

Sécurité et Cryptographie

Fabrice Huet

Sécurité

Principe Généraux

Introduction

- La sécurité informatique s'intéresse à la protection des ordinateurs et des données
- Confidentialité
 - S'assurer que les données ne sont pas visibles par ceux qui n'ont pas l'autorisation
- Intégrité
 - S'assurer que les données ne sont pas modifiées par ceux qui n'ont pas l'autorisation
- Disponibilité
 - S'assurer que personne ne peut interférer avec le fonctionnement d'un ordinateur

Confidentialité

- Comment protéger des données ?
- Droits d'accès aux fichiers
 - Protection très limitée
 - Accès physique à la machine permet d'ignorer les droits
- Cryptage
 - Rendre les données illisibles sans un mot de passe
 - Fourni en standard par beaucoup d'OS
 - Attention au cryptage Windows qui stock les clefs sur le disque...
 - TrueCrypt multi OS

Intégrité des données

- Comment s'assurer qu'une donnée n'a pas été modifiée ?
 - Fiabilité
 - Sécurité
- En faire une copie de sauvegarde et la comparer?
 - Couteux en ressources
 - Inapplicable pour une communication réseau
- Il faut une sorte de « signature » des données
 - Quelque chose de petit qui nous permet de savoir si elles ont été modifiées

Hachage

- Le hachage est une opération qui prend des données de grande taille pour obtenir un ensemble plus petit de taille fixe
 - Exemple : *MD5 hash* produit 128 bits
- Soit k les données et $h()$ la fonction de hachage
 - $h(k)$ est la valeur de hachage de k
- Soient k et k' tels que $k \neq k'$
 - Alors $h(k) \neq h(k')$ avec une très grande probabilité
- Soient $h(k)$ et $h(k')$ tels que $h(k) = h(k')$
 - Alors $k = k'$ avec une très grande probabilité

Hachage et intégrité

- Soit k la donnée à transmettre
 - On calcule $h(k)$
- On transmet la donnée
 - Le receveur reçoit k' et veut savoir si c'est k
 - Il calcule $h(k')$, si c'est égal à $h(k)$, c'est bon
- Problèmes
 - Il faut transmettre $h(k)$ de manière fiable
 - Un ennemi pourrait créer k'' tel que $h(k'')=h(k)$
 - Si $h()$ bien choisi, on montre que c'est impossible en un temps fini

Disponibilité

- Empêcher quelqu'un d'interrompre un service
 - *Denial Of Service (DOS)*
- Ex : un utilisateur envoie 2000 fichiers à l'imprimante, plus personne d'autre ne peut imprimer
- Gérable en local
 - Mettre des limites au niveau de l'OS pour chaque utilisateur
- Difficile en distribué
 - 200000 machines parlent en même temps au même site web et le bloquent...

Menaces Logicielles

Malware/Spyware, Virus, Worm,
Adware

Menaces Logicielles

- Menace Logicielle
 - Programme écrit pour nuire à la sécurité
- Plusieurs catégories
 - Malware
 - Logiciel nuisant à l'utilisateur ou à une machine
 - Virus
 - Worms
 - AdWare
 - Logiciel affichant de la publicité sur la machine

Malware

- Désigne tout logiciel hostile, nuisant à l'utilisateur ou la machine
- Virus et vers sont des Malware
- Souvent utilisés pour prendre le contrôle d'une machine et s'en servir pour
 - Stocker des données illégales
 - Envoyer du spam
 - Attaquer d'autres machines

Adware

- Logiciel affichant de la publicité
 - Pop-up, barre de pub...
- En général installé sans le consentement de l'utilisateur
 - Souvent installé avec un logiciel « utile »
 - Mentionné dans le EULA, au milieu de 2k lignes
- Très difficile à désinstaller
 - Utilise toutes les astuces possibles pour rester sur la machine
- Des logiciels existent pour les désinstaller

Spyware

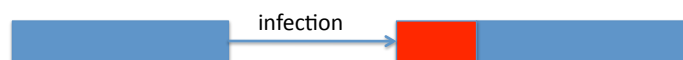
- Logiciel spécialisé dans l'espionnage
- Essaie d'enregistrer des informations sensibles
 - Login/password, numéro de CB, ...
- Keylogger
 - Logiciel enregistrant les frappes clavier
- Solution:
 - Éviter les frappes clavier... (utilisation d'une interface + souris)

Virus

- Programme auto répliquant capable d'infecter des programmes
- Terme souvent utilisé pour n'importe quelle menace logicielle
 - Abus de langage
- Étude théorique
 - Fred Cohen démontre en 1987 qu'il n'est pas possible de détecter **tous** les virus

Infection

- Un virus est un programme
 - et doit donc être exécuté
 - mais personne (?) ne le ferait volontairement
- Il se greffe sur un programme de la machine
 - Quand le programme est exécuté, le virus l'est aussi
 - Il peut ensuite décider d'une stratégie de réplication
 - Il peut écraser une partie du programme, se placer avant, après, au milieu...
 - Intérêt?



