertoire log
snort -dev -l ./log

Voir le fichier de log
snort -dv -r ./log/snort.log.1175196590

Voir seulement les paquets icmp du fichier de log
snort -dv -r ./log/snort.log.1175196590 icmp

Détecter des intrusions

Permet d'analyser et détecter des intrusions suivant des règles
snort -dev -l ./log -h 192.168.0.0/24 -c /etc/snort/snort.conf

```
Running in IDS mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Var 'tun0_ADDRESS' defined, value len = 24 chars, value = 10.8.0.1/255.255.255.255
Var 'any_ADDRESS' defined, value len = 15 chars, value = 0.0.0.0/0.0.0.0
Var 'lo_ADDRESS' defined, value len = 19 chars, value = 127.0.0.0/255.0.0.0
Initializing Preprocessors!
Initializing Plug-ins!
```

III.4 LA SURVEILLANCE ET ANALYSE DES JOURNAUX LOGS AVEC LES OUTILS DE SECURITE INFORMATIQUE SYSLOG-N ET AWASTATS

III.4.1 LA SURVEILLANCE DES JOURNAUX LOGS

a) Généralité

La surveillance (gestion de logs) comprend une approche de la gestion de grands volumes des messages de log générés par l'ordinateur (aussi connu comme journaux d'évènements, journalisation, etc.). La gestion des logs concerne en général[1]:

- La collecte des logs
- L'agrégation centralisée des logs
- Le stockage à long terme et la durée de rétention des logs
- La rotation des fichiers de logs
- L'analyse des logs (en temps réel et en vrac après une période de stockage)
- Les rapports et l'étude des logs.

Les principaux enjeux de l'implémentation de la gestion des logs concernent la sécurité[2], les opérations systèmes et réseaux et la conformité du système. Les logs sont générés par presque tous les appareils informatiques, et peut souvent être dirigé vers différents endroits à la fois locales, dans le système de fichiers ou à dans un système distant.

L'analyse effective de grands volumes de divers journaux peuvent poser de nombreux défis, tels que:

- La volume: les journaux/logs peuvent atteindre des centaines de giga-octets de données par jour pour une grande organisation. La collecte, la centralisation et le stockage de données à ce volume peut être difficile.
- Normalisation: les journaux sont produites dans de multiples formats. Le processus de normalisation est conçu pour fournir une sortie commune pour l'analyse de diverses sources.
- Vitesse: La vitesse à laquelle les journaux sont produites à partir de dispositifs peuvent rendre la collecte et l'agrégation difficile
- Véracité: Journal des événements peut ne pas être exacte. Cela est particulièrement problématique à partir de systèmes qui effectuent la détection, tels que les systèmes de détection d'intrusion.

Les utilisateurs et les utilisateurs potentiels de la gestion du journal peuvent acheter des d'outils propriétaires complets ou construire leur propre outil de gestion de logs et outils d'intelligence, ou bien utiliser un assemblage de composants open-sources, ou bien encore ou d'acquérir des (sous-)systèmes de fournisseurs commerciaux. La gestion des journalisations est un processus complexe et les organisations font souvent des erreurs dans leurs approches[3].

b) Surveillance de journaux Logs avec Syslog-ng

Présentation

Syslog-ng est un gestionnaire de journaux système de nouvelle génération (ng), anciennement syslog. A partir de syslog-ng, on peut centraliser les journaux de l'ordinateur lui-même mais aussi ceux des postes du réseau sur lequel il est connecté et permet de les trier facilement. Ce paquet est indispensable pour les administrateurs réseaux qui souhaitent garder un œil sur les performances de leurs machines. Parmi les avantages on peut citer la portabilité (Linux, FreeBSD, HP-UX, Solaris ou AIX), la configuration de base très avancée, l'export des logs vers une base de données mysql, la possibilité de chiffrer les logs envoyés via la technologie SSL, l'utilisation des protocoles UDP et TCP lors du transport des logs ou bien encore la possibilité de trier les logs suivant les contenus, leur provenance ou leur facilité, et ce en pouvant utiliser les expressions régulières.

Pour les postes clients sous linux il suffit de configurer syslog-ng pour que la destination soit l'ordinateur qui centralise les journaux dans le fichier de configuration syslog-ng.conf. Sous Windows il faut installer le logiciel Snare qui permet d'envoyer des trames syslog à la norme RFC.

Installation

Dans un premier temps il faut configurer mysql :

```
mysql_install_db
mysql -u root -p mysql
```

On va pouvoir ensuite installer et configurer php-syslog-ng, on va d'abord télécharger l'archive contenant les pages php de php-syslog-ng sur le site http://www.phpwizardry.com/php-syslog-ng/phpsyslogng-2.8.tar.gz .

Pour l'installer on va exécuter les commandes suivantes :

```
tar -xvzf phpsyslogng-2.8.tar.gz /var/www/html/php-syslog-ng
chown -R apache:apache /var/www/html/php-syslog-ng
```

On peut maintenant créer les tables permettant de stocker et ordonner les logs. Avant d'exécuter le script de configuration, il faut l'ouvrir et remplacer toutes les occurrences de PW_HERE par le mot de passe que l'on souhaite utiliser pour se connecter à l'interface :

```
mysql -uroot -p < dbsetup.sql
mysql -u root -p
> SET PASSWORD FOR syslogfeeder@localhost = PASSWORD ('syslogfeederpassword');
> SET PASSWORD FOR syslogadmin@localhost = PASSWORD ('syslogadminpassword');
```

Le script dbsetup.sql fourni avec le paquet php-syslog-ng est bien utile pour faire cela.

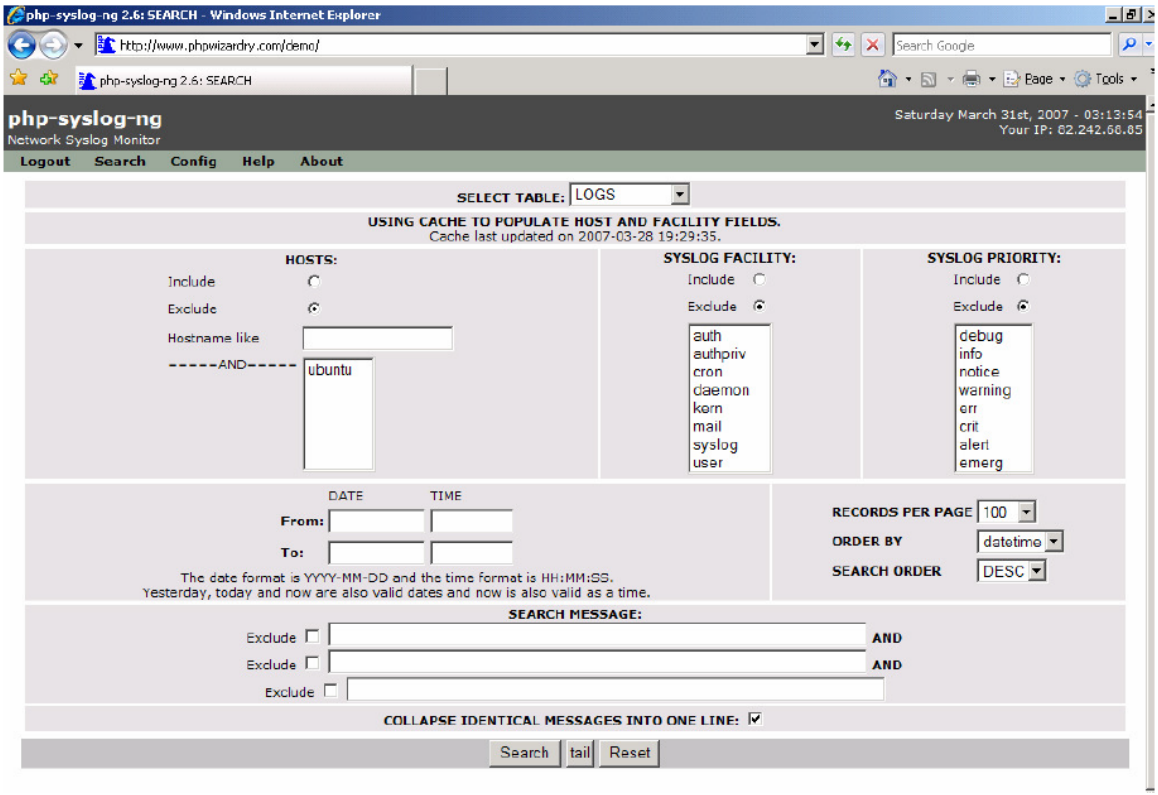
Ensuite il faut configurer syslog-ng pour qu'il rajoute les informations dans la base de données, pour cela on va rajouter dans le fichier /etc/syslog-ng/syslog-ng.conf les lignes suivantes :

```
destination d_mysql {
  pipe("/var/log/mysql.pipe"
    template("INSERT INTO logs
    (host, facility, priority, level, tag, datetime, program, msg)
    VALUES ( '$HOST', '$FACILITY', '$PRIORITY', '$LEVEL', '$TAG', '$YEAR-$MONTH-$DAY
$HOUR:$MIN:$SEC',
    '$PROGRAM', '$MSG' );\n") template-escape(yes));
};
```

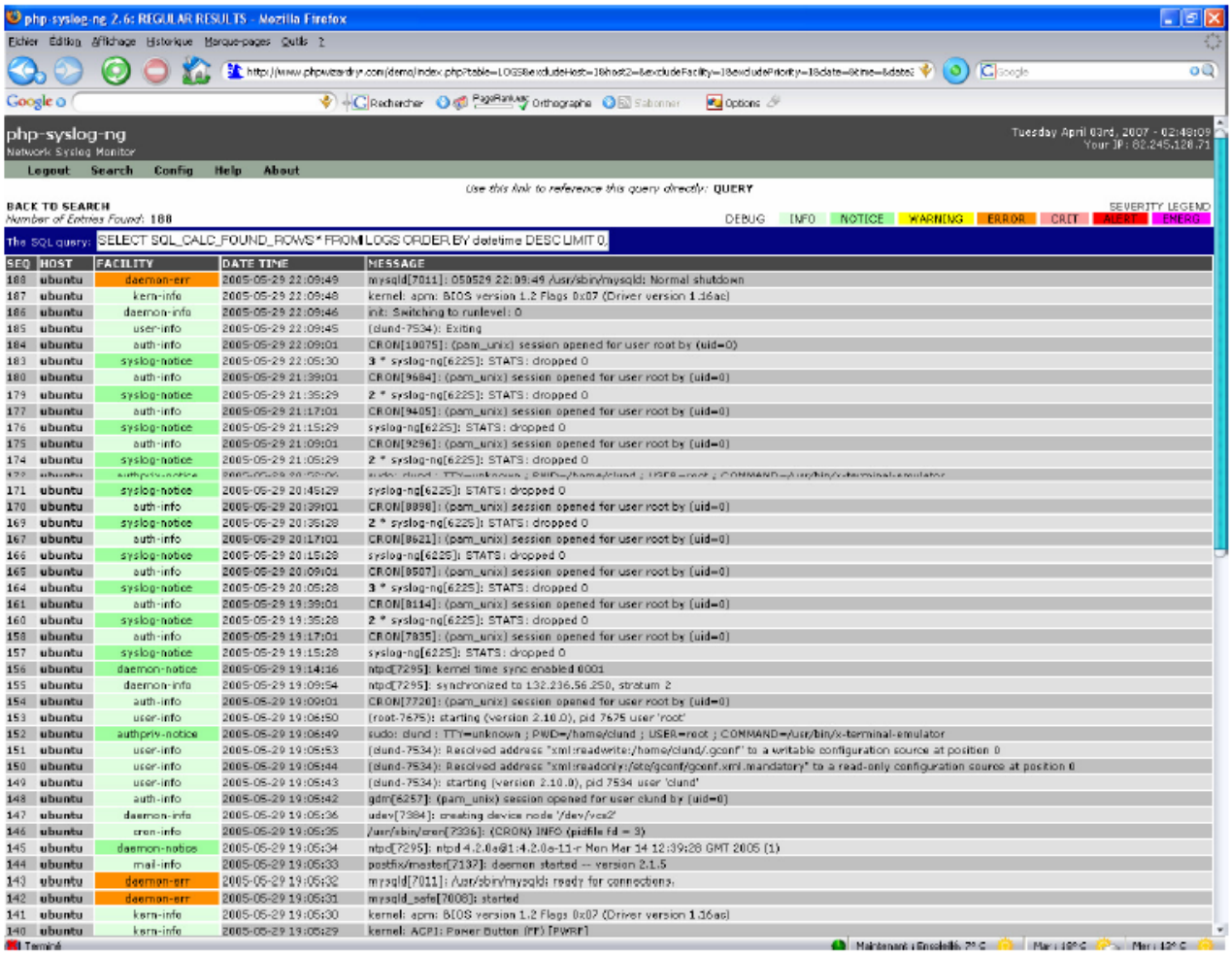
Il s'agit d'un nouveau champ destination permettant de rediriger les logs vers la base de données logs. Pour que l'ajout soit fait il faut rajouter un champ log qui va récupérer les données à partir de la source et les rediriger vers la destination. Nous avons ajouté le champ destination et le champ source doit déjà être présent dans le fichier de configuration de syslog-ng :

```
log {
  source(s_sys);
  destination(d_mysql);
};
```

Utilisation



La page d'accueil de php-syslog-ng permet de choisir ce que l'on veut afficher. On peut ainsi sélectionner les postes distants que l'on veut afficher ou non, les types de logs à afficher, en fonction du niveau d'erreur ou de la date…



La capture d'écran si dessus montre comment sont affichées les informations : numéro d'entrée, le poste, le type de logs avec le niveau d'erreur suivant la couleur, la date et l'heure et enfin le message associé.

