Introduction à la cryptographie
TD3 – Mots de passe

Cécile Pierrot

14 Janvier 2019

1 Attaquer un mot de passe

Question 1. Calculez ou estimez le nombre de mots de passe possibles dans les cas suivants :
- le mot de passe est un prénom ;
- c'est un mot du dictionnaire ;
- il est composé de quatre chiffres ;
- il est composé de huit caractères alphanumériques.

Question 2. Calculez ou estimez le nombre maximal d'essais que l'on peut autoriser pour rentrer un mot de passe si l'on veut limiter la probabilité de succès d'une attaque directe à 2^{-11} et que le mot de passe est choisi suivant l'un des types précédents.

Question 3. Calculez ou estimez le temps nécessaire pour pénétrer dans un système qui est protégé par un mot de passe de chacun des types précédents si l'on suppose que le système attend une seconde avant de vous redemander le mot de passe.

Afin de ne pas stocker les mots de passe des utilisateurs en clair sur un système, il est recommandé d'utiliser une *fonction de hachage cryptographique*. Il s'agit d'une fonction $H : \{0,1\}^* \to \{0,1\}^n$ qui, à partir d'une chaîne de bits quelconque m, produit un *haché* (ou *empreinte*) $h = H(m)$ de taille fixe (n bits). Une telle fonction de hachage est dite *cryptographique* lorsqu'elle vérifie un certain nombre de propriétés. Entre autres, elle doit être dure à inverser : étant donné $h \in \{0,1\}^n$, il doit être difficile de trouver m tel que $H(m) = h$.
Ainsi, plutôt que de stocker directement un mot de passe *pass*, un système pourra se contenter de ne stocker que son haché $hash = H(pass)$.

Question 4. Sur un tel système, comment la vérification d'un mot de passe saisi par l'utilisateur est-elle effectuée ?

Question 5. Calculez ou estimez le temps nécessaire pour pénétrer dans un système qui est protégé par un mot de passe de chacun des types précédents si l'on suppose que le système compare les hachés et :
- vous possédez le haché du mot de passe et vous faites une attaque par force brute sur ce haché ;
- vous possédez une liste des hachés des mots de passe de L utilisateurs du système, et vous devez retrouver un mot de passe parmi ceux-ci.

Remarque : on pourra supposer que le temps de calcul du haché d'un mot de passe sur un ordinateur actuel est de l'ordre d'une microseconde.

Certaines fonctions de hachage comme *bcrypt* ou *scrypt* peuvent aussi être utilisées pour protéger des mots de passe. La particularité de ces fonctions est qu'un calcul de haché de mot de

passe est beaucoup plus coûteux (entre 10 000 et 100 000 fois plus cher) qu'avec les fonctions de hachage cryptographiques classiques (comme SHA-256).

Question 6. Quel est l'intérêt d'utiliser une telle fonction ? Quel impact (positif ou négatif) aura-t-elle sur un système si celui-ci l'utilise pour protéger les mots de passe de ses utilisateurs ? Quel impact aura-t-elle sur une attaque ?

Le *salage* (*salting* en anglais) est une technique classique employée afin de contrer l'attaque précédente. Elle consiste à rajouter une chaîne de caractères aléatoire (le *sel*) au mot de passe avant de calculer son haché. Cette chaîne est alors stockée en clair avec le haché, afin de toujours pouvoir effectuer la vérification du mot de passe.
Pour chaque mot de passe *pass* que l'on veut stocker, on procède ainsi de la manière suivante :

1. choisir aléatoirement *salt*, une chaîne de ℓ caractères (typiquement, $\ell = 8$) ;
2. calculer le haché $hash = H(salt \,\|\, pass)$ (l'opérateur $\|$ désigne la concaténation de chaînes de caractères, et H une fonction de hachage cryptographique) ;
3. stocker le couple $(salt, hash)$.

Question 7. Avec ce système, comment la vérification d'un mot de passe saisi par l'utilisateur est-elle effectuée ?

Question 8. Reprenez à nouveau la question 5, en supposant cette fois-ci que vous possédez une liste de hachés *salés* des mots de passe de L utilisateurs du système. Concluez sur l'intérêt de cette technique.