Infection - 2

- Une fois le virus exécuté, il peut
 - Se terminer après certaines actions : virus non résident
 - Rester en mémoire : virus résident
- Un virus résident va s'associer à certaines fonctions de l'OS
 - Il sera exécuté à chaque fois
 - Lecture de fichier, exécution de programme...
- Le but du virus est de se propager
 - Infecter de nouveaux programmes
 - Infecter de nouvelles machines

Propagation

- Si le virus est résident, propagation locale facilitée
 - Pourra infecter tout fichier lu, programme exécuté...
 - À priori, ne s'intéresse qu'aux exécutables
- Propagation distante
 - Le virus va chercher à atteindre de nouvelles machines
 - Utilisation des contacts de l'utilisateur (MSN, mail...)
 - Le virus envoie un programme en attachement à un utilisateur
 - Propagation basée sur la confiance

Antivirus

- Un antivirus (AV) est un programme chargé de lutter contre les menaces logicielles
 - Y compris les virus
- Il doit surtout prévenir l'infection
 - Une machine infectée est très difficile à nettoyer
 - L'AV peut lui aussi être infecté et devenir vecteur de propagation
- L'AV utilise plusieurs techniques pour détecter les virus
 - Guerre électronique : chaque technique de l'AV est contrée par une technique du Virus

Détection de Virus

- Par définition, le virus a besoin d'un programme
 - L'AV recherche les programmes modifiés
- Critères (et contre mesure)
 - Date de modification
 - Le virus préserve la date de modification
 - Taille
 - Le virus efface une partie du programme pour préserver sa taille
 - Il peut aussi compresser le programme original pour conserver une taille totale constante
 - Modification du contenu, repérée par hachage
 - Le virus peut effacer la signature des fichiers

Signature d'un virus

- On peut facilement détecter un virus dans un fichier
 - Mais comment savoir de qui il s'agit ?
- Signature
 - Caractéristique unique d'un virus
 - Suite d'instructions ou de bits
- L'AV recherche dans sa base une signature correspondante
 - Ne marche que pour les virus **connus**

Signature d'un virus - 2

- La signature doit être trouvée
 - Travail fourni pas les sociétés d'AV
 - Effectué le plus tôt possible, dès le début de l'infection
- Contre mesure : le virus change sa signature
- Virus polymorphique
 - À chaque nouvelle contagion, le virus modifie son code, tout en préservant son algorithme
 - Cryptage du code, ou réordonnancement des instructions

Détection par heuristique

- Comment détecter des virus inconnus?
 - Pas de signatures connues
 - Mais on connaît en gros leur comportement (infection, propagation...)
- L'AV observe le comportement des programmes
 - Recherche de programmes se comportant « comme un virus »
- Heuristique : Utilisation de règles empiriques, apprentissage par la prise en compte d'activités antérieures
- Peut conduire à des faux positifs
 - Programmes innocents détectés comme virus
 - Recours à l'humain pour valider

Worms (vers)

- Un ver (worm) est similaire à un virus
 - Autonome, n'a pas besoin de programmes
- Internet Worm :
 - Premier ver
 - Lancé le 2 novembre 1988 par Robert T. Morris
- Stratégie
 - Il se connecte à distance aux machines et essaie des mots de passe connus
 - Il utilise des failles (bugs) dans certains programmes
- Réplication excessive
 - Le ver demande à une machine si elle est déjà infectée
 - Même si elle répond oui, 7/10 fois, la réinfecte. Pourquoi?
 - Taux trop élevé, au bout de quelques jours, les machines ne faisaient qu'exécuter le ver.