III.4.2 ANALYSEUR DES JOURNAUX D'EVENEMENTS (LOG) AWSTATS

Présentation

Awstats est un outil gratuit, puissant et très fonctionnel qui fournit des statistiques graphiques de serveurs web, de streaming, ftp ou de mail. Cet analyseur de fichiers logs fonctionne en tant que CGI ou en ligne de commande et résume toutes les informations contenues dans ces fichiers sur des graphiques. Il sait lire les fichiers logs de la majorité des serveurs web tel que Apache (NCSA combined/XLF/ELF log format or common/CLF log format), WebStar, IIS (W3C log format) et d'autres serveurs web, proxy, streaming, mail et ftp.

Caractéristiques/Logiciels	AWStats	Analog	Webalizer
Version – Date	6.5 – Décembre 2005	6.0 - Décembre 2004	2.01-10 - Avril 2002
Langage	Perl	C	C
Plateformes	UNIX/Windows		
Sources disponibles	Oui		
Prix/License	Gratuit/GPL		
Format Apache combined	Oui		
Format Apache common	Quelques fonctionnalités		
Format IIS (W3C)	Oui		Avec un patch
Format personnalisé	Oui		Non
Web/Ftp/Mail log	Oui/oui/oui	Oui/non/non	

Configuration

Step 1

Lancer le script awstats_configure.pl (dans le répertoire /usr/bin sous Gentoo et dans /usr/share/awstats/tools/awstats_configure.pl sous fedora) comme ceci

```
perl awstats_configure.pl
```

Vérifier les points suivant dans httpd.conf

- *CustomLog /yourlogpath/yourlogfile combined* (au lieu de *common*)
- Utilisation à partir du site web

```
#
# Directives to add to your Apache conf file to allow use of AWStats as a CGI.
# Note that path "/usr/local/awstats/" must reflect your AWStats Installation path.
#
Alias /awstatsclasses "/usr/local/awstats/wwwroot/classes/"
Alias /awstatscss "/usr/local/awstats/wwwroot/css/"
Alias /awstatsicons "/usr/local/awstats/wwwroot/icon/"
ScriptAlias /awstats/ "/usr/local/awstats/wwwroot/cgi-bin/"
#
# This is to permit URL access to scripts/files in AWStats directory.
```

```
#
<Directory "/usr/local/awstats/wwwroot">
Options None
AllowOverride None
Order allow,deny
Allow from all
</Directory>
```

Dans Gentoo on /usr/share/webapps/awstats/6.5-r2/htdocs/ au lieu de /usr/local/awstats/wwwroot/ et /usr/share/webapps/awstats/6.5-r2/hostroot/cgi-bin au lieu de /usr/local/awstats/wwwroot/cgi-bin

Redémarrer Apache

```
/etc/init.d/apache2 restart
```

Step 2

Après avoir créé le awstats.mon.site.conf (dans /etc/awstats sous linux Gentoo) il est nécessaire de vérifier certains points dans ce fichier:

- **LogFile** : chemin relatif des logs du serveur
- **LogType** : W pour analyser les fichiers logs web
- **LogFormat** : 1 pour (pour "NCSA apache combined/ELF/XLF log format") ou utiliser un autre format (http://awstats.sourceforge.net/docs/awstats_config.html#LogFormat)
- **SiteDomain** : nom de domaine du site (www.mon.site.com)

Step 3

Il faut ensuite exécuter la commande suivante (sous Gentoo awstats.pl se trouve dans /usr/share/webapps/awstats/6.5-r2/hostroot/cgi-bin/) :

```
perl awstats.pl -config=mon.site –update
```

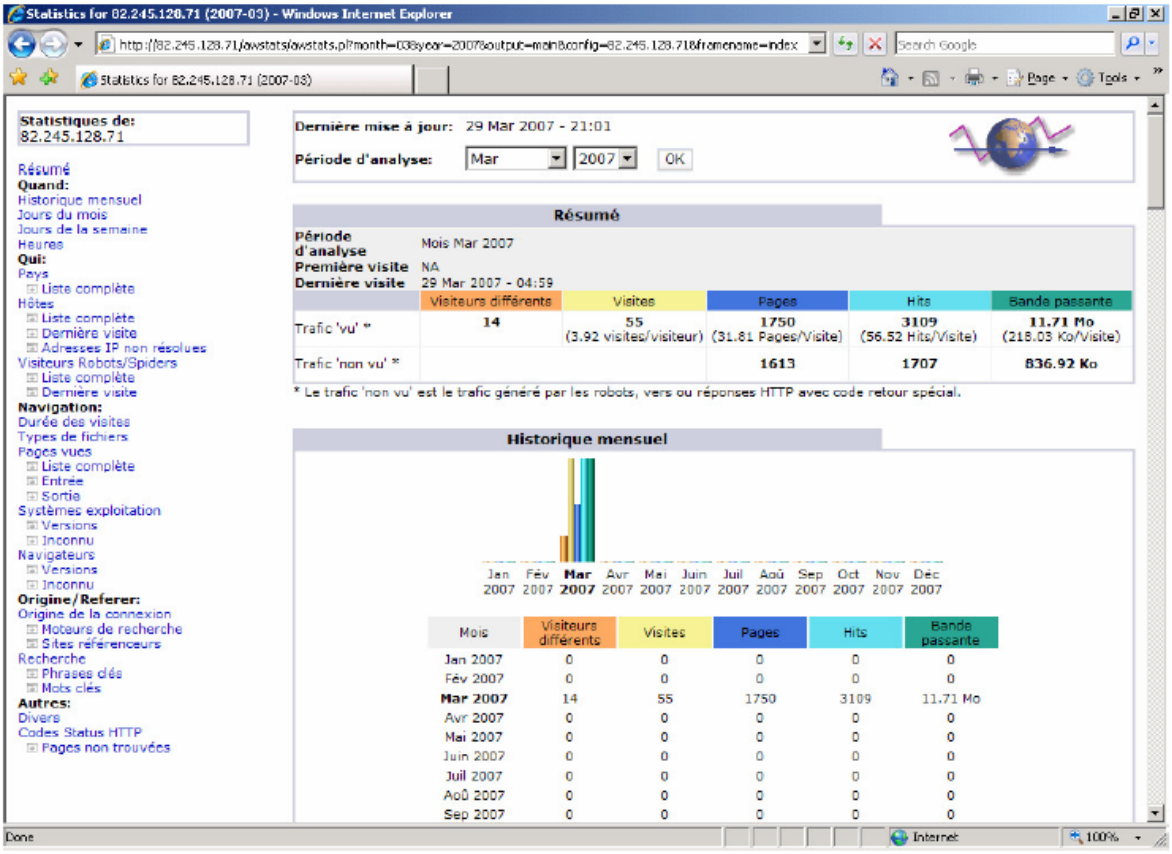
On devrait alors obtenir :

```
Update for config "/etc/awstats/awstats.mysite.conf"
With data in log file "/pathtoyourlog/yourlog.log"...
Phase 1 : First bypass old records, searching new record...
Searching new records from beginning of log file...
Phase 2 : Now process new records (Flush history on disk after 20000 hosts)...
Jumped lines in file: 0
Parsed lines in file: 225730
 Found 122 dropped records,
 Found 87 corrupted records,
 Found 0 old records,
 Found 225521 new qualified records.
```

Utilisation

http://www.mon.site/awstats/awstats.pl?config=mon.site

Si le paramètre AllowToUpdateStatsFromBrowser est positionné à 1 dans awstats.mon.site.conf il sera possible de mettre à jour les informations à partir du site web.



Comme le montre la capture d'écran d'Internet Explorer 7, Awstats fournit dans un navigateur les statistiques graphiques du serveur web de notre machine. Sur la gauche, un menu permet d'accéder aux sous rubriques : Résumé, Historique mensuel, Historique hebdomadaire, Historique journalier, Heures, Pays des visiteurs, Hôtes, Durées de visite, Types de fichier, Pages les plus visitées, Systèmes d'exploitation, Navigateurs…

CHAP.IV LES NORMES ET METHODES DE SECURITE INFORMATIQUE

Introduction

La sécurité du système d'information d'une entreprise est un requis important pour la poursuite de ses activités. Qu'il s'agisse de la dégradation de son image de marque, du vol de ses secrets de fabrication ou de la perte de ses données clients ; une catastrophe informatique a toujours des conséquences fâcheuses pouvant aller jusqu'au dépôt de bilan.
Organiser cette sécurité n'est pas chose facile, c'est pourquoi il existe des méthodes reconnues pour aider les responsables informatiques à mettre en place une bonne politique de sécurité et à procéder aux audits permettant d'en vérifier l'efficacité.
Le but de ce document n'est pas d'expliquer comment concevoir une politique de sécurité mais de présenter les méthodes existantes.

A. Politique de sécurité

Une politique de sécurité peut être vue comme l'ensemble des modèles d'organisation, des procédures et des bonnes pratiques techniques permettant d'assurer la sécurité du système d'information.
Mais qu'est-ce que la sécurité d'un SI ? Elle tourne autour des 5 principaux concepts suivants : l'intégrité des données, la confidentialité de l'information et des échanges, la disponibilité des services, l'authentification des utilisateurs et la non répudiation des transactions.
Pour garantir la sécurité, une politique de sécurité est généralement organisée autour de 3 axes majeurs : la sécurité physique des installations, la sécurité logique du système d'information et la sensibilisation des utilisateurs aux contraintes de sécurité.

B. Audit

Un audit de sécurité permet de mettre en évidence les faiblesses de la mise en œuvre d'une politique de sécurité. Le problème peut venir de la politique elle-même : mal conçue ou inadaptée aux besoins de l'entreprise, ou bien d'erreurs quand à sa mise en application.
Des audits sont nécessaires : suite à la mise en place initiale d'une politique de sécurité, puis régulièrement pour s'assurer que les mesures de sécurité sont mises à niveau et que les usages restent conformes aux procédures.

C. Normes de sécurité informatique

En ce qui concerne les normes de sécurité des SI, la famille de normes ISO 27000 constitue un véritable espoir pour les RSSI dans la mesure où elle apporte une aide indéniable dans la

définition, la construction et la déclinaison d'un SMSI efficace à travers une série de normes dédiées à la sécurité de l'information :

§ ISO/CEI 27001 **:** système de Gestion de la Sécurité de l'Information (ISMS) -Exigences ;

§ ISO/CEI 27002 : code de bonnes pratiques pour la gestion de la sécurité de l'information (anciennement ISO/CEI 17799) ;

§ ISO/CEI 27003 **:** système de Gestion de la Sécurité de l'Information (ISMS) - Guide d'implémentation ;

§ ISO/CEI 27004 **:** mesure de la gestion de la sécurité de l'information ;

§ ISO/CEI 27005 **:** gestion du risque en sécurité de l'information ;

§ ISO/CEI 27006 **:** exigences pour les organismes réalisant l'audit et la certification de Systèmes de Gestion de la Sécurité de l'Information (ISMS) ;

§ ISO/CEI 27007 **:** guide pour l'audit de Systèmes de Gestion de la Sécurité de l'Information (ISMS).

C. Présentation des principales normes

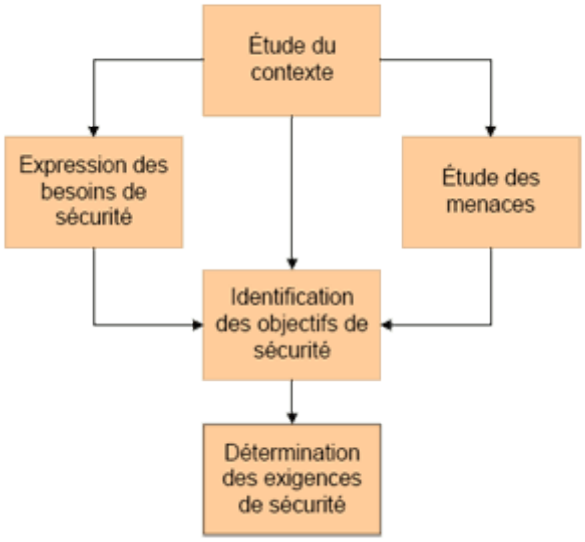
Seront abordées ici les principales méthodes employées en Europe et en Amérique du Nord.

