Chapitre 14

Sécurité informatique

« Seul un ordinateur éteint, enfermé dans un coffre-fort et enterré six pieds sous terre dans un endroit tenu secret peut être considéré comme sécurisé. Et encore... »

Bruce Schneier

14.1. Logiciels malveillants

Les débuts (années 1970-1980)

- **1971 – Creeper.** Le premier programme souvent considéré comme un virus informatique est « Creeper », créé par Bob **Thomas** chez BBN Technologies. Creeper se propageait à travers ARPANET, l'ancêtre d'Internet, et affichait le message « I'm the creeper: catch me if you can ».
- **1982 - Elk Cloner.** L'un des premiers virus à se propager largement, Elk Cloner, fut écrit par un lycéen nommé Richard **Skrenta**. Il infectait les ordinateurs Apple II via des disquettes et affichait un poème après la 50e utilisation de la disquette infectée.

Crédit photo - LIFE - Time Inc 26 sept. 1988

Les années 1980-1990 : l'essor des virus

- **1986 – Brain.** Brain est le premier virus pour PC, créé par deux frères pakistanais, Basit et Amjad **Farooq Alvi**. Ce virus infectait le secteur de démarrage des disquettes.
- **1987 – Lehigh.** Un virus qui infectait les fichiers de démarrage de DOS.
- **1988 - Morris Worm.** Un des premiers vers informatiques, créé par Robert Tappan **Morris**. Il se propageait via les réseaux et infectait les ordinateurs UNIX, causant des perturbations importantes sur Internet.
- **1990 - Dark Avenger.** Un virus bulgare qui marquait l'avènement des virus polymorphes, capables de modifier leur code pour éviter la détection.

Les années 1990-2000 : les virus se multiplient

- **1995 – Concept.** Le premier macrovirus qui infectait les documents Microsoft Word.
- **1999 – Melissa.** Un virus de type macro qui se propageait via des documents Word envoyés par e-mail. Il a causé des perturbations importantes et des dommages financiers.
- **2000 – ILOVEYOU.** Un des virus les plus destructeurs de l'histoire, ILOVEYOU se propageait via des e-mails en tant que lettre d'amour et infectait des millions d'ordinateurs, causant des milliards de dollars de dommages.

Les années 2000-2010 : l'ère des vers et des trojans

- **2001 - Code Red.** Un ver qui attaquait les serveurs web Microsoft IIS. Il infectait des milliers de machines en quelques heures.
- **2003 – Blaster.** Un ver qui exploitait une vulnérabilité dans les systèmes Windows et causait des redémarrages fréquents.
- **2004 – MyDoom.** Un des vers les plus rapides à se propager, il causait des ralentissements importants sur Internet.
- **2007 - Storm Worm.** Un cheval de Troie qui se propageait par e-mail en utilisant des

tactiques de phishing et infectait des millions d'ordinateurs pour les utiliser dans des botnets.

Les années 2010 à nos jours : les menaces sophistiquées

- **2010 – Stuxnet.** Un ver informatique sophistiqué, développé apparemment par les États-Unis et Israël pour cibler les installations nucléaires iraniennes. C'est le premier exemple connu de malware conçu pour endommager des infrastructures physiques.
- **2013 – CryptoLocker.** Un ransomware qui chiffrait les fichiers des utilisateurs et demandait une rançon pour les déchiffrer. Ce type de malware a depuis proliféré.
- **2016 – Mirai.** Un malware qui infecte les appareils IoT (Internet of Things) pour créer des botnets utilisés dans des attaques DDoS massives.
- **2017 – WannaCry.** Un ransomware qui exploitait une vulnérabilité de Windows pour se propager rapidement à travers le monde, affectant des centaines de milliers d'ordinateurs et causant des perturbations majeures.
- **2020 – SolarWinds.** Une cyberattaque majeure impliquant une compromission de la chaîne d'approvisionnement de SolarWinds (une société américaine d'informatique), affectant de nombreuses organisations gouvernementales et entreprises privées.

Aujourd'hui

Les virus et malwares continuent d'évoluer, devenant de plus en plus sophistiqués. Les cyberattaques ciblent désormais non seulement les ordinateurs personnels mais aussi les infrastructures critiques, les appareils IoT, et les systèmes industriels. Les enjeux de la cybersécurité n'ont jamais été aussi importants, et la lutte contre ces menaces nécessite des efforts concertés des gouvernements, des entreprises et des individus.

Cette histoire montre comment les virus informatiques ont évolué d'expériences de laboratoire à des menaces globales, avec des motivations allant de la simple curiosité à des attaques géopolitiques et criminelles.

14.1.1. Virus

Le virus classique est un programme, souvent écrit en assembleur, qui s'intègre dans un programme « hôte », le plus souvent à la fin du code, mais aussi au début, ou au milieu. Chaque fois que l'utilisateur exécute ce programme infecté, il active le virus qui va s'intégrer dans d'autres programmes. De plus, lorsqu'il contient une charge utile[1], il peut, après un certain temps (qui peut être très long) ou un événement particulier, exécuter une action prédéterminée. Cette action peut aller d'un simple message apparaissant à l'écran à la détérioration de certains fichiers voire même à la destruction complète de toutes les données de l'ordinateur.

Il peut se répandre à travers tout moyen d'échange de données numériques comme Internet, mais aussi les disquettes, les cédéroms, les clefs USB, etc.

Son appellation provient d'une analogie avec le virus biologique puisqu'il présente des similitudes dans sa manière de se propager en utilisant les facultés de reproduction de la cellule hôte.

Les virus informatiques ne doivent pas être confondus avec les vers, qui sont des programmes capables de se propager et de se dupliquer par leurs propres moyens, mais sans contaminer de programme hôte.

Origine

En août 1961, trois amis informaticiens, Doug **McIlroy**, Victor **Vyssotsky** et Robert **Morris** Senior, ingénieurs des laboratoires *Bell Telephone*, créent un jeu de survie et d'évolution baptisé Darwin. Un nouveau jeu, fortement inspiré de Darwin, apparaît en 1984 : Core War, que le grand public découvrira réellement au travers de la publication de A. K. Dewdney dans la revue *Scientific American* : « *In the game called Core War hostile programs engage in a battle of bits* ».

Pour ce jeu, chaque joueur écrit un programme, ensuite chargé en mémoire vive. Le système d'exploitation exécute tour à tour une instruction de chacun des logiciels. L'objectif du jeu est de détruire les programmes adverses tout en assurant sa propre prolifération. Les joueurs ne connaissent évidemment pas l'emplacement du programme adverse. Les logiciels sont capables de se recopier, de

1 En informatique, on utilise ce terme au figuré pour désigner la partie du code exécutable d'un virus qui est spécifiquement destinée à nuire (par opposition au code utilisé par le virus pour se répliquer notamment).

se réparer, de se déplacer eux-mêmes en différentes zones de la mémoire et « d'attaquer » le logiciel adverse en écrivant aléatoirement dans d'autres zones de la mémoire. C'est exactement un des principes de programmation des virus. La partie se termine au bout d'un temps défini ou lorsque l'un des joueurs voit tous ses programmes inactifs ou détruits. Le vainqueur est celui qui possède le plus grand nombre de copies actives.

14.1.2. Ver

Un ver informatique (« worm » en anglais) est un logiciel malveillant qui se reproduit sur plusieurs ordinateurs en utilisant un réseau informatique comme Internet. Le terme « ver informatique » fit son apparition pour la première fois en 1975 dans un livre de science-fiction de John **Brunner** intitulé *The Shockwave Rider*[2].

Un ver, contrairement à un virus informatique, n'a pas besoin d'un programme hôte pour se reproduire. Il exploite les différentes ressources de l'ordinateur qui l'héberge pour assurer sa reproduction.

L'objectif d'un ver n'est pas seulement de se reproduire. Le ver a aussi habituellement un objectif malicieux, par exemple :

- espionner l'ordinateur où il se trouve ;
- offrir une porte dérobée[3] à des pirates informatiques ;
- détruire des données sur l'ordinateur où il se trouve ou y faire d'autres dégâts ;
- envoyer de multiples requêtes vers un serveur Internet dans le but de le saturer (déni de service[4]).

L'activité d'un ver a souvent des effets secondaires comme :

- le ralentissement de la machine infectée ;
- le ralentissement du réseau utilisé par la machine infectée ;
- le plantage de services ou du système d'exploitation de la machine infectée.

Des vers écrits sous forme de script peuvent être intégrés dans un courriel ou sur une page HTML. Ces vers sont activés par les actions de l'utilisateur qui croit accéder à des informations lui étant destinées. Un ver peut aussi être programmé en C, C++, assembleur, ou dans un autre langage de programmation. La plupart du temps, les vers utilisent des failles de logiciels pour se propager. Ces failles sont habituellement corrigées par les éditeurs de logiciels dès que les vers apparaissent.

Origine

Creeper est considéré comme le premier ver informatique. Développé, en 1971, pour les ordinateurs fonctionnant avec le système d'exploitation Tenex par Bob **Thomas** de la société Bolt Beranek and Newman (BBN), il rentre dans la catégorie des vers créés par des chercheurs soucieux d'améliorer notre quotidien.

