Attaque par force brute sur mot de passe.

(FBA Force Brut Attac).

(première partie information de Wikipedia suivi par optimization)

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 est en général considérée comme la plus simple. Elle permet de casser tout mot de passe en un temps fini indépendamment de la protection utilisée, mais le temps augmente avec la longueur du mot de passe. En théorie la complexité d'une attaque par force brute est une fonction exponentielle de la longueur du mot de passe, la rendant en principe impossible pour des mots de passe de longueur moyenne. En pratique des optimisations heuristiques peuvent donner des résultats dans des délais beaucoup plus courts.

Cette méthode est souvent combinée avec l'attaque par dictionnaire pour trouver la solution plus rapidement.

Complexité théorique de l'attaque

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 ;
- 36^N si le mot de passe mélange des chiffres et des lettres de l'alphabet totalement en minuscules ou en 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.

Type	1 caractère	3 caractères	6 caractères	9 caractères
lettres minuscules	26	17 576	308 915 776	$5{,}4 \times 10^{12}$
lettres minuscules et chiffres	36	46 656	2 176 782 336	$1{,}0 \times 10^{14}$
minuscules, majuscules et chiffres	62	238 328	$5{,}6 \times 10^{10}$	$1{,}3 \times 10^{16}$

Un ordinateur personnel est capable de tester plusieurs centaines de milliers voire quelques millions de mots de passe par seconde. Cela dépend de l'algorithme utilisé pour la protection mais on voit qu'un mot de passe de seulement 6 caractères, eux-mêmes provenant d'un ensemble de 62 symboles (minuscules ou majuscules accompagnés de chiffres), ne tiendrait pas très longtemps face à une telle attaque.

L'attaque par force brute pour casser un mot de passe.

Le principe général de l'attaque par force brute reste toujours de tester l'ensemble des mots de passe possibles, cependant l'ordre de test peut être optimisé afin d'obtenir de meilleurs rendements qu'une attaque par ordre alphabétique.

Attaque par dictionnaire

Plutôt que d'utiliser des chaînes de caractère aléatoires comme mot de passe, les utilisateurs ont tendance à utiliser des mots courants plus faciles à retenir. Or, s'il existe un nombre important de combinaisons aléatoires pour une chaîne de longueur donnée, le nombre de mots présents dans un ou plusieurs langages est beaucoup plus faible (à titre d'exemple l'Académie française estime que les dictionnaires encyclopédiques comptent environ 200 000 mots[1]). Connaissant ce phénomène culturel, il peut être judicieux de tester ces mots courants et leurs dérivés (y compris argot, dialectes, mots avec fautes d'orthographe courante…) en priorité.

Optimisation par observations statistiques

De manière générale, pour développer le principe de l'attaque par dictionnaire, l'attaquant peut tirer parti du fait qu'en l'état actuel des connaissances, il est toujours possible en observant de grands échantillons de mots de passe d'identifier des tendances permettant un résultat généralement meilleur qu'une recherche alphabétique ou aléatoire.

Exemple 1:

On vous demande de choisir un mot de passe 10 charactères avec majuscule, minuscule, des nombres et caractères spéciaux.

La majorité des personnes va choisir un prénom ou le nom de son animal de compagnie.

De plus elle va mettre en majuscule la première lettre, puis a la fin du nom elle choisira des chiffres (probablement le numéro de sa rue) et si besoin d'un caractères spécial il sera en en dernier et sera probablement la virgule, le + ou le - , ou le point d'exclamation.

Exemple 2 :

Ou si on vous demande un mot de passe de 6 chiffres beaucoup de personnes mettrons une date : de naissance, de mariage ou autres.

Accroissement de la puissance de calcul

Outre ces améliorations algorithmiques, l'attaque peut également être accélérée en augmentant la puissance de calcul matérielle consacrée à celle-ci.

Défense contre l'attaque par force brute sur mot de passe.

Limiter le nombre d'essai.

La première solution et la plus efficace est la limitation de nombre d'essai comme avec le code secret de la carte bleue.

Utilisation de mots de passe robustes.

La défense consiste à renforcer le mot de passe en évitant les écueils qu'exploitent les attaques par force brute optimisée. Renforcer la force brute du mot de passe consiste à :

- allonger le mot de passe ou la clé si cela est possible ;
- utiliser la plus grande gamme de symboles possibles (minuscules, majuscules, ponctuations, chiffres) ; l'introduction de caractères nationaux (Â, ÿ…) rend plus difficile le travail des pirates (mais parfois aussi l'entrée de son mot de passe quand on se trouve à l'étranger).

Limitation temporelle des connexions

La principale méthode pour neutraliser la puissance de calcul d'un attaquant consiste à limiter les tentatives possibles dans le temps. La méthode la plus restrictive et la plus sûre (qu'on retrouve sur les cartes bancaires en France) consiste à n'autoriser qu'un nombre limité d'erreurs avant verrouillage du système. Des méthodes moins contraignantes peuvent être de limiter le nombre de tentatives par unité de temps.

Deux brevets principaux existent à ce sujet :

- Un des Laboratoires Bell consistant à doubler le temps d'attente après chaque essai infructueux, pour le faire redescendre ensuite en vol plané après un certain temps sans attaques ;
- Un de la compagnie IBM consistant à répondre « Mot de passe invalide » après N essais infructueux en un temps T, *y compris si le mot de passe est valide*[3] : le pirate a

alors toutes les chances de rayer de façon erronée le mot de passe valide en le considérant invalide. De plus, cette méthode empêche toute attaque visant à un déni de service pour l'utilisateur.

Durcissement de connexions.

En plus du mot de passe on peut durcir et limiter les tentatives de connexion avec l'utilisation de **CAPTCHA**. Ces dispositifs peuvent poser des difficultés significatives pour une machine tout en restant acceptables pour un utilisateur humain.

Exemple : afficher des images et demander à l'utilisateur de cliquer sur les voitures ou les trains affichés.

Renouvellement des mots de passe

Une solution peut consister à limiter la durée de validité des mots de passe à une durée inférieure à celle estimée pour leur cassage en les renouvelant à intervalles réguliers. Ceci peut passer soit par une politique de sécurité informatique appliquée avec rigueur pour des périodes de renouvellement jusqu'à quelques jours ou par des dispositifs physiques *token* pour des fréquences de renouvellement très élevées.

Utiliser un tiers pour valider la connexion.

C'est actuellement la solution sécuritaire la plus en vogue.

Pour valider un mot de passe on demande une validation complémentaire sur un appareil tiers (exemple : téléphone) qui doit appartenir au propriétaire. (Appareil au préalable déjà enregistré).