C.1 EBIOS

EBIOS (Expression des Besoins et Identification des Objectifs de Sécurité) permet d'identifier les risques d'un SI et de proposer une politique de sécurité adaptée aux besoin de l'entreprise (ou d'une administration). Elle a été créée par la DCSSI (Direction Centrale de la Sécurité des Systèmes d'Information), du Ministrère de la Défence (France). Elle est destinée avant tout aux administrations françaises et aux entreprises.
La méthode EBIOS se compose de 5 guides (*Introduction, Démarche, Techniques, Outillages*) et d'un logiciel permettant de simplifier l'application de la méthodologie explicitée dans ces guides. Le logiciel libre et gratuit (les sources sont disponibles) permet de simplifier l'application de la méthode et d'automatiser la création des documents de synthèse.
La DCSSI possède un centre de formation où sont organisés des stages à destination des organismes publics français. Un club d'utilisateurs EBIOS a été créé en 2003 et constitue une communauté experts permettant le partage des expériences. Une base de connaissances à laquelle se connecte le logiciel EBIOS permet d'avoir à accès à la description d'un ensemble de vulnérabilités spécifiques, de contraintes de sécurité, de méthodes d'attaques. Elle peut être enrichie via le logiciel.
La méthode EBIOS est découpée en 5 étapes :

Les 5 étapes de la méthode EBIOS

1. étude du contexte
2. expression des besoins de sécurité
3. étude des menaces
4. identification des objectifs de sécurité
5. détermination des exigences de sécurité



Démarche EBIOS globale

L'**étude du contexte** permet d'identifier quel système d'information est la cible de l'étude. Cette étape délimite le périmètre de l'étude : présentation de l'entreprise, architecture du système d'information, contraintes techniques et réglementaires, enjeux commerciaux. Mais est aussi étudié le détail des équipements, des logiciels et de l'organisation humaine de l'entreprise.

L'**expression des besoins de sécurité** permet d'estimer les risques et de définir les critères de risque. Les utilisateurs du SI expriment durant cette étape leurs besoins de sécurité en fonction des impacts qu'ils jugent inacceptables.

L'**étude des menaces** permet d'identifier les risques en fonction non plus des besoins des utilisateurs mais en fonction de l'architecture technique du système d'information. Ainsi la liste des vulnérabilités et des types d'attaques est dressée en fonction des matériels, de l'architecture réseau et des logiciels employés. Et ce, quelles que soient leur origine (humaine, matérielle, environnementale) et leur cause (accidentelle, délibérée).

L'**identification des objectifs de sécurité** confronte les besoins de sécurité exprimés et les menaces identifiées afin de mettre en évidence les risques contre lesquels le SI doit être protégé. Ces objectifs vont former un cahier des charges de sécurité qui traduira le choix fait sur le niveau de résistance aux menaces en fonction des exigences de sécurité.

La **détermination des exigences de sécurité** permet de déterminer jusqu'où on devra aller dans les exigences de sécurité. Il est évident qu'une entreprise ne peut faire face à tout type de risques, certains doivent être acceptés afin que le coût de la protection ne soit pas exhorbitant. C'est notamment la stratégie de gestion du risque tel que cela est défini dans un plan de risque qui sera déterminé ici : accepter, réduire ou refuser un risque. Cette stratégie est décidée en fonction du coût des conséquences du risque et de sa probabilité de survenue. La justification argumentée de ces exigences donne l'assurance d'une juste évaluation.

EBIOS fournit donc la méthode permettant de contruire une politique de sécurité en fonction d'une analyse des risques qui repose sur le contexte de l'entreprise et des vulnérabilités liées à son SI.
.

C.2 Mehari

Mehari (MEthode Harmonisée d'Analyse de RIsques) est dévelopée par le CLUSIF depuis 1995, elle est dérivée des méthodes Melisa et Marion. Existant en langue française et en anglais, elle est utilisée par de nombreuses entreprises publiques ainsi que par le secteur privé.
Le logiciel RISICARE développé par la société BUC SA est un outil de gestion des risques basé sur la méthode Mehari.
La démarche générale de Mehari consiste en l'analyse des enjeux de sécurité : quels sont les scénarios redoutés ?, et en la classification préalable des entités du SI en fonction de trois critères de sécurité de base (confidentialité, intégrité, disponibilité). Ces enjeux expriment les dysfonctionnements ayant un impact direct sur l'activité de l'entreprise. Puis, des audits identifient les vulnérabilités du SI. Et enfin, l'analyse des risques proprement dite est réalisée.

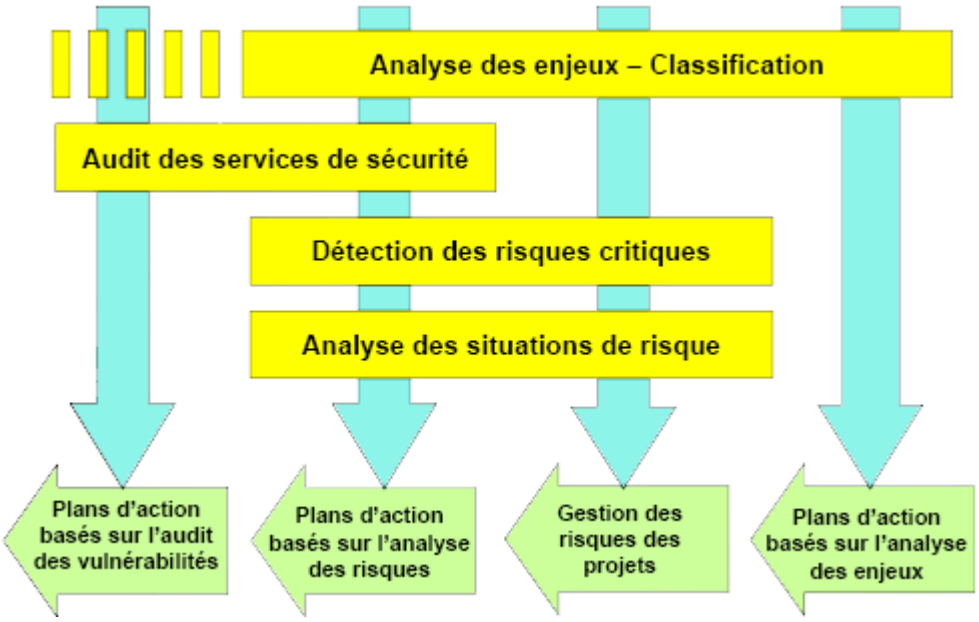


Schéma général de la méthode Mehari

Mehari s'articule autour de 3 types de livrables :

Plans de la méthode Mehari

1. le Plan Stratégique de Sécurité (PSS)
2. les Plans Opérationnels de Sécurité (POS)
3. le Plan Opérationnel d'Entreprise (POE)

Le **Plan Stratégique de Sécurité** fixe les objectifs de sécurité ainsi que les métriques permettant de les mesurer. C'est à ce stade que le niveau de gravité des risques encourus par l'entreprise est évalué. Il définit la politique de sécurité ainsi que la charte d'utilisation du SI pour ses utilisateurs.

Les **Plans Opérationnels de Sécurité** définissent pour chaque site les mesures de sécurité qui doivent être mises en oeuvre. Pour cela, ils élaborent des scénarios de compromission et audite les services du SI. Sur la base de cet audit, une évaluation de chaque risque (probabilité, impact) est réalisée permettant par la suite d'exprimer les besoins de sécurité, et par la même les mesures de protections nécessaires. Enfin, une planification de la mise à niveau de la sécurité du SI est faite.

Le **Plan Opérationnel d'Entreprise** assure le suivi de la sécurité par l'élaboration d'indicateurs sur les risques identifiés et le choix des scénarios de catastrophe contre lesquels il faut se prémunir.

Des bases de connaissances permettent d'automatiser certains calculs de gravité des scénarios de risques, proposent des liens entre menaces et parades...
Mehari apporte une démarche centrée sur les besoins de continuité d'activité de l'entreprise et fournit des livrables types aidés d'un guide méthodologie. Les audits qu'elle propose permettent la création de plan d'actions concrets. Cette méthode permet donc de construire une politique de sécurité destinée à pallier les vulnérabilités constatées lors des audits du *Plans Opérationnels de Sécurité* et d'atteindre le niveau de sécurité correspondant aux objectifs fixés dans le *Plan Stratégique de Sécurité*.

CHAPITRE 5. NOTIONS DE CRYPTOGRAPHIE

5.1 INTRODUCTION A LA CRYPTOGRAPHIE

1.1 La Cryptographie est l'étude des méthodes donnant la possibilité d'envoyer des données de manière confidentielle sur un support donné. Les fonctions principales de la Cryptographie sont le Chiffrement, le Déchiffrement et la Clé.

La cryptographie utilise des concepts issus de nombreux domaines (Informatique, Mathématiques, Electronique). Toutefois, les techniques évoluent et trouvent aujourd'hui régulièrement racine dans d'autres branches (Biologie, Physique, etc.)

2.1 Vocabulaire de base

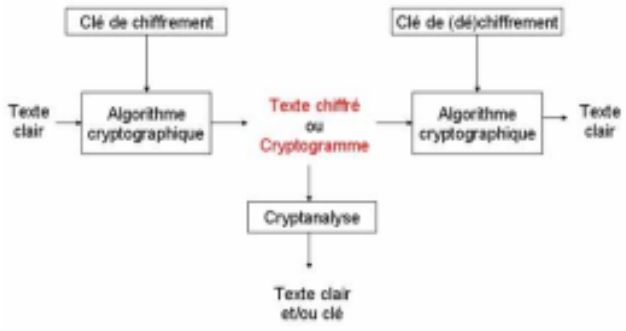


FIG. 2.1 – Protocole de chiffrement

Cryptologie : Il s'agit d'une science mathématique comportant deux branches : la cryptographie et la cryptanalyse

Cryptographie : La cryptographie est l'étude des méthodes donnant la possibilité d'envoyer des données de manière confidentielle sur un support donné.

Chiffrement : Le chiffrement consiste à transformer une donnée (texte, message, ...) afin de la rendre incompréhensible par une personne autre que celui qui a créé le message et celui qui en est le destinataire. La fonction permettant de retrouver le texte clair à partir du texte chiffré porte le nom de *déchiffrement*.

Texte chiffré : Appelé également *cryptogramme*, le texte chiffré est le résultat de l'application d'un chiffrement à un texte clair.

Clef : Il s'agit du paramètre impliqué et autorisant des opérations de chiffrement et/ou déchiffrement. Dans le cas d'un algorithme symétrique, la clef est identique lors des deux opérations. Dans le cas d'algorithmes asymétriques, elle diffère pour les deux opérations.

Cryptanalyse : Opposée à la cryptographie, elle a pour but de retrouver le texte clair à partir de textes chiffrés en déterminant les failles des algorithmes utilisés.

Cryptosystème : Il est défini comme l'ensemble des clés possibles (espace de clés), des textes clairs et chiffrés possibles associés à un algorithme donné.

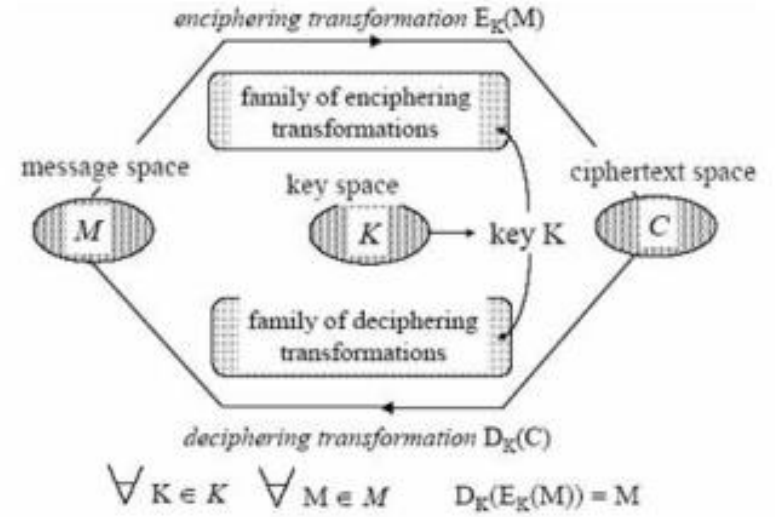


FIG. 2.2 – Schéma d'un cryptosystème

L'algorithme est en réalité un triplet d'algorithmes :
- l'un générant les clés K,
- un autre pour chiffrer M, et
- un troisième pour déchiffrer C.