Creeper est un programme de démonstration développé en réponse aux besoins des contrôleurs aériens. Ces derniers, à partir de l'ARPANET, souhaitaient recevoir une notification lorsque les avions sous leur contrôle passaient de la surveillance d'un ordinateur à l'autre. Le ver se propage de machine en machine et s'efface des machines qu'il visite, au fur et à mesure de sa progression. À chaque exécution, le message *I'm creeper! Catch me if you can!* s'affiche sur les consoles. D'autres versions du ver ont vu le jour par la suite. Dans ces versions, Creeper se reproduisait de machine en machine sans s'effacer à chaque passage. Face à ce dysfonctionnement, Bob Thomas développe un autre ver Reaper qui efface et élimine Creeper et s'auto-détruit ensuite. Le premier ver antiviral est né.


Robert Tappan **Morris**

Durant ses études à l'Université de Cornell, Robert Tappan **Morris** conçoit un programme dans le but de mesurer la taille du réseau Internet. Le programme a pour objectif de s'introduire sur les systèmes Unix reliés au réseau Internet et de se répliquer pour propager son activité. C'est le 2 novembre 1988 que le premier ver Internet a été lancé à partir du MIT. Le « ver Morris » n'a pas été écrit pour causer des dommages. Cependant, le mécanisme de reproduction forcée mis en place va favoriser l'infection sur le réseau Internet, mais va également engendrer une saturation des ressources de chaque système. En quelques jours, ce sont plus de 6000 serveurs Unix qui vont être infectés sur

2 Le livre a été traduit en français avec le titre « Sur l'onde de choc ».
3 Dans un logiciel, une *porte dérobée* (de l'anglais *backdoor*, littéralement *porte de derrière*) est une fonctionnalité inconnue de l'utilisateur légitime, qui donne un accès secret au logiciel.
4 Voir paragraphe 14.2.6

un parc d'environ 60'000 serveurs reliés au réseau Internet de l'époque, soit un taux d'infection de 10%. S'apercevant des dégâts qu'il était en train de causer, Robert T. Morris alerta certains membres de la communauté informatique et une parade fut finalement trouvée. Morris fut condamné à une peine de travaux d'intérêt général et à une amende de 10'000 dollars. La communauté Internet - alors réduite - pris conscience de la menace de ce type de code malveillant et c'est ainsi que fut créé le CERT (*Computer Emergency Response Team*).

I LOVE YOU est le nom d'un ver informatique, apparu pour la première fois le 4 mai 2000 et envoyé sous forme d'une pièce jointe à un courriel intitulé I LOVE YOU. Le destinataire naïf du courriel croit que la pièce jointe est un fichier de texte. En fait, en ouvrant le fichier, l'utilisateur déclenche l'exécution d'un programme contenu dans le fichier. Ce programme explore la liste des contacts de l'utilisateur et envoie à tous ces contacts un courriel contenant la même pièce jointe infectée, assurant ainsi sa reproduction. Il s'est répandu en quatre jours sur plus de 3,1 millions de machines dans le monde et causé de gros dégâts : I LOVE YOU s'en prend à de multiples extensions de fichiers, qui sont alors perdus.

Les vers actuels se propagent principalement comme le faisait le ver I LOVE YOU.

14.1.3. Cheval de Troie (Trojan)

Un cheval de Troie est un logiciel d'apparence légitime, mais conçu pour exécuter des actions à l'insu de l'utilisateur. En général, un cheval de Troie tente d'utiliser les droits appartenant à son environnement pour détourner, diffuser ou détruire des informations, ou encore pour ouvrir une porte dérobée qui permettra à un attaquant de prendre à distance le contrôle d'un ordinateur, via Internet.

Le téléchargement de programmes gratuits et le partage des programmes sont les principales sources de diffusion des chevaux de Troie. Ils sont également très fréquents dans certains types de courriels.

Un cheval de Troie n'est pas un virus informatique, car il ne se reproduit pas par lui-même. Un cheval de Troie est conçu pour être reproduit lors de téléchargements ou de copies par des utilisateurs naïfs, attirés par les fonctionnalités du programme.

14.1.4. Bombe logique

On appelle « bombes logiques » les dispositifs programmés 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 simultanément sur un grand nombre de machines. Par exemple : la bombe logique TCHERNOBYL s'est activée le 26 avril 1999, jour du 13ème anniversaire de la catastrophe nucléaire...

Généralement, le but des bombes logiques est de créer un déni de service !

14.1.5. Logiciel espion

Un logiciel espion (en anglais « *spyware* ») est un logiciel malveillant qui s'installe dans un ordinateur dans le but de collecter et transférer des informations sur l'environnement dans lequel il s'est installé, très souvent sans que l'utilisateur n'en ait connaissance. L'essor de ce type de logiciel est associé à celui d'Internet qui lui sert de moyen de transmission de données.

Un logiciel espion est composé de trois mécanismes distincts :

- l'infection, qui installe le logiciel. Ce mécanisme est identique à celui utilisé par les virus, les vers ou les chevaux de Troie ;
- la collecte d'information ;
- la transmission à un tiers, généralement assurée via le réseau Internet. Le tiers peut être le concepteur du programme ou une entreprise.

Le logiciel espion peut afficher des offres publicitaires, télécharger un virus, installer un cheval de Troie, capturer des mots de passe en enregistrant les touches pressées au clavier ou encore espionner les sites Internet visités.

14.1.6. Machine zombie

En sécurité informatique, une machine zombie est un ordinateur contrôlé à l'insu de son utilisateur par un pirate informatique. Ce dernier l'utilise alors le plus souvent à des fins malveillantes, par exemple afin d'attaquer d'autres machines en dissimulant sa véritable identité. Un zombie est souvent infesté à l'origine par un ver ou un cheval de Troie.

Réseau de machines zombies (Botnet)

Un **réseau de machines zombies** peut être constitué et contrôlé par une ou plusieurs personnes, afin d'obtenir une capacité considérable et d'avoir un impact plus important.

Des armées de zombies sont utilisées dans les attaques de type déni de service ou des tâches diverses comme les envois en masse de courriers non sollicités (spam).

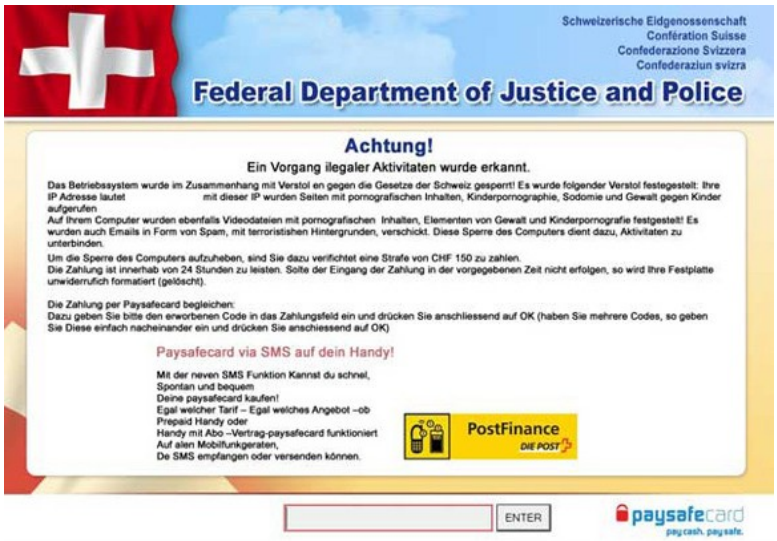
Un réseau de machines zombies peut aussi être utilisé afin de fournir aux pirates une puissance de calcul phénoménale, leur permettant de déchiffrer un code en un temps considérablement plus court que sur une machine.

En dehors de l'aspect économique, les attaques informatiques peuvent devenir une arme de propagande ou de rétorsion, notamment lors de conflits armés ou lors d'événements symboliques. Par exemple, lors du conflit entre la Russie et la Géorgie en 2008, le réseau géorgien a été attaqué sous de multiples formes pour le rendre indisponible. En 2007, une attaque d'importance contre l'Estonie a également eu lieu : la motivation des pirates aurait été le déplacement d'un monument en hommage aux soldats russes du centre de la capitale estonienne.

14.1.7. Rançongiciel

Un **rançongiciel** (ou *ransomware* en anglais) est un logiciel malveillant qui prend en otage des données personnelles. Pour ce faire, un rançongiciel chiffre des données personnelles puis demande à leur propriétaire d'envoyer de l'argent en échange de la clé qui permettra de les déchiffrer.

Un ransomware peut aussi bloquer l'accès de tout utilisateur à une machine jusqu'à ce qu'une clé ou un outil de débridage soit envoyé à la victime en échange d'une somme d'argent. Les modèles modernes de rançongiciels sont apparus en Russie initialement, mais on constate que le nombre d'attaques de ce type a grandement augmenté dans d'autres pays, entre autres l'Australie, l'Allemagne, les États-Unis.