Conclusion

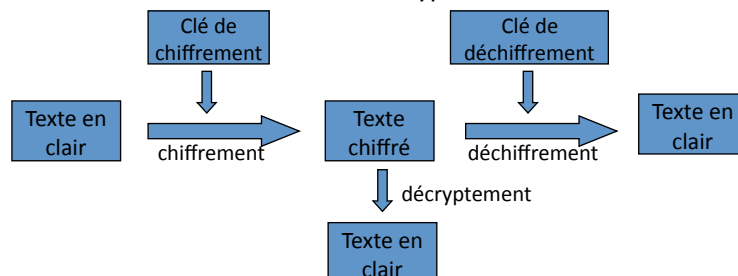
- Nous n'avons vu que très peu de menaces
- Souvent initiées par l'utilisateur lui-même
 - Exécute un fichier sans précaution, sur confiance
- Première solution :
 - Ne rien exécuter dont on ne soit pas certain de la provenance
- Deuxième solution :
 - Ne rien exécuter en étant Administrateur/root
- Troisième solution :
 - Protéger sa machine avec un AV (nombreux gratuits)

Cryptographie

Clef symétrique et Asymétrique, RSA

Introduction

- La cryptographie est l'étude de méthodes permettant de transmettre une information de manière confidentielle
- On applique à un message initial une transformation le rendant incompréhensible, c'est le chiffrement
- Le chiffrement utilise une clé de chiffrement
- Le déchiffrement permet de reconstruire le texte en clair à partir du texte chiffré en utilisant une clé de déchiffrement
- L'opération consistant à trouver le texte en clair à partir du texte chiffré sans utiliser la clé est le décryptement



Introduction - 2

- Principe de Kerckhoff: un ennemi connaît en détail vos algorithmes de cryptographie
- Corollaire: la sécurité d'un système cryptographique ne doit reposer que sur la clé et non sur le secret de l'algorithme
- Si la clé de chiffrement et la clé de déchiffrement sont identiques:
 - chiffrement symétrique ou à clé secrète
- Si elles sont différentes
 - Chiffrement à clé publique clé privée

Attaques classiques

- Attaque exhaustive
 - On essaie toutes les clé possibles
 - Bien adapté au chiffrement symétrique
 - Si l'espace des clé est grand, impossible en pratique
- Attaque par observation du texte chiffré
 - Étude du texte chiffré pour en déduire des informations sur la clé
- Attaque à texte clair connu
 - On compare un texte en clair et sa version chiffrée pour en déduire la clé
- Impersonnation
 - On s'interpose entre les parties désirant communiquer en toute sécurité et on se fait passer pour l'une auprès de l'autre

Code de César

- On applique à chaque lettre une opération de décalage
- La valeur de décalage est la clé
- Algorithme de chiffrement
 - Soit c le caractère en clair, $ce = (c + \text{clé}) \bmod 26$
- Algorithme de déchiffrement
 - $c = (ce - \text{clé}) \bmod 26$
- Exemple: « vive les algos » devient « tgtc jcq yjemq » avec la clé -2
- Codage symétrique
- Problèmes:
 - Seulement 26 clés possibles, au pire on essaie tous les décalages
 - Le chiffrement préserve les propriétés statistiques du texte en clair
 - Si le texte chiffré a 50% de Z, alors on peut en déduire que Z est la lettre E dans le texte en clair et donc trouver la clé

Limitations des systèmes à clé secrète

- Toutes les personnes impliquées doivent avoir une copie de la clé
 - Nécessaire d'avoir un moyen sécurisé pour la distribuer (valise diplomatique...)
 - Augmente le risque que la clé tombe dans les mains de l'adversaire
- Si une personne reçoit les messages de n autres
 - Cle unique: un membre peut lire les messages d'un autre
 - Nécessite n clés différentes pour assurer le secret entre les n membres
- Combien pour avoir une confidentialité n vers $n-1$?

Échange d'un secret sur un canal non sécurisé

- Le protocole Diffie-Hellman (1976) permet à deux ordinateurs de générer une clé privée
- Cette clé pourra ensuite être utilisée pour un chiffrement symétrique
- Il n'y a pas besoin de canal sécurisé
- Permet de résoudre le problème de la distribution des clés
- Utilise la notion de racine primitive (ou générateur primitif)
 - Soit p un nombre premier, une racine primitive g est un nombre tel que, quand n va de 1 à $p-1$, $g^n \bmod p$ prend toutes les valeurs de $1..p-1$