Remarque : On parle de "décryptage" pour désigner l'action permettant de retrouver le texte clair sans connaître la clef de déchiffrement. On emploie également parfois les termes "cryptage" et "crypter" pour qualifier l'action de chiffrer un message. Les mots "encryptage" et "(en)cryptement" sont des anglicismes dérivés du verbe "to encrypt".

2.2 Notations

En cryptographie, la propriété de base est que

$$M = D(E(M))$$

où

- M représente le texte clair,
- C est le texte chiffré,
- K est la clé (dans le cas d'un algorithme à clé symétrique), E_k et D_k dans le cas d'algorithmes asymétriques,
- $E(x)$ est la fonction de chiffrement, et
- $D(x)$ est la fonction de déchiffrement.

Ainsi, avec un algorithme à clef symétrique,

$$M = D(C) \text{ si } C = E(M)$$

2.3 Principe de Kerckhoff

> La sécurité du chiffre ne doit pas dépendre de ce qui ne peut pas être facilement changé.

En d'autres termes, aucun secret ne doit résider dans l'algorithme mais plutôt dans la clé. Sans celle-ci, il doit être impossible de retrouver le texte clair à partir du texte chiffré. Par contre, si on connaît K, le déchiffrement est immédiat.

On parle aussi de la Maxime de Shannon, dérivée du principe énoncé ci-dessus : *L'adversaire connait le système.*

Remarque : Il faut distinguer les termes "Secret" et "Robustesse" d'un algorithme. Le secret de l'algorithme revient à cacher les concepts de celui-ci, ainsi que les méthodes utilisées (fonctions mathématiques). La robustesse quant à elle désigne la résistance de l'algorithme à diverses attaques qui seront explicitées dans la suite de ces notes.

2.4 La publication des algorithmes

Selon l'endroit où réside le secret, on peut parler d'algorithme secret ou d'algorithme publié[1]. Chacun possède ses atouts et inconvénients.

2.4.1 Algorithme secret

- La cryptanalyse, souvent basée sur le secret de la clé, doit ici en plus retrouver l'entièreté de l'algorithme (mécanisme de récupération).
- Souvent, de tels algorithmes sont utilisés par un plus petit nombre d'utilisateurs. Et comme souvent dans ce cas, moins il y a de monde l'utilisant, moins il y a d'intérêts à le casser.
- De tels algorithmes sont rarement distribués par delà les frontières, afin de garder un nombre d'utilisateurs restreint.

2.4.2 Algorithme publié

- Puisque l'algorithme est publié, tout le monde a le droit de l'explorer. Ainsi, les failles (laissées intentionnellement ou non par les concepteurs) peuvent être plus facilement découvertes. La sécurité en est donc améliorée.
- Comme la publication est autorisée, il n'est pas nécessaire de chercher à protéger le code contre le reverse-engineering.
- Cette publication permet d'étendre les travaux sur l'algorithme au niveau mondial. Toute une série d'implémentations logicielles peuvent donc être réalisées.
- Tout le monde utilise la même version publique ce qui permet une standardisation générale.

En conséquence, on préférera les algorithmes publiés, souvent plus sûrs pour les raisons explicitées ci-dessus.

2.3 Principe de Kerckhoff

> La sécurité du chiffre ne doit pas dépendre de ce qui ne peut pas être facilement changé.

En d'autres termes, aucun secret ne doit résider dans l'algorithme mais plutôt dans la clé. Sans celle-ci, il doit être impossible de retrouver le texte clair à partir du texte chiffré. Par contre, si on connaît K, le déchiffrement est immédiat.

On parle aussi de la Maxime de Shannon, dérivée du principe énoncé ci-dessus : *L'adversaire connait le système.*

Remarque : Il faut distinguer les termes "Secret" et "Robustesse" d'un algorithme. Le secret de l'algorithme revient à cacher les concepts de celui-ci, ainsi que les méthodes utilisées (fonctions mathématiques). La robustesse quant à elle désigne la résistance de l'algorithme à diverses attaques qui seront explicitées dans la suite de ces notes.

2.4 La publication des algorithmes

Selon l'endroit où réside le secret, on peut parler d'algorithme secret ou d'algorithme publié[1]. Chacun possède ses atouts et inconvénients.

2.4.1 Algorithme secret

- La cryptanalyse, souvent basée sur le secret de la clé, doit ici en plus retrouver l'entièreté de l'algorithme (mécanisme de récupération).
- Souvent, de tels algorithmes sont utilisés par un plus petit nombre d'utilisateurs. Et comme souvent dans ce cas, moins il y a de monde l'utilisant, moins il y a d'intérêts à le casser.
- De tels algorithmes sont rarement distribués par delà les frontières, afin de garder un nombre d'utilisateurs restreint.

2.4.2 Algorithme publié

- Puisque l'algorithme est publié, tout le monde a le droit de l'explorer. Ainsi, les failles (laissées intentionnellement ou non par les concepteurs) peuvent être plus facilement découvertes. La sécurité en est donc améliorée.
- Comme la publication est autorisée, il n'est pas nécessaire de chercher à protéger le code contre le reverse-engineering.
- Cette publication permet d'étendre les travaux sur l'algorithme au niveau mondial. Toute une série d'implémentations logicielles peuvent donc être réalisées.
- Tout le monde utilise la même version publique ce qui permet une standardisation générale.

En conséquence, on préférera les algorithmes publiés, souvent plus sûrs pour les raisons explicitées ci-dessus.

2.5 Les principaux concepts cryptographiques

2.5.1 Cryptosystème à clé symétrique

Caractéristiques :

- Les clés sont identiques : $K_E = K_D = K$,
- La clé doit rester secrète,
- Les algorithmes les plus répandus sont le DES, AES, 3DES, ...
- Au niveau de la génération des clés, elle est choisie aléatoirement dans l'espace des clés,
- Ces algorithmes sont basés sur des opérations de transposition et de substitution des bits du texte clair en fonction de la clé,
- La taille des clés est souvent de l'ordre de 128 bits. Le DES en utilise 56, mais l'AES peut aller jusque 256,
- L'avantage principal de ce mode de chiffrement est sa rapidité,
- Le principal désavantage réside dans la distribution des clés : pour une meilleure sécurité, on préfèrera l'échange manuel. Malheureusement, pour de grands systèmes, le nombre de clés peut devenir conséquent. C'est pourquoi on utilisera souvent des échanges sécurisés pour transmettre les clés. En effet, pour un système à N utilisateurs, il y aura $N.(N-1)/2$ paires de clés.

FIG. 2.3 – Chiffrement symétrique

2.5.2 Cryptosystème à clé publique

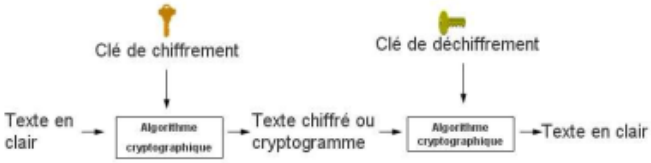


FIG. 2.4 – Chiffrement asymétrique

Caractéristiques :

- Une clé publique P_K (symbolisée par la clé verticale),
- Une clé privée secrète S_K (symbolisée par la clé horizontale),
- Propriété : La connaissance de P_K ne permet pas de déduire S_K,
- $D_{S_K}(E_{P_K}(M)) = M$,
- L'algorithme de cryptographie asymétrique le plus connu est le RSA,
- Le principe de ce genre d'algorithme est qu'il s'agit d'une fonction unidirectionnelle à trappe. Une telle fonction à la particularité d'être facile à calculer dans un sens, mais difficile voire impossible dans le sens inverse. La seule manière de pouvoir réaliser le calcul inverse est de connaître une trappe. Une trappe pourrait par exemple être une faille dans le générateur de clés. Cette faille peut être soit intentionnelle de la part du concepteur (définition *stricte* d'une trappe) ou accidentelle.
- Les algorithmes se basent sur des concepts mathématiques tels que l'exponentiation de grands nombres premiers (RSA), le problème des logarithmes discrets (ElGamal), ou encore le problème du sac à dos (Merkle-Hellman).
- La taille des clés s'étend de 512 bits à 2048 bits en standard. Dans le cas du RSA, une clé de 512 bits n'est plus sûre au sens "militaire" du terme, mais est toujours utilisable de particulier à particulier.
- Au niveau des performances, le chiffrement par voie asymétrique est environ 1000 fois plus lent que le chiffrement symétrique.
- Cependant, à l'inverse du chiffrement symétrique où le nombre de clés est le problème majeur, ici, seules n paires sont nécessaires. En effet, chaque utilisateur possède une paire (S_K, P_K) et tous les transferts de message ont lieu avec ces clés.
- La distribution des clés est grandement facilitée car l'échange de clés secrètes n'est plus nécessaire. Chaque utilisateur[2] conserve sa clé secrète sans jamais la divulguer. Seule la clé publique devra être distribuée.

2.5.3 Fonction de hachage

Il s'agit de la troisième grande famille d'algorithmes utilisés en cryptographie. Le principe est qu'un message clair de longueur quelconque doit être transformé en un message de longueur fixe inférieure à celle de départ. Le message réduit portera le nom de "Haché" ou de "Condensé". L'intérêt est d'utiliser ce condensé comme empreinte digitale du message original afin que ce dernier soit identifié de manière univoque. Deux caractéristiques (théoriques) importantes sont les suivantes :

1. Ce sont des fonctions unidirectionnelles :

$$\textit{A partir de } H(M) \textit{ il est impossible de retrouver } M.$$

2. Ce sont des fonctions sans collisions :

$$\textit{A partir de } H(M) \textit{ et } M \textit{ il est impossible de trouver } M' \neq M \textit{ tel que } H(M') = H(M).$$

Il est bien entendu que le terme "impossible" n'est pas toujours à prendre au pied de la lettre ! Il s'agit ici de concepts théoriques. La réalité est quelque peu différente. Ainsi, pour le caractère "sans collision", dans les faits, cela est "très difficile" dans le meilleur des cas, mais jamais impossible, comme le bon sens le laisse penser.

2.5.4 Protocoles cryptographiques

Dès que plusieurs entités sont impliquées dans un échange de messages sécurisés, des règles doivent déterminer l'ensemble des opérations cryptographiques à réaliser, leur séquence, afin de sécuriser la communication C'est ce que l'on appelle les protocoles cryptographiques.

Lorsque l'on parle de "sécuriser un échange", on souhaite prêter attention aux 3 services suivants : la confidentialité, l'intégrité et l'authentification.

Signalons la distinction entre "services" (confidentialité, intégrité, etc.) et "mécanismes" (les moyens utilisés : chiffrement, signature, hachage, etc.).

2.5.4.1 Confidentialité

Elle est amenée par le chiffrement du message. Dans le cas de systèmes à clés symétriques, la même clé est utilisée pour $E_K(M)$ et $D_K(C)$. Ce type de chiffrement nécessite un échange sûr préalable de la clé K entre les entités A et B.

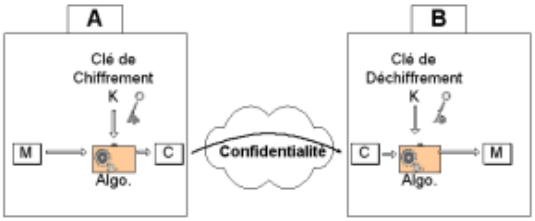


FIG. 2.5 – Confidentialité d'un système symétrique

Comme dit précédemment, à l'aide d'un cryptosystème asymétrique, cet échange préalable n'est pas nécessaire. Chaque entité possède sa propre paire de clés. On aura donc la paire P_{KA}, S_{KA} pour l'entité A et la paire P_{KB}, S_{KB} pour l'entité B.

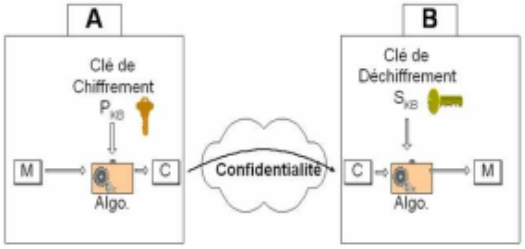


FIG. 2.6 – Confidentialité d'un système asymétrique

Comme dit précédemment, à l'aide d'un cryptosystème asymétrique, cet échange préalable n'est pas nécessaire. Chaque entité possède sa propre paire de clés. On aura donc la paire P_{KA}, S_{KA} pour l'entité A et la paire P_{KB}, S_{KB} pour l'entité B.