Par exemple, le ransomware **Federal Department of Justice and Police** vous empêchera d'accéder à vos fichiers ou de lancer des programmes. Vous ne verrez même pas votre bureau. À la place, vous verrez un message menaçant affiché par le virus, qui vous ordonnera de payer une amende parce que vous auriez soi-disant visité des sites à contenu illégal. Petit raffinement : votre webcam va s'allumer, vous faisant croire que vous avez été pris en photo.

Le but est de tromper les gens afin qu'ils donnent de l'argent ou des informations personnelles et financières aux cybercriminels.

Il ne sert évidemment à rien de payer, votre ordinateur ne marchera pas mieux après...

14.1.8. Protection contre les codes malveillants

L'antivirus : le surveillant principal

Le rôle de l'antivirus est de débusquer tous les virus qui essaient de s'installer dans l'ordinateur. Mais en réalité, il n'est capable de repérer que les codes malveillants qu'il connaît déjà.

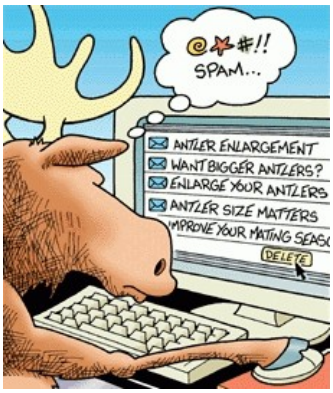
Trois méthodes principales sont utilisées pour détecter des virus et autres codes malveillants :

- **L'analyse de signatures** : elle consiste à vérifier la signature du programme ou du fichier en question. Cette signature est une partie de code générée à partir du contenu du fichier qui permet de vérifier qu'il n'a pas été modifié ultérieurement. Elle constitue en quelque sorte sa carte d'identité. Mais cette simple analyse de signatures est limitée et ne semble plus suffisante aujourd'hui. En effet, elle ne s'applique qu'aux virus connus et analysés. De plus, la gestion d'une base virale est contraignante : stockage sécurisé des données, espace de stockage, multiples mises à jour, etc.
- **L'analyse heuristique** : elle recherche des actions potentiellement dangereuses. Mais cette technique, peu fiable, est source de fausses alertes.
- **L'analyse comportementale** : il s'agit d'étudier le comportement des logiciels installés sur le PC. Analysant en permanence l'activité de l'ordinateur, l'antivirus surveille et tente de bloquer des actions qui ne sont pas répertoriées dans sa base comportementale : tentatives d'ouverture en lecture/écriture de fichiers exécutables, écriture sur une partition, etc.

Le pare-feu

Un pare-feu (« *firewall* » en anglais) est un logiciel ou un matériel, qui a pour but de faire respecter la politique de sécurité du réseau, celle-ci définissant quels sont les types de communication autorisés ou interdits. Ce système filtre en permanence toutes les connexions entrantes et sortantes d'un ordinateur, servant d'intermédiaire entre le réseau local (ou la machine locale) et un ou plusieurs réseaux externes. En surveillant les connexions, il permet de protéger un ordinateur ou un réseau d'ordinateurs des intrusions provenant d'un réseau tiers (notamment Internet).

14.2. Attaques et fraudes



14.2.1. Spam

Le pourriel (québécisme) ou spam (anglicisme) désigne une communication électronique non sollicitée, en premier lieu via le courrier électronique. Il s'agit en général d'envois en masse effectués à des fins publicitaires, malhonnêtes ou sans but premier sinon que d'agacer leur destinataire.

Selon *Statista*, la part des spams dans le trafic total de mails était de 45,2 % dans le monde en décembre 2022.

Histoire

Le premier pourriel a été envoyé le 3 mai 1978 par Gary **Thuerk**, marketeur travaillant chez DEC. Il envoya son message à près de la totalité des utilisateurs d'ARPAnet (ancêtre d'Internet) vivant sur la côte ouest des États-Unis, soit environ 600 personnes. Il fit cela sans mauvaise intention, afin d'inviter ces utilisateurs technophiles à une démonstration de la gamme DEC. Voulant éviter d'écrire un message à chaque adresse, il mit les 600 adresses directement dans le champ « Destinataire ». Les réactions furent vives et contrastées, l'administration américaine gérant le réseau condamnant d'office la pratique, la jugeant non-conforme aux termes d'utilisation du réseau.

L'association de spam et de indésirable provient d'un sketch comique des Monty Python dans lequel le même mot, désignant un jambon en boîte de basse qualité, envahit la conversation et le menu d'un petit restaurant. SPAM est la contraction de Spiced Pork And Meat (jambon et viande épicés) et est une marque créée et déposée par Hormel Foods en 1937.

L'anti-spam

Filtrant les courriels, l'anti-spam s'appuie sur différentes banques de données, notamment des listes noires constituées d'adresses internet ou de pays connus pour être à l'origine de nombreux *spams*. Une seconde famille d'anti-spams utilise l'authentification des expéditeurs. Ce filtrage peut éviter de recevoir les messages qui véhiculent les virus.

Dans les blogs

La présence de liens vers un site web est un critère important de classement dans les moteurs de recherche. Afin d'augmenter artificiellement le nombre de liens pointant vers leurs sites, certains créent des blogs, ou mettent des messages de commentaires dans des blogs préexistants, uniquement pour ajouter des liens vers un ou plusieurs sites web à promouvoir.

L'automatisation de ce genre de pollution a mené plusieurs logiciels de blog à introduire des contrôles (Captcha) qui rendent cette automatisation par une machine plus complexe à réaliser.

14.2.2. Hameçonnage (phishing)

L'hameçonnage, appelé en anglais « *phishing* », est une technique utilisée par des fraudeurs pour obtenir des renseignements personnels dans le but d'usurper une identité. La technique consiste à faire croire à la victime qu'elle s'adresse à un tiers de confiance — banque, administration, etc. — afin de lui soutirer des renseignements personnels : mot de passe, numéro de carte de crédit, date de naissance, etc. L'hameçonnage peut se faire par courrier électronique, par des sites web falsifiés ou autres moyens électroniques.

Les criminels informatiques utilisent généralement l'hameçonnage pour voler de l'argent. Les cibles les plus populaires sont les services bancaires en ligne, et les sites de ventes aux enchères tels que **eBay** ou **Paypal**. Les adeptes de l'hameçonnage envoient habituellement des courriels à un grand nombre de victimes potentielles.

Une approche souvent utilisée est de se faire passer pour une banque et d'indiquer à la victime que son compte a été désactivé à cause d'un problème et que la réactivation ne sera possible qu'en cas d'action de sa part. Le message fournit alors un lien qui dirige l'utilisateur vers une page web qui ressemble à s'y méprendre au vrai site de la société. Arrivé sur cette page trompeuse, l'utilisateur est invité à saisir des informations confidentielles qui sont alors enregistrées par le criminel.

Il faut savoir que **jamais** une banque ne demandera à un client de lui fournir un code confidentiel.

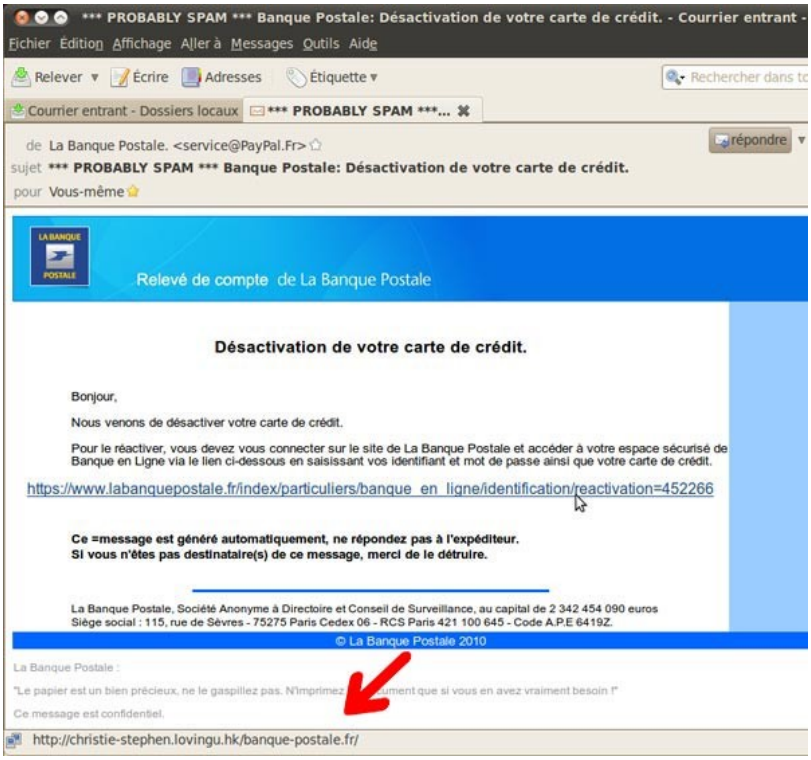
Trois règles pour détecter le phishing

1. Vérifiez le contexte du mail

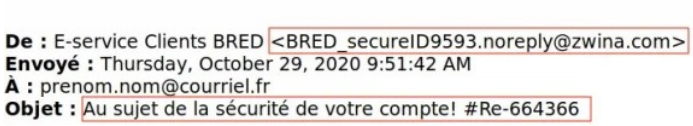
- Est-ce qu'il a du sens ? Est-ce normal que vous le receviez ?
- Détectez-vous des anomalies ? Est-ce qu'il a l'air bizarre ?
- Est-ce que le message est urgent ? Êtes-vous mis sous pression ?
- Est-ce qu'il comporte des fichiers joints ?
- Est-ce que vous attendiez le message ?