Diffie-Hellman

- Alice et Bob veulent communiquer
- Ils choisissent 2 nombres premiers g et p avec p grand (> 512 bits) et g racine primitive de p
- Ces deux nombres n'ont pas besoin d'être gardés secrets
- Alice et Bob choisissent chacun un grand nombre aléatoire (resp. a et b) et le gardent secret
- Alice calcule $A = g^a \bmod p$ et l'envoie à Bob
- Bob calcule $B = g^b \bmod p$ et l'envoie à Alice
- Alice et Bob calculent leur clé privée
 - Alice: $K = B^a \bmod p = (g^b)^a \bmod p$
 - Bob: $K = A^b \bmod p = (g^a)^b \bmod p$
- Cette clé peut maintenant être utilisée pour un codage symétrique
- Une attaque nécessiterait de calculer a à partir de A et b à partir de B , ce qui est extrêmement difficile
- Vulnérable à l'impersonnation

Systemes à clé publique clé privée

- Inventé en 1976 par Whitfield Diffie et Martin Hellman
- Idée: avoir 2 clefs, une publique servant a chiffrer, une privée servant a déchiffrer
- La clé publique est ... publique(!), n'importe qui peut l'avoir sans compromettre le système
- La clé privée sert a déchiffrer et doit rester secrète
- Avantages:
 - Plus besoin de canal sécurisé de distribution des clés: on met les clés publiques sur Internet
 - Communications de n vers 1 privées
- Principe général
 - On fabrique une clé privée
 - On en déduit (par une fonction mathématique) une clé publique
 - Toute la force repose sur le fait que retrouver une clé privée depuis une clé publique est impossible en un temps fini, même pour un ordinateur
 - On utilise pour cela des fonctions dites à sens unique (one way)

Cryptosystème RSA

- Inventé en 1987 par Rivest, Shamir, et Adleman
- Donné naissance à la compagnie RSA en 1992
- Système à clé publique clé privée
- Basé sur la théorie des grands nombres

Rappels mathématiques

- Tout nombre peut se décomposer en un produit unique de nombres premiers
 - $1176 = 2 * 2 * 2 * 3 * 7 * 7$
- Deux nombres sont premiers entre eux si leur PGCD vaut 1
 - $\text{PGCD}(21,10) = 1$
 - $\text{PGCD}(15,10) = 5$
- Inverse modulo
 - L' inverse de e modulo n est d tel que $e*d = 1 \pmod n$
- Fonction phi d'Euler
 - $\Phi(n)$ = combien de nombres entre 1 et n-1 premiers avec n
- Théorèmes
 - $\Phi(P*Q) = (P-1)*(Q-1)$ if P et Q sont des nombres premiers
 - Si $\text{PGCD}(T,R) = 1$ et $T < R$, alors $T\Phi(R) = 1 \pmod R$

RSA - algorithme

- Choisir aléatoirement 2 grands nombres premiers p et q (> 100 chiffres chacun)
- Calculer $n = pq$
- Choisir un petit entier impair e premier avec $\Phi(n)$. Combien vaut $\Phi(n)$?
- Calculer d inverse de e modulo $\Phi(n)$.
- La clé publique P est $P = (e,n)$
- La clé secrète S est $S = (d,n)$
- Chiffrement(M) = $Me \pmod n$
- Déchiffrement(C) = $Cd \pmod n$

Casser RSA

- Étant donné un message chiffré, comment retrouver le message initial?
- L'attaquant possède
 - L'algorithme
 - La clé publique (e,n)
- Il faut retrouver la clé privée (d,n) ce qui revient à trouver d
 - On cherche donc à factoriser n pour retrouver p et q
 - Ce qui nous donnera $\Phi(n)$, e et finalement d
- Est-ce difficile?
 - Extrêmement, les meilleurs algo sont proches de l'exponentielle!
- Exemple: nombre de Pentium 500Mhz et mémoire requise pour factoriser une clé en 1 an en utilisant le General Number Field Sieve (GNFS)

Taille (bits)	Machines	Mémoire
430	1	trivial
760	215 000	4 Go
1020	342 000 000	170 Go
1620	1.6×10^{15}	120 To

<http://www.rsasecurity.com/>

Signature électronique

- Comment authentifier la provenance d'un message?
 - On le signe
- Comment s'assurer que la signature est associée au bon message?
 - Il faut lier la signature au message
- On ne signe pas le message mais un hash du message!
- Comment on signe ce hash ?
 - On le chiffre avec notre clé privée
 - N'importe qui ayant notre clé publique (i.e. tout le monde) peut déchiffrer la signature, trouver la valeur de hash et vérifier que c'est bien celle du message

Exemple de signature numérique