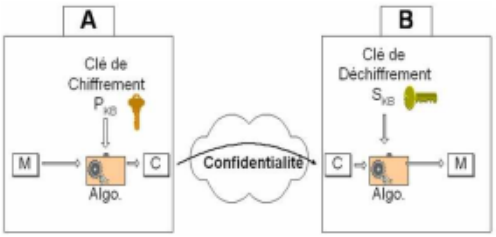


FIG. 2.6 – Confidentialité d'un système asymétrique

En marge de ces deux systèmes, existe également un système appelé "hybride" (figure 2.7), reposant comme son nom l'indique sur les deux systèmes précédents. Par l'intermédiaire du système à clé publique, on sécurise l'échange de la clé K. Ensuite, les deux parties ayant acquis de manière sécurisée cette clé de chiffrement K^3, on utilisera le système à clé symétrique pour chiffrer le message.

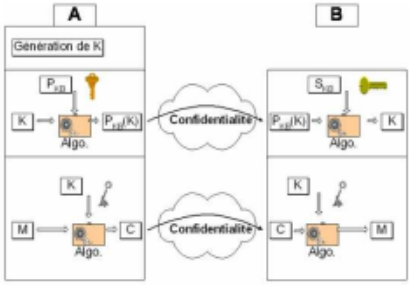


FIG. 2.7 – Confidentialité d'un système hybride

Il faut ici vérifier si le message n'a pas subi de modification durant la communication. C'est ici qu'interviennent les fonctions de hachage.

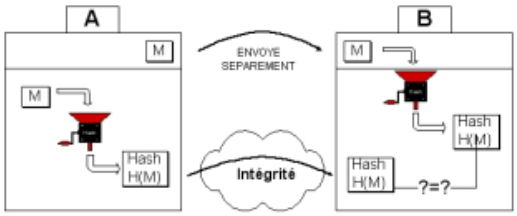


FIG. 2.8 – Vérification de l'intégrité par fonction de hachage

Dans la figure 2.8, on ne parle pas de l'envoi du message. On prête uniquement l'attention à la vérification de l'intégrité.

2.5.4.3 Authentification

Elle a lieu à plusieurs niveaux.

– Au niveau des parties communicantes, dans le cas d'un système symétrique (figure 2.9) ou asymétrique (figure 2.10). A la première figure, R_a est une nonce (p. ex. nombre aléatoire), propre à l'utilisateur A. Les lettres A et B représentent des identificateurs personnels.

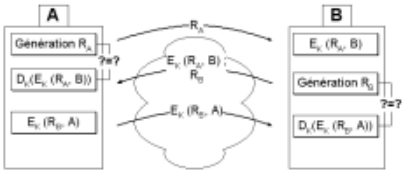


FIG. 2.9 – Authentification dans un système symétrique

A la seconde figure, la clé de chiffrement utilisée est bien la clé privée. Comme le propriétaire de cette clé est le seul à la connaitre, cela prouve qu'il est bien la personne ayant chiffré le message. Attention, dans cet exemple, seule l'authentification est souhaitée. Le message envoyé pourra être lu par toute personne possédant la clé publique, c'est-à-dire, n'importe qui. La confidentialité est ici nulle.

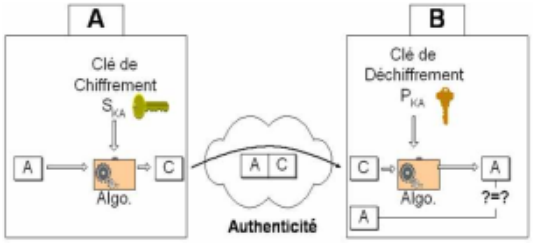


FIG. 2.10 – Authentification dans un système asymétrique

– Au niveau du message

– Par l'utilisation d'un MAC (Message Authentication Code) généré à l'aide d'un cryptosystème à clé symétrique où le MAC est constitué des derniers digits de C (figure 2.11), ou généré à l'aide d'une fonction de hachage (figure 2.12), la clé secrète K utilisée étant partagée par les deux entités A et B. Dans les deux cas, l'authentification repose sur l'utilisation de la clé K.

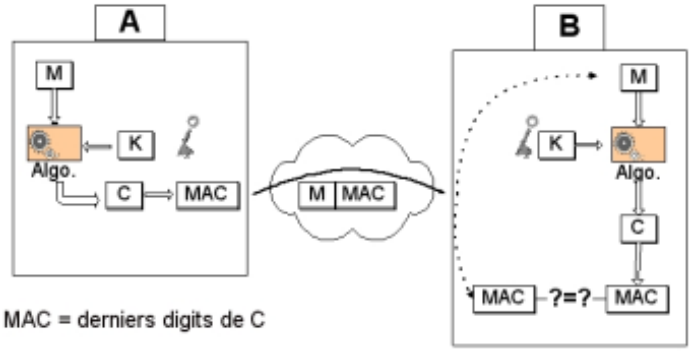


FIG. 2.11 – Authentification par MAC et système symétrique

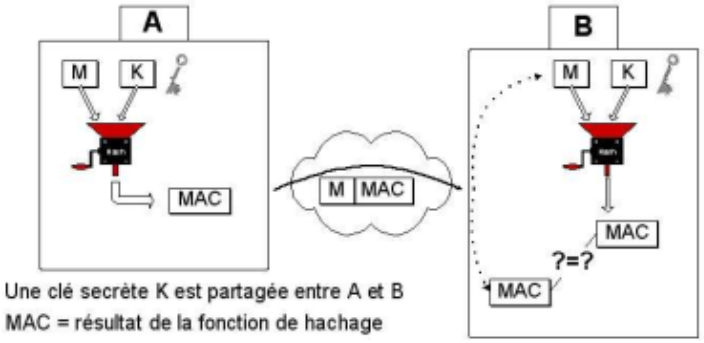


FIG. 2.12 – Authentification par MAC et fonction de hachage

– Par l'utilisation d'une signature digitale. Parmi les propriétés remarquables de ces signatures, on peut dire qu'elles doivent être authentiques, infalsifiables, non-réutilisables, non-répudiables, et inaltérables. Dans la figure 2.13, on fait abstraction de la confidentialité. C'est l'authentification qui importe.

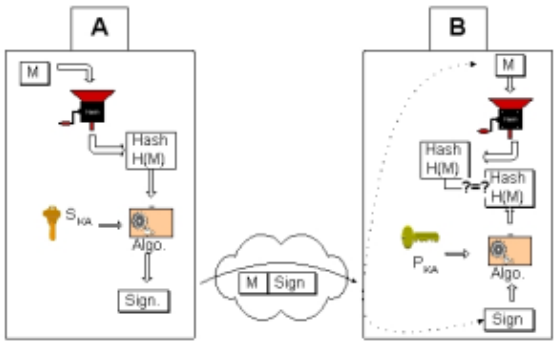


FIG. 2.13 – Authentification par signature (technique asymétrique)

2.5.4.4 Synthèse

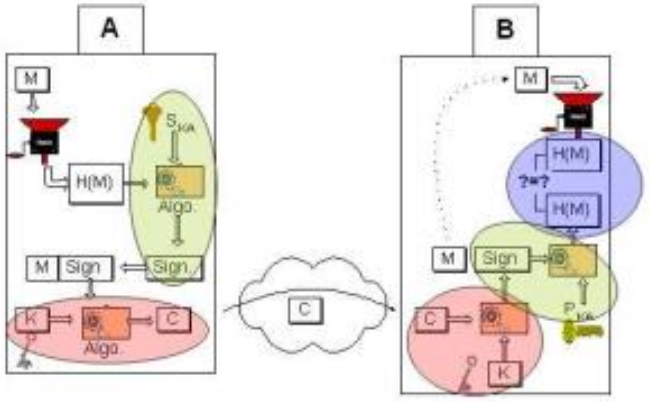


FIG. 2.14 – Confidentialité(Rouge), Intégrité(Violet), Authentification(Vert)

5.2 LA CRYPTOGRAPHIE CLASSIQUE

5.2.1 Domaine de cryptographie classique

Dans le schéma ci-dessous figurent les différentes branches de la cryptographie classique.

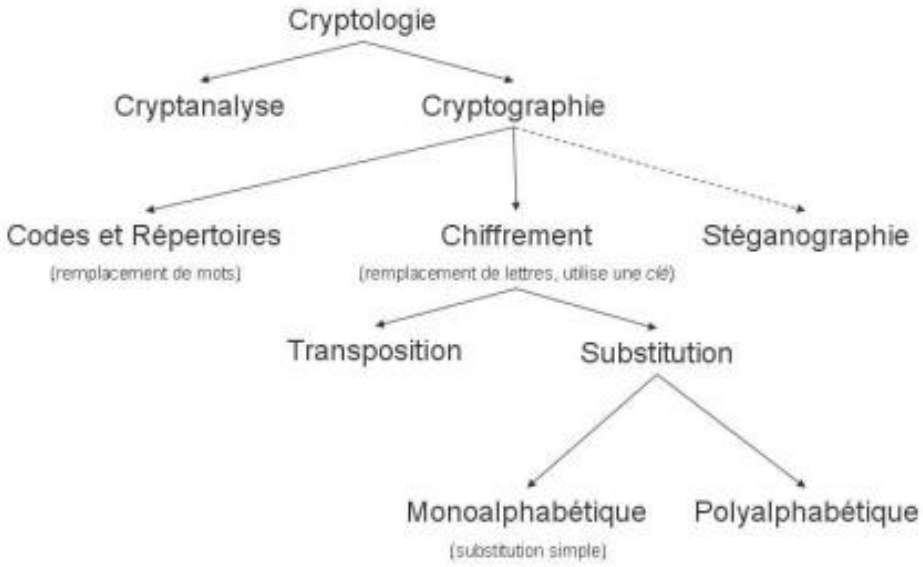


FIG. 3.1 – Domaines inclus dans la cryptologie.

3.1 Substitution monoalphabétique

Chaque lettre est remplacée par une autre lettre ou symbole. Parmi les plus connus, on citera le chiffre de César, le chiffre affine, ou encore les chiffres désordonnés. Tous ces chiffres sont sensibles à l'analyse de fréquence d'apparition des lettres (nombre de fois qu'apparait une même lettre dans un texte). De nos jours, ces chiffres sont utilisés pour le grand public, pour les énigmes de revues ou de journaux.

Historiquement, on recense des procédés de chiffrement remontant au Xème siècle avant JC. On trouve par exemple, l'Atbash des Hébreux (-500), la scytale à Sparte (-400), le carré de Polybe (-125), ... Des langues anciennes sont également parfois classifiées dans les codes secrets : le Rongo-Rongo, le linéaire A, les écritures du disque de Phaistos en sont des exemples. Intraduisibles à l'heure actuelle, on les place (à tort ?) dans ce domaine.

3.1.2 Analyse de fréquences

Lorsque la langue de départ et la technique de chiffrement sont connus, on peut exploiter les régularités du langage par le principe d'analyse de la fréquence d'une lettre. Cette technique ne fonctionne bien que si le message chiffré est suffisamment long pour avoir des moyennes significatives.

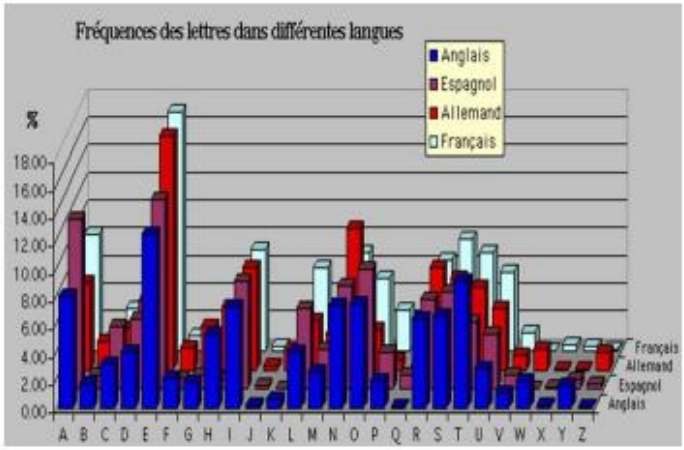


FIG. 3.2 – Exemple d'analyse de fréquence

3.1.3 Chiffre affine

On dit qu'une fonction est affine lorsqu'elle est de la forme $x \rightarrow a * x + b$, c'est-à-dire un polynôme de degré 1. Une fonction linéaire est une fonction affine particulière.

L'idée est d'utiliser comme fonction de chiffrement une fonction affine du type

$$y = (ax + b) \bmod 26,$$

où a et b sont des constantes, et où x et y sont des nombres correspondant aux lettres de l'alphabet (A=0,B=1,...). On peut remarquer que si $a = 1$, alors on retrouve le chiffre de César où b est le décalage (le k du chiffre de César).