2. Vérifiez où les liens internet du message vous redirigent

- Pour voir la destination d'un lien, passez votre souris dessus sans cliquer.



3. Ne tenez pas compte de l'adresse de l'expéditeur



- L'adresse de l'expéditeur n'est pas pertinente parce qu'on peut la falsifier.
- Même si elle semble valide, ce n'est pas la preuve que le message est légitime.

14.2.3. Email spoofing

Un hacker prétend vous avoir piraté et exige un virement de 500 euros en Bitcoin. Le mail est plutôt crédible, car il est envoyé depuis votre propre adresse email. Mais ne vous fiez pas aux apparences. Ce hacker n'a pas accès à votre boîte mail, il utilise simplement une technique informatique permettant d'usurper votre identité.

Cette arnaque par email aurait commencé lors de l'été 2018. Les victimes reçoivent un email écrit par un hacker qui prétend avoir piraté leur ordinateur.

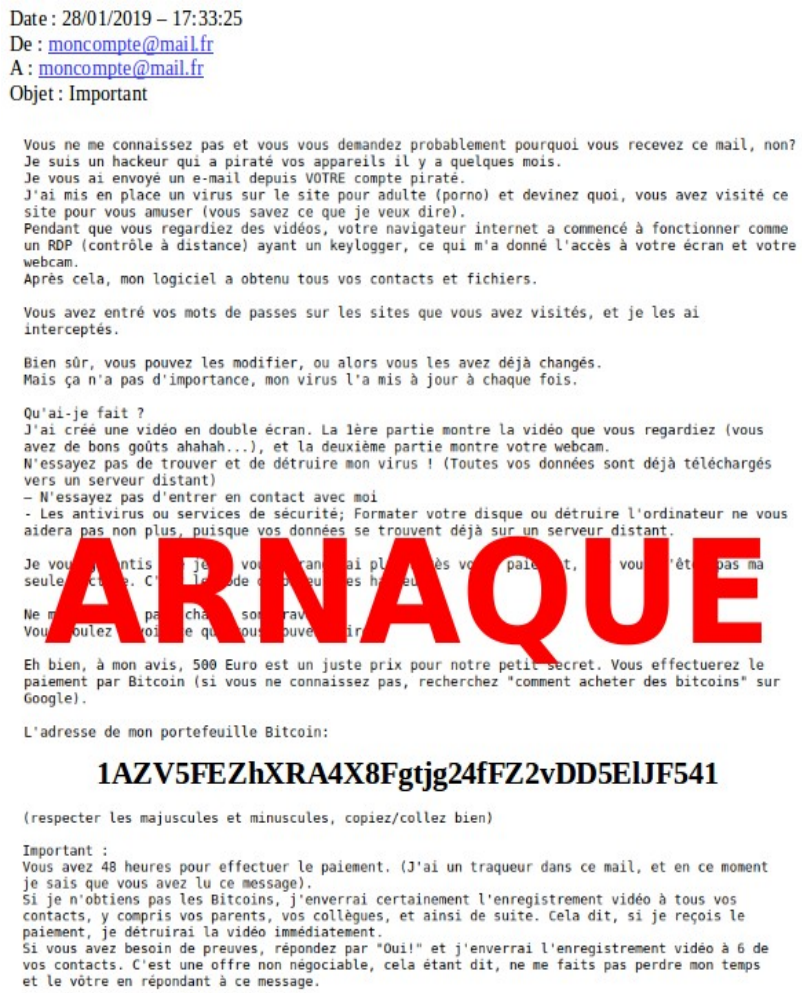
C'est troublant : vous jetez un œil à l'expéditeur, et vous découvrez votre propre adresse email, voire votre prénom, votre nom et votre photo (en fonction de votre messagerie). Ces premiers éléments crédibilisent le piratage. Le hacker poursuit en tentant de vous embrouiller : il explique qu'il a exploité une faille de sécurité puis emploie des mots techniques que 99 % des internautes ne comprendront pas. Il indique qu'il est inutile de changer votre mot de passe, qu'il a parcouru votre historique web, qu'il a accès à votre webcam, qu'il détient des photos et des vidéos de vous… et menace de tout envoyer à vos contacts si vous ne lui transférez pas quelques centaines d'euros en Bitcoin.

Ne le faites surtout pas : c'est un piège. Le hacker n'a pas réellement accès à vos données personnelles, il a simplement usurpé votre adresse email.

Cette arnaque est basée sur de l'email spoofing, l'usurpation de votre adresse email. L'adresse d'expédition d'un email est une donnée peu sécurisée : une personne mal intentionnée qui s'y connait un peu en informatique peut facilement masquer sa véritable identité et choisir l'adresse email

indiquée dans le champ Expéditeur. Cette donnée peut être falsifiée et c'est précisément ce qu'il s'est passé. Le hacker n'a pas vraiment envoyé ce mail depuis votre messagerie, il a simplement remplacé sa véritable identité par votre adresse email – et votre système de messagerie n'y a vu que du feu.

Le plus simple, pour un hacker mal intentionné, est d'utiliser l'email spoofing avec la même adresse email en expéditeur et en destinataire : la victime reçoit un email provenant de sa propre adresse, la menace semble crédible (encore une fois, vous n'avez rien à craindre). Mais cette technique peut également être utilisée pour des escroqueries plus complexes : des comptables peuvent ainsi recevoir un email les invitant à réaliser de prochains virements sur un autre compte bancaire, détenu par des escrocs. Ces derniers se font alors passer pour des personnes qui font autorité, tel que le Président de l'entreprise ou un fournisseur bien identifié.



Dans tous les cas : si vous avez reçu un email d'un prétendu hacker, envoyé avec votre adresse email, ne payez pas la rançon demandée, c'est une arnaque. Et en cas de doute sur l'identité de l'expéditeur, ou si des sommes d'argent sont en jeu, n'hésitez pas à contacter la personne directement, par téléphone par exemple, pour être sûr qu'elle est bien à l'origine de la demande.

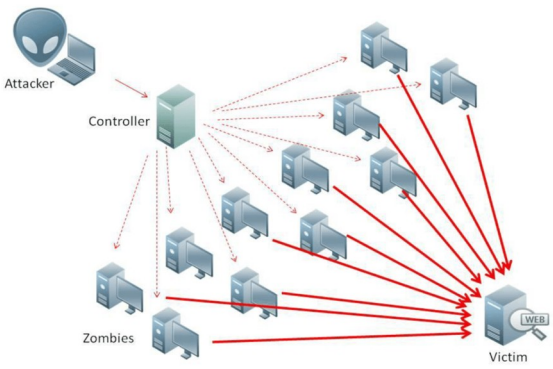
14.2.4. Attaque par déni de service

Une attaque par déni de service (en anglais, *Denial of Service attack* ou *DoS attack*) est une attaque sur un serveur informatique qui a pour conséquence l'incapacité pour le serveur de répondre aux requêtes de ses clients.

Pour illustrer ce dont il s'agit, il suffit d'imaginer ce qui arriverait si des centaines de personnes téléphonaient sans arrêt au même restaurant pour commander des choses qu'elles ne veulent pas vraiment. Les lignes téléphoniques seraient alors saturées, et la cuisine serait si débordée qu'elle ne

pourrait prendre d'autres commandes.

Une attaque par déni de service distribuée (en anglais, *Distributed Denial of Service attack* ou *DDoS attack*) est une attaque par déni de service dans laquelle le serveur cible est attaqué par plusieurs ordinateurs simultanément. Dans ce type d'attaque, les pirates se dissimulent souvent derrière des machines zombies, utilisées à l'insu de leurs propriétaires.



Depuis quelques années, l'attaque par déni de service distribuée est aussi utilisée à des fins de chantage auprès d'entreprises dont l'activité commerciale repose sur la disponibilité de leur site Web. Ces fraudes sont habituellement le fait d'organisations criminelles mafieuses et non de pirates isolés.

Le 6 août 2009, Twitter, Facebook et certains sites appartenant à Google ont été victimes d'une cyberattaque. Les millions d'utilisateurs se sont vus dans l'impossibilité de se connecter pendant plusieurs heures.