2 Choisir un mot de passe

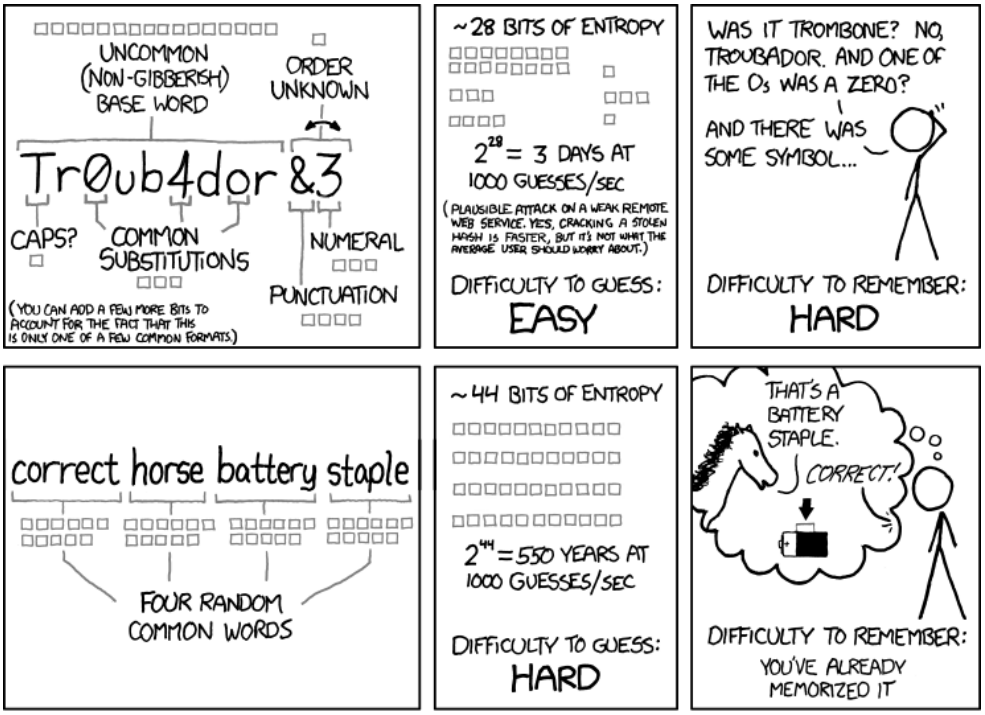
Supposons maintenant que vous souhaitiez choisir un mot de passe qui vous permette de résister à une attaque par force brute.

Question 1. Combien de caractères doit comporter votre mot de passe si vous souhaitez une sécurité en 2^{80} ? en 2^{100} ? en 2^{128} ?

Question 2. Comment doivent être choisis les mots de passe si vous souhaitez vraiment que la complexité des attaques soit celle que vous visez ?

3 xkcd #936 – Password Strength

Question 1. Au regard des questions précédentes, lisez et commentez le webcomic suivant.



(Source : R. Monroe, *xkcd #936 – Password Strength*, `https://xkcd.com/936/`)

4 Mots de passe compromis

Le 14 décembre 2009, le site d'informations spécialisées TechCrunch publiait un article sur le piratage de RockYou, dont voici un extrait :

> *Over the weekend, the security firm Imperva issued a warning to RockYou that there was a serious SQL Injection flaw in their database. Such a flaw could grant hackers access to the service's entire list of user names and passwords in the database, they warned. Imperva said that after it notified RockYou about the flaw, it was apparently fixed over the weekend. But that's not before at least one hacker gained access to what they claim is all of the 32 million accounts. 32,603,388 to be exact. The best part? The database included a full list of unprotected plain text passwords. And email addresses. Wow.*
>
> (Source : M.G. Siegler, *One Of The 32 Million With A RockYou Account? You May Want To Change All Your Passwords. Like Now.*, 2009, `http://techcrunch.com/2009/12/14/rockyou-hacked/`)

Question 1. Commentez cette vulnérabilité à la lumière de ce que vous avez appris sur les mots de passe.

Question 2. Combien de mots de passe de huit caractères sont possibles si l'on exclut les mots de passe compromis ?

Question 3. Si l'on choisit un mot de passe de huit caractères de manière strictement aléatoire, quelle est la probabilité que le mot de passe choisi appartienne à l'ensemble compromis ?

Question 4. Si l'on prend un mot de passe de huit caractères de son choix, que pouvez-vous dire de la probabilité qu'il appartienne à l'ensemble compromis ?

En fait, la base de données piratée contenait seulement 14 344 391 mots de passe différents, dont voici les 40 plus fréquents (avec leur nombre d'occurrences) :

`123456`	290 729	`nicole`	16 227	`111111`	13 272	`friends`	10 731
`12345`	79 076	`daniel`	15 308	`iloveu`	13 134	`butterfly`	10 560
`123456789`	76 789	`babygirl`	15 163	`000000`	13 028	`purple`	10 547
`password`	59 462	`monkey`	14 726	`michelle`	12 714	`angel`	10 508
`iloveyou`	49 952	`lovely`	14 331	`tigger`	11 761	`jordan`	10 167
`princess`	33 291	`jessica`	14 103	`sunshine`	11 489	`liverpool`	9 764
`1234567`	21 725	`654321`	13 984	`chocolate`	11 289	`justin`	9 708
`rockyou`	20 901	`michael`	13 981	`password1`	11 112	`loveme`	9 704
`12345678`	20 553	`ashley`	13 488	`soccer`	10 836	`fuckyou`	9 610
`abc123`	16 648	`qwerty`	13 456	`anthony`	10 755	`123123`	9 516

Question 5. En tant qu'attaquant, quel mot de passe essayeriez-vous en premier ? Quelle serait la probabilité de succès ?

Question 6. Sachant que la somme des nombres d'occurrences de ces 40 mots de passe est 1 034 098, quelle est la probabilité de succès d'une attaque par dictionnaire qui n'essayerait que ces mots de passe ?

Question 7. Quelle(s) précaution(s) de sécurité l'affaire RockYou vous inspire-t-elle ?