Propriété de neutralité : si $b = 0$, alors "a" est toujours chiffré "A" car il ne subit aucun décalage. En effet, si aucun décalage n'a lieu, l'alphabet de départ se retrouve chiffré par lui même, et donc ne subit aucune modification.

3.2 Chiffrement polygraphique

Il s'agit ici de chiffrer un groupe de n lettres par un autre groupe de n symboles. On citera notamment le chiffre de Playfair et le chiffre de Hill. Ce type de chiffrement porte également le nom de *substitutions polygrammiques*.

3.2.1 Chiffre de Playfair (1854)

On chiffre 2 lettres par 2 autres. On procède donc par digramme. On dispose les 25 lettres de l'alphabet (W exclu car inutile à l'époque, on utilise V à la place) dans une grille de 5x5, ce qui donne la clef. La variante anglaise consiste à garder le W et à fusionner I et J.

Il y a 4 règles à appliquer selon les deux lettres à chiffrer lors de l'étape de substitution. Pour le déchiffrement, on procède dans l'ordre inverse.

1. Si les lettres sont sur des "coins", les lettres chiffrées sont les 2 autres coins.
   Exemple : OK devient VA, RE devient XI ...
2. Si les lettres sont sur la même ligne, il faut prendre les deux lettres qui les suivent immédiatement à leur droite.
3. Si les lettres sont sur la même colonne, il faut prendre les deux lettres qui les suivent immédiatement en dessous.
4. Si elles sont identiques, il faut insérer une *nulle* (habituellement le X) entre les deux pour éliminer ce doublon.
   Exemple : "balloon" devient "ba" "lx" "lo" "on".

Pour former ces grilles de chiffrement, on utilise un mot-clef secret pour créer un alphabet désordonné avec lequel on remplit la grille ligne par ligne. Ensuite, on comble la grille avec les lettres restantes de l'alphabet.

FIG. 3.5 – Exemple du chiffre de Playfair

3.2.2 Chiffre de Hill (1929)

Les lettres sont d'abord remplacées par leur rang dans l'alphabet. Les lettres P_k et P_{k+1} deviennent C_k et C_{k+1}

$$\begin{pmatrix} C_k \\ C_{k+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} P_k \\ P_{k+1} \end{pmatrix} (\text{mod} 26)$$

Les composantes de cette matrice doivent être des entiers positifs. De plus la matrice doit être inversible dans Z_{26}. Cependant, sa taille n'est pas fixée à 2. Elle grandira selon le nombre de lettres à chiffrer simultanément.

Chaque digramme clair (P_1 et P_2) sera chiffré (C_1 et C_2) selon :

$$C_1 \equiv aP_1 + bP_2 (\text{mod} 26)$$
$$C_2 \equiv cP_1 + dP_2 (\text{mod} 26)$$

3.3 Substitutions polyalphabétiques

3.3.1 Chiffre de Vigenère (1568)

C'est une amélioration décisive du chiffre de César. Sa force réside dans l'utilisation non pas d'un, mais de 26 alphabets décalés pour chiffrer un message. On parle du *carré de Vigenère*. Ce chiffre utilise une clef qui définit le décalage pour chaque lettre du message (A : décalage de 0 cran, B : 1 cran, C : 2 crans, ..., Z : 25 crans).

Exemple : chiffrer le texte **"CHIFFRE DE VIGENERE"** avec la clef **"BACHELIER"** (cette clef est éventuellement répétée plusieurs fois pour être aussi longue que le texte clair)

Clair	C	H	I	F	F	R	E	D	E	V	I	G	E	N	E	R	E
Clef	B	A	C	H	E	L	I	E	R	B	A	C	H	E	L	I	E
Décalage	1	0	2	7	4	11	8	4	17	1	0	2	7	4	11	8	4
Chiffré	D	H	K	M	J	C	M	H	V	W	I	I	L	R	P	Z	I

FIG. 3.8 – Application du carré de Vigenère

□

La grande force du chiffre de Vigenère est que la même lettre sera chiffrée de différentes manières d'où perte de la fréquence des lettres, ce qui rend inutilisable l'analyse de fréquence classique. La figure 3.9 illustre cette perte des fréquences dans une fable de Lafontaine, codée par substitution simple et par Vigenère.

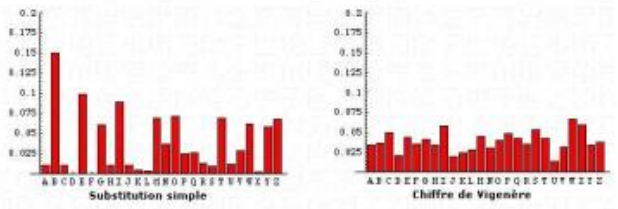


FIG. 3.9 – Perte de la fréquence des lettres

3.3.2 Chiffre de Vernam (One Time Pad - 1917)

Le masque jetable est défini comme un chiffre de Vigenère avec la caractéristique que la clef de chiffrement a la même longueur que le message clair.

Clair	M	A	S	Q	U	E	J	E	T	A	B	L	E
Clef	X	C	A	A	T	E	L	P	R	V	G	Z	C
Décalage	23	2	0	0	19	4	11	15	17	21	6	25	2
Chiffré	J	C	S	Q	N	I	U	T	K	V	H	K	G

FIG. 3.14 – Exemple de One Time Pad

Pour utiliser ce chiffrement, il faut respecter plusieurs propriétés :

- choisir une clef aussi longue que le texte à chiffrer,
- utiliser une clef formée d'une suite de caractères aléatoires,
- protéger votre clef,
- ne jamais réutiliser une clef.

Soit le masque jetable possible : **bgfbcdfbfdecdg**

Résultat : BONJOURLATERRE

Soit un autre masque jetable : **quauwtedbdisjg**

Résultat : MASQUESJETABLE

Il est donc impossible de déterminer le bon masque !

□

Le système du masque jetable, avec les précautions indiquées ci-dessus, est absolument inviolable si l'on ne connaît pas la clef. Il est couramment utilisé de nos jours par les États. En effet, ceux-ci peuvent communiquer les clefs à leurs ambassades de manière sûre via la valise diplomatique.

Le problème de ce système est de communiquer les clefs de chiffrement ou de trouver un algorithme de génération de clef commun aux deux partenaires.

De plus, la création de grandes quantités des clefs aléatoires devient vite problématique. N'importe quel système couramment utilisé pourrait exiger des millions de caractères aléatoires de façon régulière.

La distribution des clés est également complexe. La longueur de la clé étant égale à celle du message, une bonne organisation est nécessaire.

3.4 Transpositions

Elles consistent, par définition, à changer l'ordre des lettres. C'est un système simple, mais peu sûre pour de très brefs messages car il y a peu de variantes. Ainsi, un mot de trois lettres ne pourra être transposé que dans 6 (=3!) positions différentes. Par exemple, "col" ne peut se transformer qu'en "col", "clo", "ocl", "olc", "lco" et "loc".

Lorsque le nombre de lettres croît, il devient de plus en plus difficile de retrouver le texte original sans connaître le procédé de brouillage. Ainsi, une phrase de 35 lettres peut être disposée de $35! = 10^{40}$ manières différentes. Ce chiffrement nécessite un procédé rigoureux convenu auparavant entre les parties.

Une transposition rectangulaire consiste à écrire le message dans une grille rectangulaire, puis à arranger les colonnes de cette grille selon un mot de passe donné (le rang des lettres dans l'alphabet donne l'agencement des colonnes).

Exemple :

A la figure 3.15, on a choisi comme clef GRAIN pour chiffrer le message SALUT LES PETITS POTS. En remplissant la grille, on constate qu'il reste deux cases vides, que l'on peut remplir avec des nulles (ou pas, selon les désirs des correspondants).

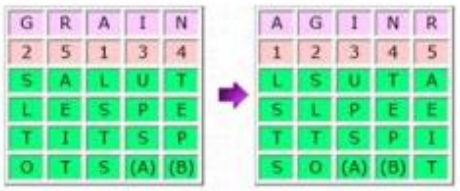


FIG. 3.15 – Application d'une transposition

5.2 LE CHIFFREMENT PAR CLE PUBLIQUE

7.1 Concept

Dans le cas des systèmes symétriques, on utilise une même clé pour le chiffrement et le déchiffrement. Le problème repose dans la transmission de la clé : il faut une clé par destinataire. Dans le cas des systèmes asymétriques, chaque personne possède 2 clés distinctes (une privée, une publique) avec impossibilité de déduire la clé privée à partir de la clé publique. De ce fait, il est possible de distribuer librement cette dernière.

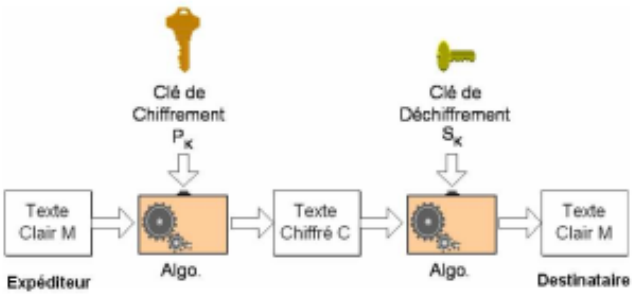


FIG. 7.1 – Chiffrement à clé publique

On peut classer l'utilisation des algorithmes à clé publique en 3 catégories :
- Chiffrement/déchiffrement : cela fournit le secret.
- Signatures numériques : cela fournit l'authentification.
- Échange de clés (ou des clefs de session).

Quelques algorithmes conviennent pour tous les usages, d'autres sont spécifiques à un d'eux.

Le concept date officiellement de 1976 de Diffie et Hellman. Officieusement, les bases existent depuis 1969 par Ellis. La première implémentation a lieu en 1978 par Rivest, Shamir et Adleman sous la forme de l'algorithme RSA bien que, là aussi, les fondements de ce système datent de 1973, par Cocks.

La sécurité de tels systèmes repose sur des problèmes calculatoires :

- RSA : factorisation de grands entiers
- ElGamal : logarithme discret
- Merkle-Hellman : problème du sac à dos (knapsacks)
- ...

La recherche des clés par force brute est toujours théoriquement possible mais les clefs utilisées sont trop grandes (> 512bits). La sécurité se fonde sur une assez grande différence en termes de difficulté entre les problèmes faciles (déchiffrement) et difficiles (décryptement) :
- généralement le problème difficile est connu, mais il est trop complexe à résoudre en pratique,
- La génération des clés exige l'utilisation de très grands nombres.

En conséquence, ce type de chiffrement est lent si on le compare aux chiffrements symétriques.

7.2 Merkle-Hellman

7.2.1 Définition du problème

Soit un havresac de capacité T et un ensemble S d'objets occupant les espaces $S = \{a_1, a_2, a_3, ..., a_n\}$. Il faut trouver un vecteur de sélection $V = \{v_1, v_2, v_3, ..., v_n\}$ satisfaisant la relation $\sum(a_i * v_i) = T$.

Ce problème n'a pas toujours une solution. Aussi, si T et S sont très grands, il est beaucoup plus difficile de trouver le vecteur associé.

Exemple : Soit S = {17, 38, 73, 4, 11, 1} et T = 53 = 38 + 4 + 11. Donc V = {0, 1, 0, 1, 1, 0}.
Pour T = 45, il n'y a pas de solution.

7.2.2 Idée de base

Un bloc de texte clair de longueur égale au nombre d'objets d'un tas sélectionnerait des objets. Les bits du texte clair correspondraient aux valeurs des v_i : un 1 signifierait que l'objet est présent et 0 un objet absent. Et le texte chiffré serait la somme résultante.

Exemple :

M :	1	1	1	0	0	1	0	1	0	1	1	0
Tas :	1	5	6	11	14	20	1	5	6	11	14	20
C :	1+5+6+20 = 32						5+11+14 = 30					

7.2.3 Les empilements

Il y a 2 problèmes d'empilement :
- 1 soluble en temps linéaire
- 1 soluble en temps exponentiel

L'empilement facile peut être transformé pour créer un empilement difficile. Pour la clé publique, on utilisera un empilement difficile qui servira à chiffrer. La clé privée quant à elle, utilisera un empilement facile, qui donne un moyen simple de déchiffrer les messages. Bien sûr, ceux qui ne connaissent pas la clé privée sont obligés de résoudre le problème d'empilement difficile, ce qui est infaisable en pratique.

7.2.3.1 Empilement facile

Si la liste des poids est super-croissante[1], on utilise un algorithme (appelé *glouton*) de la manière suivante :

1. Prendre le poids total et le comparer avec le plus grand nombre de la suite.

[1] Une liste est super-croissante lorsque tout terme est plus grand que la somme des termes qui le précède.