Pour quelles raisons un pirate en voudrait-il à Twitter au point de rendre son site indisponible ? Au rang des motivations des dénis de service, on compte :

- **le chantage** : c'est la principale motivation pour les attaques. D'ordinaire, ce type d'attaque vise plutôt des sites de e-commerce, ou de jeu d'argent en ligne : c'est-à-dire, des sociétés dont l'existence même repose entièrement sur leur visibilité sur Internet. L'attaque commence par quelques minutes d'indisponibilité ; puis le site revient en ligne, et l'entreprise reçoit un e-mail du pirate, demandant le versement d'une somme d'argent en échange de l'arrêt des hostilités. Pour ces sociétés, la perte du chiffre d'affaires d'une seule journée est une catastrophe. Et même si ces sociétés n'aiment pas en faire la publicité, dans de nombreux cas, les escrocs reçoivent bien la rançon demandée (c'est d'ailleurs à la remise de la rançon que sont réalisées la plupart des arrestations). Ici, la société Twitter répond bien à ce critère de dépendance extrême vis-à-vis d'Internet ; plus de twitter.com = plus de société. Cependant, le chiffre d'affaires réalisé par la société sur Internet est théoriquement nul, et elle se remettra donc très bien de quelques heures d'indisponibilité, malgré avoir perdu quelques plumes dans les médias. Cela cadre donc mal avec une tentative de racket ;
- **la censure** : il est possible que des messages aient été relayés sur Twitter, et que ces messages n'aient pas plu à quelqu'un, qui aurait décidé en conséquence d'attaquer le site. Ce type d'attaques est monnaie courante, de nombreux chercheurs en sécurité en faisant régulièrement les frais : le site www.abuse.ch est régulièrement pris pour cible en raison du contenu qui y est publié ; la société SpamHaus est régulièrement attaquée également ; et la fondation CastleCops (qui n'existe plus) était en son temps parmi les priorités de nombreux pirates, qui ont poussé le vice jusqu'à provoquer un déni de service avec des donations issues de comptes bancaires volés, l'objectif étant de compromettre la fondation aux yeux de sa propre banque. Dans le cas de Twitter, il ne serait pas étonnant que parmi les millions de tweets postés chaque jour, certaines aient déplu à des pirates, le site étant devenu un relai fulgurant des rumeurs et de la contestation politique (souvenons-nous de l'appel au déni de service contre des cibles iraniennes relayé sur la plateforme) ;
- **la démonstration de force** : les pirates cherchent souvent à démontrer leurs capacités de nuisance sur des cibles réelles, de manière à séduire des « clients » potentiels qui pourraient faire appel à leurs services. En matière de déni de service, ce type de démonstration de force est quasi systématique : pour vérifier la puissance du botnet, le pirate choisit une cible neutre, sans rapport avec l'usage réel qu'il fera du botnet ultérieurement. Son futur client peut ainsi être rassuré sur la qualité de l'infrastructure qu'il va louer. C'est notamment ce qu'il semble s'être passé début 2007, avec l'attaque massive contre les serveurs DNS racines. Cette hypothèse est plausible, Twitter étant une cible particulièrement médiatique, la publicité du pirate est assurée ; cependant, Twitter semble avoir quelques soucis chroniques de bande passante, lors de pics d'activité le site est parfois indisponible en raison de sa propre popularité (par exemple à la mort de Michael Jackson). Twitter n'est donc pas une cible hors de portée pour un pirate doté d'un outillage « moyen ».

Puis, on notera comme motivations plus improbables : l'attaque pour le plaisir, ou encore l'attaque politique par une puissance étrangère.

On ne saura sans doute jamais le fin mot de l'histoire, mais ce type d'attaques est monnaie

> courante, de nombreuses sociétés en faisant les frais jour après jour. Se prémunir contre ce type d'attaque est très complexe : chaque fois qu'une solution de filtrage est mise en place, les pirates modifient leur attaque : en mobilisant d'autres machines de leur botnet pour multiplier les IP sources, en modifiant les requêtes ou les paquets de données envoyés, en prenant pour cible la partie applicative du site web plutôt que les couches réseaux, etc. Ces attaques ont donc de beaux jours devant elles.
>
> Pierre Caron – Cert-Lexsi (www.zdnet.fr)

14.3. Intégrité

Un des problèmes annexes à la cryptographie est la vérification de l'intégrité d'un message : comment savoir si le message que l'on reçoit n'a pas été modifié en cours de route ? Ce problème peut aussi se poser avec Internet : comment savoir si la version du logiciel X mis en téléchargement sur le web est bien le logiciel original et non pas une version avec un virus ? Quand la police confisque un disque dur pour une enquête, elle a aussi besoin d'assurer l'intégrité de ce disque pour prouver que son contenu n'a pas été modifié entre la confiscation et l'éventuel procès.

La réponse à ces problèmes est la notion de fonction de hachage à sens unique. Cette fonction (H), qui doit être rapide à calculer, transforme un message M de longueur arbitraire en une **empreinte numérique** h de taille fixée : $h = H(M)$, où h est de longueur m.

Cette fonction doit en outre avoir les propriétés suivantes :

- étant donné un message M, il est facile de calculer l'empreinte h,
- étant donné une empreinte h, il est **très** difficile de calculer le message M,
- résistance forte à la collision : il est difficile de trouver deux messages M et M' tels que $H(M)=H(M')$.

Alice envoie à Bob le message lui-même, puis son empreinte qu'elle aura calculée. Bob calcule à son tour l'empreinte du message puis la compare avec celle qu'il a reçue. Si ce sont les mêmes, le message n'a pas été modifié.

On peut voir une analogie avec les empreintes digitales : c'est aussi une manière fiable de vérifier l'identité d'un individu avec une petite quantité d'informations.

Les empreintes numériques sont aussi utilisées pour crypter les mots de passe avant de les stocker sur un serveur.

14.3.1. Exemples de fonction de hachage

Ron **Rivest**
(né en 1947)

MD4 et MD5

MD5 est une version améliorée de MD4, tous deux conçus par Ron **Rivest**. MD signifie « Message Digest », qui peut être traduit par « empreinte » en français. MD4 et MD5 produisent des empreintes de 128 bits. MD5 est un peu plus lent que MD4. Des faiblesses ont été trouvées et son utilisation se raréfie.

SHA-1

SHA signifie « **S**ecure **H**ash **A**lgorithm » et on utilise souvent le terme SHA-1 pour désigner la version. Cette fonction de hachage, elle aussi basée sur MD4, a été publiée conjointement par la NSA[5] et le NIST[6]. Elle renvoie une empreinte de 160 bits.

Les fonctions de hachage cryptographiques comme MD5 et SHA-1 sont considérées cassées car des attaques permettant de réduire significativement leur résistance aux collisions ont été trouvées. SHA-1 est remplacé par SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512) et SHA-3.

5 National Security Agency est un organisme gouvernemental des États-Unis, responsable de la collecte et de l'analyse de toutes les formes de communications, aussi bien militaires et gouvernementales que commerciales ou même personnelles.

6 National Institute of Standards and Technology. Son but est de promouvoir l'économie en développant des technologies, la métrologie et des standards de concert avec l'industrie.

14.3.2. Attaque des anniversaires

Combien faut-il réunir de personnes pour avoir 1 chance sur 2 que deux d'entre elles soient nées le même jour ? Si l'on pose cette question dans la rue, on aura beaucoup de réponses différentes. La plupart des gens pensent qu'il faudra 183 personnes (la moitié de 365). Il n'en est rien. En fait, il suffit de... 23 personnes !

Calculons la probabilité qu'aucune des personnes présentes ait le même anniversaire : nous avons 365 jours possibles pour le premier, 364 pour le deuxième et ainsi de suite. Ce qui nous donne, pour n personnes, $365 \cdot 364 \cdot 363 \cdot \ldots \cdot (365-n+1)$ cas favorables. Il y a bien sûr 365^n cas possibles.

Donc :

$$p = \frac{364}{365} \cdot \frac{363}{365} \cdot \frac{362}{365} \cdot \ldots \cdot \frac{365-n+1}{365}$$

Or, cette probabilité tombe en dessous de 0.5 quand $n = 23$.

Ce « paradoxe » a son importance en cryptographie, lorsqu'on étudie les fonctions de hachage. Pour que cette fonction soit fiable, il ne faut pas que l'on puisse produire deux textes aux sens très différents mais donnant la même empreinte.

Si l'empreinte est codée sur b bits, il y a 2^b empreintes possibles. Si l'on prend k textes différents, la probabilité pour que deux textes aient la même empreinte est donc :

$$1 - \left(1 - \frac{1}{2^b}\right)\left(1 - \frac{2}{2^b}\right)\ldots\left(1 - \frac{k-1}{2^b}\right)$$

Combien l'attaquant doit-il essayer de textes avant de trouver la même empreinte avec une probabilité d'au moins 0.5 ?

Le tableau suivant donne des valeurs pour différentes valeurs de b :

Nombre de bits de l'empreinte (b)	Nombre d'empreintes	Nombre de textes à essayer (k)
8	256	13
16	65536	213
32	$4.3 \cdot 10^9$	54562
64	$1.8 \cdot 10^{19}$	$9.6 \cdot 10^9$
128	$3.4 \cdot 10^{38}$	$1.5 \cdot 10^{19}$
160	$1.4 \cdot 10^{48}$	$1.0 \cdot 10^{24}$
256	$1.1 \cdot 10^{77}$	$2.8 \cdot 10^{38}$

On voit sur ce tableau qu'on a environ une chance sur deux de trouver deux textes à l'empreinte identique en en essayant un nombre de l'ordre de la racine carrée du nombre d'empreintes.

On considère en général que, pour obtenir un niveau de sécurité correct, il faut prendre une taille d'empreinte d'au moins 128 bits.