   - Si le poids total est inférieur à ce nombre, alors celui-ci n'est pas dans le tas. On recommence l'opération avec le nombre suivant dans le tas (qui, par définition de la suite, sera plus petit)
   - Si le poids total est supérieur à ce nombre, alors celui-ci est dans le tas
2. Réduire le poids du tas à créer de ce nombre et passer au plus grand nombre suivant de la suite.
3. Répéter jusqu'à ce que ce soit terminé.
4. Si le poids total a pu être ramené à 0 : il y a une solution.

7.2.3.2 Empilement difficile

Dans le cas présent, on ne connaît pas d'algorithme rapide. Il faut tester méthodiquement toutes les solutions possibles, ce qui, si la suite des poids est suffisamment longue, est impraticable. Ces algorithmes sont exponentiels.

Le cryptosystème de Merkle-Hellman exploite cette propriété. La clé privée est une suite de poids super-croissante. A partir de celle-ci, on calcule la clé publique. Ce calcul consiste à prendre la suite super-croissante, et à la multiplier par (n modulo m), avec m supérieur à la somme de tous les termes de la suite, et n ne devant avoir aucun facteur commun avec m.

7.2.4 Algorithme

Le chiffrement consiste à additionner les termes où un 1 apparaît. Pour le déchiffrement, on calcule n^{-1} tel que

$$n * n^{-1} \equiv 1 \bmod m$$

Ensuite, on multiplie chaque valeur du texte chiffré par n^{-1} mod m.

En pratique, les sacs contiennent environ 250 éléments. Et chaque terme a une longueur de 200 à 400 bits. Le module a une longueur de 100 à 200 bits.

Exemple de calcul de la clé publique Soit S, une séquence super-croissante de h entiers : par exemple $S = \{1, 2, 4, 9\}$.

Choisissons un multiplicateur n et un module m : soit $n = 15$ et $m = 17$

- $1 * 15 \bmod 17 \Rightarrow 15$
- $2 * 15 \bmod 17 \Rightarrow 13$
- $4 * 15 \bmod 17 \Rightarrow 9$
- $9 * 15 \bmod 17 \Rightarrow 16$

Le havresac difficile est donc H = $\{15, 13, 9, 16\}$, et représente la clé publique.

Le message est ainsi traité comme une séquence de bits :

$$P = [p_1, p_2, p_3, ..., p_k]$$

On le divise en blocs de h bits :

$$P_0 = [p_1, p_2, p_3, ..., p_h], P_1 = [p_{h+1}, p_{h+2}, p_{h+3}, ..., p_{2*h}], ...$$

On utilise chaque bloc comme vecteur V du problème de havresac.

Exemple de chiffrement Soit P = 0100101110100101 $\Rightarrow$ 0100 1011 1010 0101

- $[0, 1, 0, 0] * [15, 13, 9, 16] \Rightarrow 13$
- $[1, 0, 1, 1] * [15, 13, 9, 16] \Rightarrow 40$
- $[1, 0, 1, 0] * [15, 13, 9, 16] \Rightarrow 24$
- $[0, 1, 0, 1] * [15, 13, 9, 16] \Rightarrow 29$

Le message chiffré est donc $\{13, 40, 24, 29\}$ en utilisant le havresac public (la clef publique) H = [15, 13, 9, 16].

□

Pour le déchiffrement, le destinataire légitime connaît le havresac simple S et les valeurs de n et de m. Il peut donc déterminer n^{-1}.

Exemple de déchiffrement Avec $n = 15$ et $m = 17$, n^{-1} vaut 8 car $15 * 8 = 120 = 7 * 17 + 1$. On a alors, par l'algorithme glouton :

- $13 * 8 \bmod 17 = 104 \bmod 17 = 2 = [1,2,4,9] * [0100]$
- $40 * 8 \bmod 17 = 320 \bmod 17 = 14 = [1,2,4,9] * [1011]$
- $24 * 8 \bmod 17 = 192 \bmod 17 = 5 = [1,2,4,9] * [1010]$
- $29 * 8 \bmod 17 = 232 \bmod 17 = 11 = [1,2,4,9] * [0101]$

et le texte en clair est 0100 1011 1010 0101 $\Rightarrow$ 0100101110100101. On a donc bien retrouvé le texte original.

□

7.2.5 Sécurité

Plusieurs failles ont été découvertes, qui ont rendu l'algorithme obsolète.

Herlestam a démontré en 1978 qu'un bit de texte clair pouvait souvent être retrouvé.

Malgré les modifications apportées à l'algorithme à la suite de ces découvertes, des travaux de Shamir (1982-84) et Brickell (1985) ont poussé à l'abandon de cet algorithme.

7.3 RSA : Rivest - Shamir - Adleman

Il est basé sur le calcul exponentiel. Sa sécurité repose sur la fonction unidirectionnelle suivante : le calcul du produit de 2 nombres premiers est aisé. La factorisation d'un nombre en ses deux facteurs premiers est beaucoup plus complexe.

Il s'agit du système le plus connu et le plus largement répandu, basé sur l'élévation à une puissance dans un champ fini sur des nombres entiers modulo un nombre premier. Le nombre d'exponentiation prend environ $O((\log n)^3)$ opérations ce qui est rapide et facile. Il emploie de grands nombres entiers (par exemple représentés sur 1024 bits).

Ce cryptosystème utilise deux clés d et e, interchangeables[2]. Le chiffrement se fait selon

$$C = M^e \bmod n$$

et le déchiffrement par

$$M = C^d \bmod n.$$

La sécurité repose sur le coût nécessaire pour factoriser de grands nombres. Le nombre de factorisation prend environ $O(e^{\log n \log(\log n))})$ opérations ce qui demande un temps de calcul trop important pour les machines actuelles, dans un cadre privé. On l'utilise pour la confidentialité, l'authentification, ou encore une combinaison des 2.

7.3.1 Principes

On possède une paire de clés, l'une publique (e,n) et une privée (d,n). La première étape revient à choisir n. Il doit s'agir d'une valeur assez élevée, produit de 2 nombres premiers très grands p et q. En pratique, si p et q ont 100 chiffres décimaux, n possèdera 200 chiffres. Selon le niveau de sécurité souhaité, la taille de n peut varier : 512 bits, 768, 1024 ou 2048[3].

Dans un second temps, on choisira un très grand entier e, relativement premier à (p-1)*(q-1). La clé publique sera formée par (e,n). On choisira ensuite un d tel que

$$e * d \equiv 1 \bmod (\Phi(n)).$$

La clé privée sera donnée par (d,n).

Dernière phase : on jette p et q. Le cryptanalyste devant retrouver ces valeurs, il faut les détruire pour éviter les fuites.

7.3.1.1 Justification de l'inversibilité

Par les théorèmes d'Euler et de Fermat, on sait que

$$a^{\Phi(n)} \equiv 1 \bmod n$$

et

$$a^{\Phi(n)} \bmod n = 1$$

où (a,n)=1.

Dans le RSA, on a $n = p * q$. De plus, $\Phi(n)$ donne le nombre d'entiers positifs plus petits que n et relativement premiers à n (si p est premier, $\Phi(p) = p - 1$). Si $n = p * q$, avec p et q premiers, il vient

$$\Phi(n) = \Phi(p) * \Phi(q) = (p - 1) * (q - 1)$$

De par la façon de choisir e et d,

$$e * d \equiv 1 \bmod \Phi(n) = k * \Phi(n) + 1$$

pour un certain k.

De plus, par définition et propriétés des opérations modulo n, on a :

$$D_k(E_k(M)) = ((M)^e \bmod n)^d \bmod n = (M^e)^d \bmod n = M^{e*d} \bmod n$$

Et donc :

$$M^{e*d} = M^{k*\Phi(n)+1} = M^{k*\Phi(n)}.M \bmod n = 1.M \bmod n = M \bmod n$$

[3]Le FBI utilise le RSA 4096.

7.3.2 Résumé

1. Génération de 2 nombres premiers p et q
2. Calcul de n = p*q
3. Déterminer e tel que $3 < e < \Phi(n)$ et $(e, \Phi(n)) = 1$
4. Calculer d tel que $e * d \equiv 1 \bmod \Phi(n)$
5. Clé publique : (e,n)
6. Clé privée : (d,n)
7. p et q doivent rester secrets, voire supprimés
8. $C = M^e \bmod n$ et $M = C^d \bmod n$

Exemple :

Soient p = 31, q = 53 c'est-à-dire n=1643. $\Phi(n) = 1560$ (nombre d'éléments relativement premiers à n et < n).
Soit e = 11 (par exemple, et on a bien (e,$\Phi(n)$)=1).
On détermine que d = 851 (inverse modulaire de e sur $Z_{\Phi(n)}$).
La clé publique est donc (11,1643) et la clé privée est (851,1643).

Soit le codage par la position dans l'alphabet du mot «ANEMONE». Il vient

01 14 05 13 15 14 05

On procède selon deux conditions :

1. Découpage en morceaux de même longueur, ce qui empêche la simple substitution :

   011 405 131 514 05_

   On ajoute un padding initial si nécessaire.

   001 140 513 151 405

   Cela provoque la perte des patterns (« NE »).

2. Découpage en morceaux de valeur inférieure à n, car opération modulo n.

Lors du chiffrement, on a

$001^{11} \bmod 1643$	0001
$140^{11} \bmod 1643$	0109
$513^{11} \bmod 1643$	0890
$151^{11} \bmod 1643$	1453
$405^{11} \bmod 1643$	0374

et pour le déchiffrement,

$0001^{851} \bmod 1643$	001
$0109^{851} \bmod 1643$	140
$0890^{851} \bmod 1643$	513
$1453^{851} \bmod 1643$	151
$0374^{851} \bmod 1643$	405

Lors du déchiffrement, sachant qu'il faut obtenir des blocs de 2 éléments (grâce au codage particulier de l'exemple), on a bien

01	14	05	13	15	14	05
A	N	E	M	O	N	E

7.3.2.1 Remarques

Il n'est pas très astucieux de choisir d'aussi petites valeurs car on peut retrouver d très facilement.

En pratique, il faut prendre de très grandes valeurs de p et q. Pour retrouver ces grandes valeurs, il faudra alors utiliser le Jacobien et le test de Solovay-Strassen par exemple.

7.3.3 Sécurité

7.3.3.1 Attaques

Il existe trois approches pour attaquer le RSA :

- recherche par force brute de la clé (impossible étant donné la taille des données),
- attaques mathématiques (basées sur la difficulté de calculer $\Phi(n)$, la factorisation du module n) :
  - factoriser n=p*q et par conséquent trouver $\Phi(n)$ et puis d,
  - déterminer $\Phi(n)$ directement et trouver d,
  - trouver d directement.
- attaques de synchronisation (sur le fonctionnement du déchiffrement).

A l'heure actuelle, la factorisation connait de lentes améliorations au cours des années. La meilleure amélioration possible reste l'optimisation des algorithmes. Excepté un changement dramatique, le RSA-1024 restera sûr pour les prochaines années. D'après les projections, une clé de 2048 bits est sensée tenir jusque 2079 si on tient compte de la loi de Moore. Mais ces valeurs sont correctes uniquement si on respecte les propriétés de e, d, p et q.

Attaque de synchronisation (timing attack)

Développé dans le milieu des années 90, il s'agit d'exploiter les variations de temps pris pour effectuer certaines opérations (par exemple la multiplication par un petit ou un grand nombre).

Plusieurs contre-mesures existent telles que l'emploi de temps constants d'élévation à une puissance, l'ajout de délais aléatoires, ou le fait de rendre non visibles les valeurs utilisées dans les calculs. Dans ce dernier cas, cela reviendrait à calculer :

$$(r^e * m^e)d \bmod n$$

7.3.3.2 La menace quantique

Les valeurs précitées sont valables si on pratique la factorisation. A coté de cela, la physique pourrait faire pencher la balance, par l' utilisation d'un ordinateur quantique[4]. Celui-ci existe d'un point de vue théorique depuis 1994 (algorithme de Shor), et son prototype depuis 1996. Si son évolution se poursuit, il permettrait de réaliser la factorisation d'un nombre en un temps polynomial. Le principe est que les 0 et 1 représentés par les portes logiques des transistors sont remplacés par l'orientation du champ magnétique émit par les atomes (que l'on nomme des q-bits).

7.3.4 Conseils d'utilisation du RSA

Pour garantir une bonne sécurité, il faut respecter certains règles telles que :

- Ne jamais utiliser de valeur n trop petite,
- N'utiliser que des clés fortes (p-1 et q-1 ont un grand facteur premier),
- Ne pas chiffrer de blocs trop courts,
- Ne pas utiliser de n communs à plusieurs clés,
- Si (d,n) est compromise ne plus utiliser n.