14.4. Les mots de passe

Les mots de passe sont une protection incontournable pour sécuriser l'ordinateur et ses données ainsi que tous les accès aux services sur Internet. Mais encore faut-il en choisir un robuste. Un bon mot de passe doit être difficile à deviner par une personne tierce et facile à retenir pour l'utilisateur. Il est constitué d'au moins 12 caractères dont :

- des lettres majuscules
- des lettres minuscules
- des chiffres
- des caractères spéciaux

# OF CHARACTER	Numbers Only	Lowercase Letters	Upper & Lowercase Letters	Numbers, Upper & Lowercase Letters	Numbers, Upper & Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	Instantly	4 Seconds
7	Instantly	Instantly	22 Seconds	42 Seconds	6 Minutes
8	Instantly	3 Seconds	19 Minutes	48 Minutes	7 Hours
9	Instantly	1 Minutes	11 Hours	2 Days	2 Weeks
10	Instantly	1 Hours	4 Weeks	6 Months	5 Years
11	Instantly	23 Hours	4 Years	38 Years	356 Years
12	25 Seconds	3 Weeks	289 Years	2K Years	30K Years
13	3 Minutes	11 Months	16K Years	91K Years	2M Years
14	36 Minutes	49 Years	827K Years	9M Years	187M Years
15	5 Hours	890 Years	47M Years	613M Years	14Bn Years
16	2 Days	23K Years	2Bn Years	26Bn Years	1Tn Years
17	3 Weeks	812K Years	539.72M Years	2Tn Years	95Tn Years
18	10 Months	22M Years	7.23Bn Years	96Tn Years	6Qn Years

Un mot de passe est d'autant plus faible qu'il est court. L'utilisation d'un alphabet réduit ou de mot issu du dictionnaire le rend très vulnérable. Les mots du dictionnaire ne doivent pas être utilisés (dans toutes les langues).

Aussi à proscrire, les mots en relation avec soi, qui seront facilement devinables : prénom de ses enfants, nom du chien, dates de naissance, etc.

Réseaux sociaux, adresses mail, accès au banque en ligne, factures en ligne : les accès sécurisés se sont multipliés sur Internet. Au risque de voir tous ses comptes faire l'objet d'utilisation frauduleuse, il est impératif de ne pas utiliser le même mot de passe pour des accès différents.

Parenthèse mathématique

Si le mot de passe contient n caractères, indépendants (la présence d'un caractère ne va pas influencer un autre) et uniformément distribués (aucun caractère n'est privilégié), le nombre maximum d'essais nécessaires se monte alors à :

- 26^n si le mot de passe ne contient que des lettres de l'alphabet totalement en minuscules ou en majuscules ;
- 52^n si le mot de passe ne contient que des lettres de l'alphabet, avec un mélange de minuscules et de majuscules ;
- 62^n si le mot de passe mélange les majuscules et les minuscules ainsi que les chiffres.

Il suffit en fait d'élever la taille de « l'alphabet » utilisé à la puissance n. Il s'agit ici d'une borne supérieure et en moyenne, il faut deux fois moins d'essais pour trouver le mot de passe (si celui-ci est aléatoire). En réalité, bien peu de mots de passe sont totalement aléatoires et le nombre d'essais est bien inférieur aux limites données ci-dessus (grâce à la possibilité d'une attaque par dictionnaire).

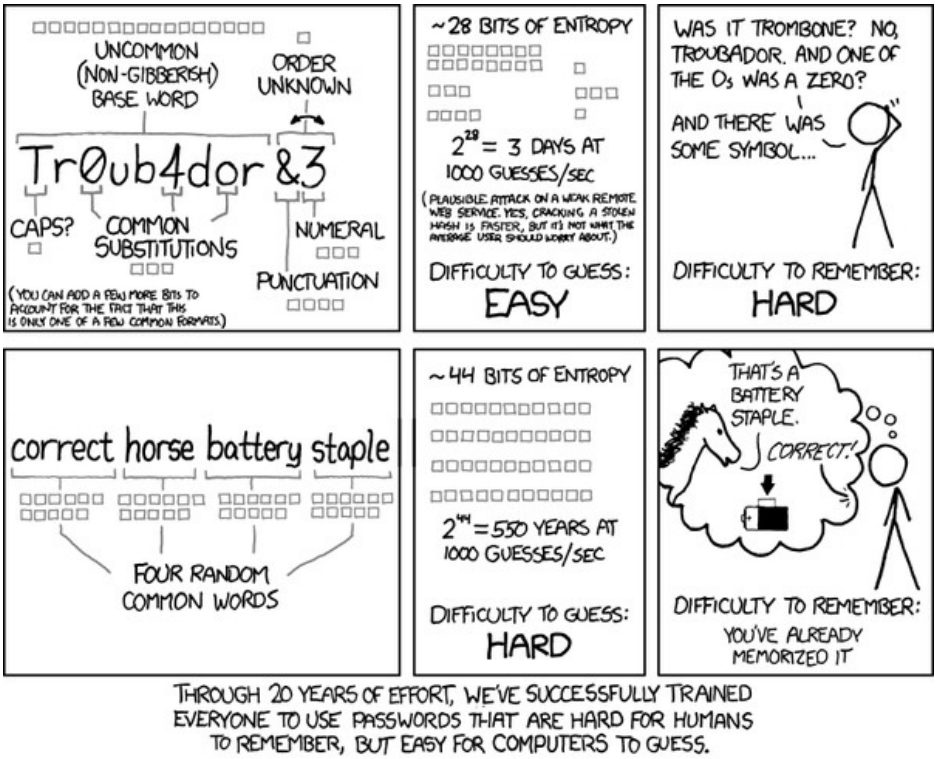
Le tableau ci-dessous donne le nombre maximum d'essais nécessaires pour trouver des mots de

passe de longueurs variables :

	3 caractères	6 caractères	9 caractères
lettres minuscules	17'576	308'915'776	$5{,}4 \times 10^{12}$
lettres minuscules et chiffres	46'656	2'176'782'336	$1{,}0 \times 10^{14}$
minuscules, majuscules et chiffres	238'328	$5{,}6 \times 10^{10}$	$1{,}3 \times 10^{16}$

Sachant qu'un ordinateur personnel est capable de tester quelques millions de mots de passe par seconde, on voit qu'un mot de passe de seulement six caractères ne tiendrait pas très longtemps face à une telle attaque.

Comment choisir et retenir un bon mot de passe ?



La méthode phonétique

Cette méthode consiste à utiliser les sons de chaque syllabe pour créer une phrase facilement mémorisable.

Exemple : « j'ai acheté huit cd pour cent euros cet après-midi » donnera : ght8CD%E7am

La méthode des premières lettres

Utiliser les premières lettres d'une phrase en variant majuscules, minuscules et caractères spéciaux.

Exemple : « un tiens vaut mieux que deux tu l'auras » donnera : 1TvmQ2tl'@

Diversifier ses mots de passe

Opter pour une politique personnelle avec, par exemple, un préfixe pour chaque type d'activité, comme BANQUE-MonMotDePasse pour la banque, IMPOTS-MonMotDePasse pour les impôts. Cette astuce complexifie vos mots de passe et vous permet de les diversifier.

Vous trouverez les mots de passe les plus utilisés sur Wikipédia.

La société *Imperva*, spécialisée en sécurité de données, a publié un rapport portant sur l'analyse de 32 millions de mots de passe d'internautes dérobés en décembre 2009 sur le site communautaire RockYou.com et dévoilés sur le Web par la suite. Selon cette étude, 50 % des mots de passe analysés sont relatifs à des prénoms, des mots du dictionnaire ou des séquences de chiffres. 123456 est le mot de passe le plus commun (il sert à près de 291'000 comptes sur RockYou.com), devant 12345 (79'000), 123456789 (77'000), Password (62'000) et iloveyou (51'600). Environ 30 % des mots de passe contiennent six caractères ou moins. Selon Imperva, la simplicité de certains mots de passe expose les internautes à des cyberattaques automatisées.

Gestionnaire de mots de passe

Un gestionnaire de mots de passe est un « coffre-fort numérique » qui permet de stocker tous vos mots de passe dans un seul et unique endroit. Il ne vous restera plus qu'à mémoriser votre mot de passe maître (qui doit évidemment être très robuste).

Par exemple, Keepass[7] est gratuit et recommandé par l'État français.

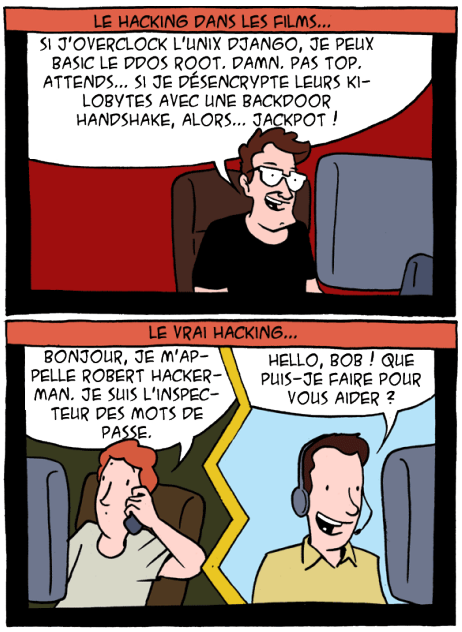
Diminuer les imprudences

Pour finir, il est utile de rappeler de ne pas stocker ses mots de passe à proximité de son ordinateur si il est accessible par d'autres personnes. L'écriture sur le post-it déposé sous le clavier est à proscrire par exemple, de même que le stockage dans un fichier de la machine.

En règle général, les logiciels proposent de retenir les mots de passe, c'est très tentant mais imprudent. Si votre ordinateur fait l'objet d'un piratage ou d'une panne, les mots de passe seront accessibles par le pirate ou perdus.

Un bon mot de passe doit être changé régulièrement et toujours en utilisant les méthodes décrites ci-dessus.

14.5. Piratage informatique



L'ingénierie sociale est une manipulation psychologique qui consiste à pousser la personne ciblée à agir d'une certaine façon, par exemple à divulguer des informations confidentielles, acheter un produit ou débloquer des fonds. Les ingénieurs sociaux espionnent l'environnement personnel de leur victime, usurpent des identités ou exploitent des caractéristiques humaines fondamentales, telles que la serviabilité, la confiance, la curiosité, la peur ou le respect de l'autorité, pour obtenir des informations confidentielles ou des services gratuits.

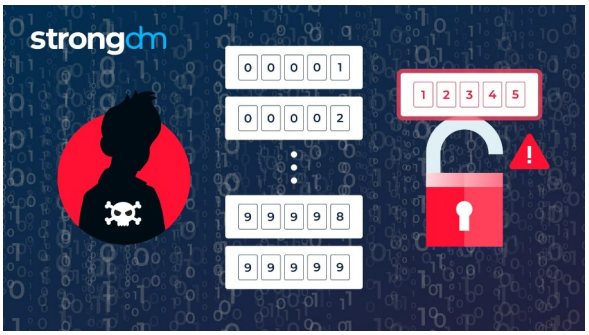
- Soyez prudents lorsqu'on vous pousse par téléphone, par courriel ou par message à effectuer une action.
- Ne vous laissez pas intimider ou mettre sous pression.
- Ne communiquez jamais par téléphone ou par courriel un mot de passe ou un code PIN. Aucun prestataire de services sérieux ne vous demandera votre mot de passe ou vos données d'accès.
- Ne communiquez sous aucun prétexte les codes ou les mots de passe que vous recevez par courriel ou par SMS.
- Ne laissez personne accéder à votre ordinateur, même si votre interlocuteur prétend travailler pour un prestataire de services de sécurité, un établissement financier, une entreprise d'assistance informatique ou autre.

7 https://keepass.info/

14.5.1. Attaque par force brute

Un mot de passe n'est jamais stockés tel quel sur un ordinateur, seule une empreinte du mot de passe est stockée sur l'ordinateur. Celle-ci résulte d'une fonction de hachage (voir § 14.3), qui est à sens unique. Les pirates ne peuvent donc pas inverser l'algorithme de hachage pour consulter votre mot de passe.

Pour tenter de trouver un mot de passe, l'attaquant va être obligé d'appliquer la fonction de hachage sur des millions de mots et les comparer avec l'empreinte dérobée, jusqu'à trouver une équivalence.L'attaque par force brute est une méthode utilisée en cryptanalyse pour trouver un mot de passe ou une clé. Il s'agit de tester, une à une, toutes les combinaisons possibles. Cette méthode de recherche exhaustive ne réussit que dans les cas où le mot de passe cherché est constitué de peu de caractères. Pour contrer cette méthode, il suffit simplement de choisir des mots de passe d'une grande longueur ou des clés suffisamment grandes.



C'est la même méthode qui est utilisée pour trouver la combinasion d'un cadenas.

Cette méthode est souvent combinée avec l'attaque par dictionnaire (voir paragraphe 14.5.2) pour obtenir de meilleurs résultats.

Conseils

Pour contrer des attaques par force brute, la meilleure solution est :

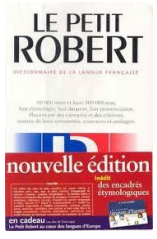
- d'allonger le mot de passe ou la clé si cela est possible (12 caractères au minimum) ;
- utiliser la plus grande gamme de symboles possibles (minuscules, majuscules, ponctuations, chiffres) ;
- faire en sorte que le mot de passe soit aléatoire ;
- et pour une sécurité optimum, empêcher de dépasser un nombre maximal d'essais en un temps ou pour une personne donnée.

Exercice 14.1

Vous avez accès à une liste d'empreintes MD5 de mots de passe.

Imaginez (et écrivez) un programme qui tentera de « cracker » ces empreintes (c'est-à-dire retrouver les mots de passe) en essayant toutes les combinaisons possibles.

Vous supposerez que le mot de passe est composé de six lettres minuscules.



14.5.2. Attaque par dictionnaire

L'attaque par dictionnaire est une méthode utilisée en cryptanalyse pour trouver un mot de passe ou une clé. Elle consiste à tester une série de mots de passe potentiels, les uns à la suite des autres, en espérant que le mot de passe utilisé pour le chiffrement soit contenu dans le dictionnaire. Si tel n'est pas le cas, l'attaque échouera.

Cette méthode repose sur le fait que de nombreuses personnes utilisent des mots de passe courants (par exemple un prénom, une couleur, le nom d'un animal, d'un héros, ...).

Contenu du dictionnaire et règles

Outre le contenu habituel d'un dictionnaire qui renferme un ensemble de mots, le dictionnaire peut être fortement amélioré en combinant les mots ou en appliquant certaines règles. Par exemple, pour chaque mot, on peut essayer de changer la casse de certaines lettres. Une autre astuce consiste à répéter deux fois le mot (par exemple « secretsecret »), dans l'espoir que l'utilisateur fasse appel à cette méthode peu sûre pour renforcer son mot de passe. On peut aussi générer des dictionnaires, par exemple pour des numéros de plaque, des numéros de sécurité sociale, des dates de naissance, etc.

Logiciels

Le logiciel **John the Ripper**[8], le plus connu dans ce domaine, est fourni avec une liste de règles qui permettent d'étendre l'espace de recherche et explorer les failles classiques dans l'élaboration des mots de passe ou des clés par les utilisateurs. Il existe des dictionnaires qui contiennent plusieurs millions de mots et qui ne sont pas fournis avec le logiciel. Citons aussi **L0phtcrack**[9] et **Cain & Abel**[10] qui fournissent des services similaires.

Ce type de logiciels n'est pas forcément utilisé dans un but malicieux. Ces outils peuvent être utiles pour les administrateurs qui veulent sécuriser les mots de passe et alerter les utilisateurs sur les risques encourus.

Exercice 14.2

Vous avez accès à une liste d'empreintes MD5 de mots de passe.

Imaginez (et écrivez) un programme qui tentera de « cracker » ces empreintes en utilisant une attaque par dictionnaire.

Comment savoir si mon mot de passe a déjà été découvert ?

Le site « Have I been pawned ? »[11] répertorie des millions de mots de passe déjà crackés dans le passé. Si le vôtre est dans la liste, changez-le sans tarder !

Exercice 14.3

Imaginez un usage du site https://haveibeenpwned.com/Passwords à des fins malveillantes.

14.5.3. Tables arc-en-ciel

Une table arc-en-ciel (appelée en anglais « *Rainbow Table* ») est une structure de données créée en 2003 à l'EPFL par Philippe **Oechslin** pour retrouver un mot de passe à partir de son empreinte. Les tables contiennent une grande quantité de chaînes qui proposent en alternance un mot de passe suivi de son empreinte. Une fonction de réduction, qui varie selon la position dans la table permet de recréer un autre mot de passe à partir de l'empreinte et ainsi de suite.

Philippe **Oechslin**

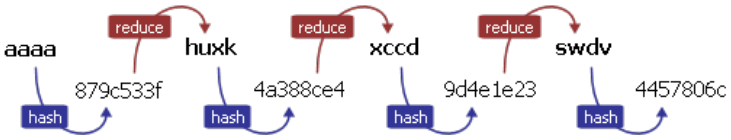
L'algorithme, déjà efficace avec des mots de passe correctement chiffrés l'est *a fortiori* avec les mots de passe de LAN Manager[12] : ils sont trouvés en l'espace de quelques secondes s'ils sont alphanumériques. Les tables peuvent être utilisées pour d'autres fonctions de hachage comme MD5 ou encore SHA-1, ces dernières sont toutefois nettement plus robustes du point de vue cryptographique que LAN Manager et nécessitent des tables plus grandes.

8 http://www.openwall.com/john/
9 http://www.l0phtcrack.com/
10 http://www.oxid.it/
11 https://haveibeenpwned.com/Passwords
12 **LAN Manager** (gestionnaire de réseau local) est un service serveur pour système d'exploitation développé par Microsoft en collaboration avec 3Com

Création de la table

Toute l'astuce des tables arc-en-ciel consiste à calculer à partir d'une empreinte un mot de passe. Attention, pas **LE** mot de passe qui correspond à l'empreinte (une telle fonction n'existe pas), mais **UN** mot de passe. On dit que l'on réduit l'empreinte. La seule chose que l'on demande à cette fonction de réduction est d'être cohérente, c'est-à-dire de toujours retourner le même mot de passe quand on lui donne la même empreinte en paramètre.