Level	Protection	Symmetric	Asymmetric
1	Attacks in "real-time" by individuals *Only acceptable for authentication tag size*	32	-
2	Very short-term protection against small organizations *Should not be used for confidentiality in new systems*	64	816
3	Short-term protection against medium organizations, medium-term protection against small organizations	72	1008
4	Very short-term protection against agencies, long-term protection against small organizations *Smallest general-purpose level, protection from 2008 to 2010*	80	1248
5	Legacy standard level *Use of 2-key 3DES restricted to 10^6 plaintext/ciphertexts, protection from 2008 to 2016*	96	1776
6	Medium-term protection *protection from 2008 to 2026*	112	2432
7	Long-term protection *Generic application-independent recommendation, protection from 2008 to 2036*	128	3248
8	"Foreseeable future" *Good protection against quantum computers*	256	15424

FIG. 7.2 – Taille des clés pour une utilisation sûre.

Un concours, connu sous le nom de "RSA Factoring Challenge", proposait une certaine somme d'argent à tout groupe ayant réussi la factorisation d'une clé de taille donnée (la récompense étant proportionnelle à la taille de la clé mise en défaut). La plus grande clé "cassée" atteignit 663 bits (2005). Le concours fut stoppé fin 2007.

5.4 GESTION DE CLES

La gestion des clés est principalement constituée de quatre domaines :

1. La génération des clés : il faut prendre garde aux caractères choisis, aux clés faibles, ... et veiller à utiliser des générateurs fiables,
2. Le transfert de la clé : l'idéal est de se rencontrer, ou d'utiliser un canal de transmission protégé. Mais cela est souvent impossible. Aussi, si A et B ont des communications sûres avec un tiers C, ce dernier peut relayer la clé entre A et B. Un tiers est un intermédiaire de confiance, à qui tous les usagers font confiance, pour négocier l'établissement de transmissions sûres entre elles,
3. La vérification des clés : par hachage, ou utilisation de certificats,
4. Le stockage des clés : que ce soit dans des fichiers, sur supports extérieurs, par surchiffrement, ...

10.1 Distribution des clés

10.1.1 Clés symétriques

Dans ce cas, il est nécessaire pour les deux usagers de partager une clé secrète commune. Bien souvent, l'échec d'un système sûr est dû à une rupture dans le schéma de distribution des clés. Comment distribuer sûrement cette clé ?

- Physiquement : par une rencontre, un canal de transmission protégé, ...
- Utiliser un tiers de confiance. Celui-ci choisit et fournit la clé,
- Utiliser une ancienne clé pour chiffrer une nouvelle clé (ce qui suppose cependant un échange préalable de cette ancienne clé),
- Distribution automatique de clés à la demande des utilisateurs. Cette solution existe, mais elle nécessite une totale confiance au système.

La figure 10.1 illustre le protocole Needham-Schroeder, dont nous reparlerons dans le chapitre consacré à l'authentification des parties. Ce scénario suppose que les deux entités (A et B) possèdent chacune une clé secrète avec le KDC (l'autorité de confiance du système).

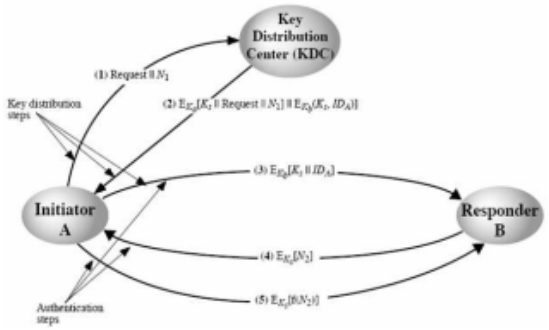


FIG. 10.1 – Exemple d'une distribution de clés dans le cas symétrique (Needham-Schroeder).

10.1.2 Clés asymétriques

Le chiffrement par clé publique permet de résoudre les problèmes de distribution de clés secrètes. Malgré tout, la distribution des clés publiques continue de poser problème, principalement au niveau de l'authentification des utilisateurs liés à ces clés. Il existe quatre solutions permettant un transfert des clés dans le cas asymétrique :

- Annonce publique
- Annuaire publiquement disponible
- Autorité de clés publique
- Certificats de clé publique

10.1.2.1 Annonce Publique

La distribution des clés publiques se fait directement aux destinataires ou par broadcast à la communauté. Il est par exemple possible d'apposer les clefs PGP aux emails ou les poster dans des newsgroups ou mailing-lists. Mais le risque majeur avec cette méthode est la contrefaçon : n'importe qui peut créer une clef en prétendant être quelqu'un d'autre et la publier. La mascarade continuera tant que la contrefaçon n'est pas découverte.

10.1.2.2 Annuaire Publique

On enregistre ici des clés dans un annuaire public, ce qui implique de faire confiance à cet annuaire. Ce dernier doit avoir plusieurs propriétés :

- Il doit contenir les entrées {nom, clef publique},
- Il doit être possible de s'inscrire de manière sécurisée dans l'annuaire,
- On doit pouvoir remplacer la clef à tout moment,
- L'annuaire doit être publié périodiquement,
- Il devrait également permettre la consultation électronique.

Même si ce schéma est clairement plus sûr que les annonces publiques individuelles, il reste vulnérable. Il est en effet nécessaire que l'annuaire soit sécurisé. Dans le cas contraire, un individu pourrait détourner l'annuaire et fournir des clefs publiques contrefaites, voire à ne pas transmettre les clés correspondant aux demandes des entités communicantes.

10.1.2.3 Autorité de clés publique

Il s'agit de renforcer le contrôle de la distribution des clefs à partir de l'annuaire. Il dispose des mêmes propriétés que ce dernier. Cependant, la sécurité est renforcée. En effet, dans le cas présent, chaque entité dispose de la clé publique de l'autorité. Ainsi, lorsqu'une entité désirera obtenir la clé publique d'un correspondant, il enverra une requête à l'autorité. Celle-ci contiendra la requête proprement dite et un marqueur temporel (*timestamp*). En retour, l'autorité renverra la clé demandée, le timestamp pour prouver le non-rejeu d'un ancien message, le tout chiffré avec sa clé privée. De cette manière, l'entité A, possédant la clé publique de l'autorité, pourra vérifier la bonne provenance de la clé publique de B. L'entité B pourra pratiquer de la même manière.

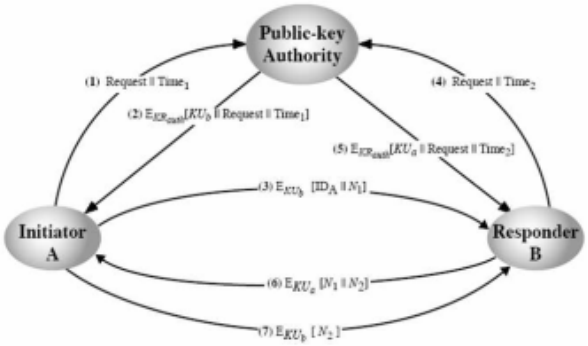


FIG. 10.2 – Utilisation d'une autorité de clés publiques

10.1.2.4 Certificats de clés publiques

Cette approche alternative (1978) consiste à employer des certificats pouvant être utilisés par des participants pour échanger des clefs sans entrer en contact avec une autorité de clés publiques, d'une façon fiable comme si les clefs avaient été obtenues directement à partir d'une telle autorité.

Chaque certificat contient une clef publique et des informations supplémentaires (la période de validité, les droits d'utilisation, etc.). Il est créé par une autorité de certification (*Certificate Authority*, CA) et est donné au participant disposant de la clef privée assortie. Un participant fournit l'information sur sa clé à un autre en transmettant son certificat. Son contenu est signé par la clé privée du CA. Cette situation est illustrée à la figure 10.3

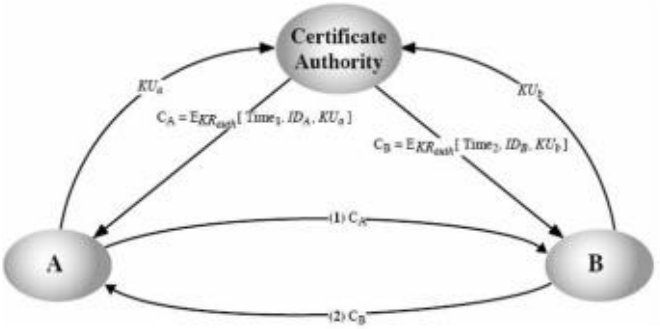


FIG. 10.3 – Certificats de clés publiques

Plusieurs propriétés permettent de garantir une meilleure sécurité :
- N'importe quel participant peut lire un certificat pour déterminer le nom et la clé publique du propriétaire du certificat.
- N'importe quel participant peut vérifier que le certificat provient réellement du CA et n'est pas contrefait.
- Seul le CA peut créer et mettre à jour des certificats.
- Tous les participants peuvent vérifier la validité des certificats (Denning - 1983)

Pour mieux comprendre ce concept, illustrons le par un exemple : si vous commandez des cds sur Internet, comment être sûr que vous envoyez bien votre numéro de carte bleue au commerçant, et non à un pirate qui aurait usurpé son identité et donné sa propre clé publique ?

Pour passer un examen, il vous faut prouver votre identité grâce à une carte d'identité, un passeport ou un permis de conduire. Un organisme supérieur (l'Etat) a signé ces certificats, s'assurant auparavant (par un acte de naissance,...) qu'il s'agit bien de vous.

Les certificats numériques fonctionnent sur le même principe. Alice veut certifier que sa clé publique lui appartient. Elle envoie sa clé à un organisme de certification, ainsi que différentes informations la concernant (nom, email, etc...). Cet organisme vérifie les informations fournies par Alice, et ajoute au certificat son propre nom, une date limite de validité, et surtout une signature numérique. Cette signature est réalisée grâce à sa clé privée et à un algorithme de hachage laissé au choix, bien que le RSA et le SHA soit maintenant préféré. En résumé, l'autorité de certification fournit un certificat tel que

$$C_A = E_{KR_{auth}}[T, ID_A, KU_a]$$

où $E_{KR_{auth}}$ représente la signature apposée au certificat, T est un timestamp, ID_A est l'ensemble des informations propres à l'entité A et KU_a est la clé publique de A.

Lorsque Bob veut envoyer un message à Alice ou simplement vérifier la validité du certificat que A lui a envoyé, il applique la clé publique de l'autorité de certification. Cette action permet de vérifier que le certificat est bien authentique :

$$D_{KU_{auth}}[C_A] = D_{KU_{auth}}[E_{KR_{auth}}[T, ID_A, KU_a]] = (T, ID_A, KU_A)$$

10.1.3 Les certificats - Service d'authentification X.509

Il s'agit d'une partie de la norme de service d'annuaire X.500, qui regroupe des serveurs distribués maintenant une base de données d'informations. Le service définit le cadre pour des services d'authentification, internationalement admis pour construire un certificat de clé publique. L'annuaire peut stocker des certificats de clé publique et les clés publiques des utilisateurs correspondants. Ces certificats sont également signés par une autorité de certification. Il utilise la cryptographie à clé publique et les signatures digitales. Aucun algorithme de chiffrement n'est imposé, mais le RSA est recommandé.

10.1.3.1 Obtention d'un certificat

N'importe quel utilisateur ayant accès au CA peut obtenir un certificat de celui-ci, mais seul le CA peut modifier un certificat. Comme il est difficile de forger un certificat, on peut sans trop de risque le placer dans un annuaire public.

La figure 10.4 illustre le format d'un certificat. L'exemple donné concerne un certificat signé par RSA et MD5.

10.1.3.2 Hiérarchie de certificats

On émet ici une hypothèse : si les deux utilisateurs partagent un CA commun alors on suppose qu'ils connaissent sa clef publique.

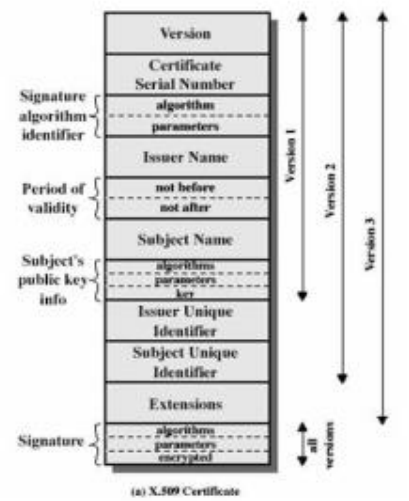


FIG. 10.4 – Format standard et exemple de certificat X.509

Sur la figure 10.5, chaque CA possède des certificats pour les clients (vers l'avant) et le parent (vers l'arrière). Chaque client fait confiance aux certificats parents. On parle alors de hiérarchie de CA. Le principe est d'employer les certificats liant les membres de la hiérarchie pour valider d'autres CA. L'objectif est de permettre la vérification de n'importe quel certificat d'un CA par des utilisateurs de tout autre CA dans la hiérarchie.