Le principe de génération d'une table arc-en-ciel est donc le suivant : on part d'un mot de passe, on calcule son empreinte, puis on calcule un nouveau mot de passe à partir de l'empreinte, on calcule l'empreinte de ce mot de passe, et on répète l'opération un certain nombre de fois. Ensuite on stocke dans la table uniquement le mot de passe initial et l'empreinte finale.

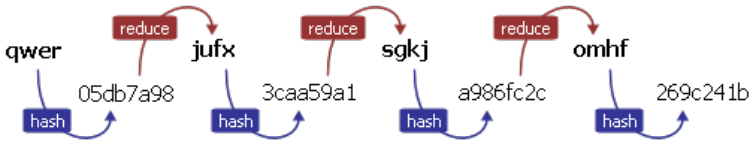


Rainbow Table

aaaa	4457806c

Puis on recommence le processus. On choisit un nouveau mot de passe, et on construit une nouvelle chaîne.

Nous venons de générer une mini table arc-en-ciel contenant deux lignes où chaque ligne représente une chaîne de quatre mots de passe.



Rainbow Table

aaaa	4457806c
qwer	269c241b

Utilisation de la table pour cracker des empreintes

Commençons par le plus facile : essayons de cracker l'empreinte « 269c241b ». Cette empreinte figure directement dans la table, à la seconde ligne, et est associée avec le mot de passe « qwer ». On sait donc que le mot de passe qui correspond à cette empreinte est le quatrième de la chaîne. Comme on ne l'a pas stocké dans la table, nous allons régénérer la chaîne comme lors de la création de la table. On prend donc le mot de passe initial, on le fait passer dans la fonction de hachage, ce qui donne « 05db7a98 ». Ensuite on fait passer cette empreinte dans la même fonction de réduction que celle qui a servi à générer la table. Elle retourne donc le deuxième mot de passe de la chaîne (« jufx ») que l'on hache puis réduit pour trouver le troisième mot de passe (« sgkj »), que l'on hache puis réduit pour donner le quatrième mot de passe : « omhf ». Pour être sûr de nous, on hache « omhf », ce qui donne « 26c241b ». Nous avons cracké l'empreinte « 26c241b » : le mot de passe est « omhf ».

Essayons maintenant de cracker l'empreinte « 9d4e1e23 ». Elle ne figure pas dans la table. Essayons de calculer une autre empreinte à partir de « 9d4e1e23 ». Un appel à la fonction de réduction nous donne « swdv », qui si on le passe à la fonction de hachage renvoie « 4457806c ». Oh surprise... cette empreinte figure dans la table à la première ligne. Prenons donc le mot de passe initial « aaaa », et comme pour le premier cas, reconstituons la chaîne : 2 coups de hachage/réduction nous donnent le mot de passe « xccd » qui une fois passé au hachoir donne « 9d4e1e23 ». Mission accomplie. Le mot de passe correspondant à notre empreinte est « xccd ».

Une dernière empreinte à cracker : « 05db7a98 »

1. L'empreinte ne figure pas dans la table.

2. reduce(hash(05db7a98)) donne 3caa59a1. Qui ne figure pas non plus dans la table. Recommençons.
3. reduce(hash(3caa59a1)) donne a986fc2c. Encore perdu. Recommençons.
4. reduce(hash(a986fc2c)) donne 269c241b, qui figure sur dans la table, à la première ligne. De plus comme nous avons tourné 3 fois pour trouver, et que la table contient des chaînes de longueur 4, le mot de passe recherché est le premier de la chaîne.
5. hash(qwer) donne 05db7a98. C'est gagné !

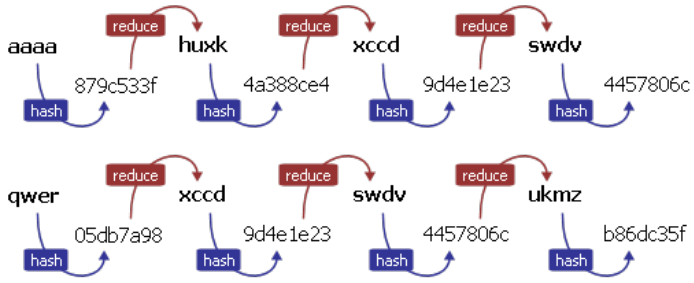
Pour stocker 100 millions de mots de passe, il suffit donc par exemple de générer une Rainbow table contenant 100'000 lignes avec des chaînes de longueur 1000. On stocke donc dans un fichier de 2 Mo une table qui pèserait 2 Go dans le cas d'un simple dictionnaire… si on a de la chance et pas de collisions !

Collisions

En pratique, les fonctions de réductions provoquent des collisions. Une collision survient quand la fonction de réduction retourne le même mot de passe pour deux empreintes différentes. Cela survient forcément car il y a toujours plus d'empreintes possibles que de mots de passe.

- Nombre de mots de passe de 8 caractères alphanumériques : $(26+26+10)^8 = 62^8 \cong 2.2\times10^{14}$
- Nombre d'empreintes SHA-1 (160 bits) : $2^{160} \cong 1.5\times10^{48}$

Reprenons l'exemple précédent et admettons que nous n'ayons pas eu de chance. Cela donnerait à peu près cela :

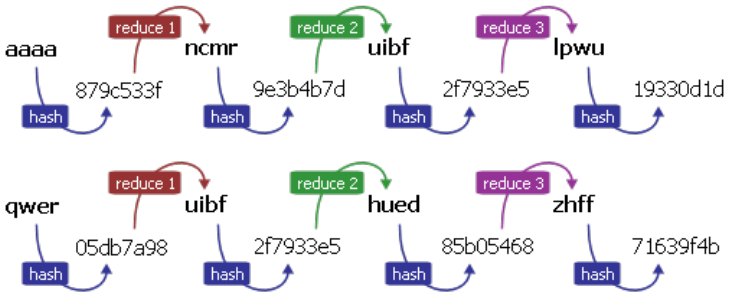


Rainbow Table

aaaa	4457806c
qwer	b86dc35f

La collision se produit lors de la réduction de 05db7a98. La fonction retourne « xccd », comme pour l'empreinte 4a388ce4. Résultat : au lieu de contenir 8 mots de passe distincts, notre table n'en contient que 6. Nous avons des chaînes qui sont dupliquées à partir de la collision (xccd → swdv).

Pour diminuer cet effet indésirable, les tables arc-en-ciel utilisent non pas une mais plusieurs fonctions de réduction (que l'on peut représenter par des couleurs différentes, d'où le nom « arc-en-ciel »). En fait, il y a une fonction de réduction différente par « colonne » :



Rainbow Table

aaaa	19330d1d
qwer	71639f4b

Cette fois, même si une collision se produit, on ne se retrouve pas avec des chaînes dupliquées : on a seulement un mot de passe dupliqué. Évidemment, si on n'a vraiment pas de chance, on peut tomber sur une collision dans la même colonne. En pratique, il s'avère que ce cas n'arrive que rarement avec de longues chaînes.

Contre-mesures

L'efficacité des tables diminue de façon significative lorsque les fonctions de hachage sont combinées à un sel. Dans le cadre d'un système de mots de passe, le sel est une composante aléatoire ou un compteur qui change en fonction de l'utilisateur. Si deux utilisateurs ont le même mot de passe, le sel permet d'éviter que les empreintes soient identiques. De manière informelle, le sel consiste en une opération du type :

```
empreinte = h(mot_de_passe + sel)
```

où l'opération + peut être une concaténation ou une opération plus complexe.

Cette mesure augmente la complexité de l'attaque : il faut non seulement inverser la fonction grâce aux tables mais il faut encore explorer l'ensemble des possibilités induites par la présence du sel. Si l'attaque réussit, il faut encore retirer le sel du mot de passe.

En pratique, certaines applications n'utilisent pas de sel et sont vulnérables. En outre, le sel doit avoir une longueur suffisante pour augmenter sensiblement la complexité.

Dans le système GNU/Linux, la fonction de hachage utilisée est MD5 avec un sel de 8 caractères en ASCII ce qui rend l'attaque impossible en pratique.

Sources

[1] Wikipédia, « Portail de la sécurité informatique », <http://fr.wikipedia.org/wiki/Portail:Sécurité_informatique>

[2] Comment ça marche, « Sécurité/Législation », <http://www.commentcamarche.net/contents/security/>

[3] « Virus informatiques », <http://vaccin.sourceforge.net/index.php>

[4] BDM/Media, « Un hacker réclame 500 euros en Bitcoin en utilisant votre adresse email : ne cédez pas aux menaces, c'est juste du « spoofing », <https://www.blogdumoderateur.com/hacker-500-euros-bitcoin-email/ >, mai 2019

[5] Mieux coder, « Rainbow tables », <http://www.mieuxcoder.com/2008/01/02/rainbow-tables/>