En quoi les politiques de création de mots de passe peuvent les affaiblir ?

Mathieu Valois, Patrick Lacharme, Jean-Marie Le Bars

En quoi les politiques de création de mots de passe peuvent les affaiblir ?

Mathieu Valois, Patrick Lacharme, Jean-Marie Le Bars
mathieu.valois@unicaen.fr, patrick.lacharme@ensicaen.fr, jean-marie.lebars@unicaen.fr

19 novembre 2018

Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC

Sommaire

Contexte

Contexte

Rôle

- dans la peau d'une RSSI
- gestion d'un parc de machines et de services
- objectif : améliorer sécurité et utilisabilité des mots de passe des utilisateurs

Acteurs/Menaces

- deux catégories d'acteurs :
  - la RSSI
  - les utilisateurs
- deux types de menaces sur les mots de passe :
  - attaques en ligne
  - attaques hors-ligne
- Non abordés :
  - autres acteurs : personnel d'entretien
  - autres attaques : physiques, phishing, social engineering

Contexte

Rôle

- dans la peau d'une RSSI
- gestion d'un parc de machines et de services
- objectif : améliorer sécurité et utilisabilité des mots de passe des utilisateurs

Acteurs/Menaces

- deux catégories d'acteurs :
  - la RSSI
  - les utilisateurs
- deux types de menaces sur les mots de passe :
  - attaques en ligne
  - attaques hors-ligne
- Non abordés :
  - autres acteurs : personnel d'entretien
  - autres attaques : physiques, phishing, social engineering

Contexte

Attaques en ligne

- Objectif : un attaquant tente de se connecter à un ou plusieurs comptes directement sur l'interface de connexion

Attaques hors ligne

- Contexte : l'attaquant s'est procuré le fichier contenant la liste des mots de passe protégés des utilisateurs
- Objectif : il souhaite retrouver un maximum de mots de passe

Contexte : mesures de protection

Attaque en ligne

Protection via infrastructure :

- bannir l'adresse IP après N essais
- verrouiller le compte temporairement après N essais
- mettre en place un captcha

Contexte : mesures de protection

Fonctions de hachage

Préconiser des fonctions de hachage coûteuses (bcrypt, Argon2, scrypt, ...) $\rightarrow$ pas suffisant car les mots de passe faibles restent attaquables

Renforcement de la robustesse

Plusieurs moyens :

- politiques de composition du mot de passe (8 caractères et 4 classes)
- listes noires (bannir les 5000 plus courants)
- mesure de la robustesse (interdire les mdps faibles)

Contexte : les politiques de composition I

Définition

Ensemble de règles que doit satisfaire impérativement un mot de passe pour être accepté

Exemples

- + de 8 caractères
- au moins une majuscule
- au moins un chiffre
- au moins un caractère spécial

Contexte : les politiques de composition II

Principe : agrandir l'espace de recherche

- augmenter la taille de l'alphabet
- augmenter la taille du mot

Inconvénients

- souvent frustrantes car trop restrictives
- basées sur aucune étude scientifique

Problème : renforcer la robustesse ?

Comment renforcer la robustesse ? I

Problème

les utilisateurs n'appliquent pas les règles de manière uniforme. Entraîne l'apparition de structures.

Structure

Induit par le comportement partagé par des utilisateurs.

Exemples :

- les mots de passe commencent par une majuscule
- les lettres d'un mot sont remplacés par des chiffres (l33t)
- les mots de passe contiennent des dates

Comment renforcer la robustesse ? II

Exploitation de ces structures

- les attaquants vont se servir de ces structures pour générer des mots de passe
- trop de règles de composition affaiblissent les mots de passe

Exemple : Attaques basées sur les chaînes de Markov I

Pour un mot $c_1...c_m$, sa probabilité

$$P(c_1...c_m) = P(c_1) \times P(c_2|c_1) \times P(c_3|c_1c_2) \times ... \times P(c_m|c_1...c_{m-1})$$

Chaînes de Markov : le modèle

Modélisation mathématique de la composition des mots et des textes d'une langue. Elle estime la probabilité d'apparition d'une lettre en fonction des n précédentes

$$P(c_1...c_m) \approx P(c_1...c_{n-1}) \times \prod_{i=n}^{m} P(c_i|c_{i-n+1}...c_{i-1})$$

n : paramètre du modèle, fixe la mémoire du processus.

En général, entre 2 et 5.

Exemple : Attaques basées sur les chaînes de Markov II

Utilisation

- pour générer un nouveau mot : tirage aléatoire de la première lettre, puis tirages aléatoires de chacune des suivantes en fonction des n précédentes (chaîne)
- *ex :* c, ch, chi, chin, chine

Efficacité

- le choix des lettres n'est pas uniforme dans une langue
- *ex :* en français $P(u|q) > P(w|q)$, $P(e|qu) > P(a|qu) > P(o|qu)$.
- ce modèle probabiliste fonctionne bien sur les textes

Exemple : Attaques basées sur les chaînes de Markov III

Méthode

1. l'attaquant entraîne ces modèles sur un corpus de mots de passe déjà trouvés (ou sur un corpus de textes de la langue)
2. il utilise un algorithme qui calcule les fréquences d'apparition des suites de n lettres
3. il génère des nouveaux mots respectant les fréquences du corpus
4. il s'en sert pour attaquer les mots de passe

Politiques de composition : une bonne solution ?

Politiques de composition

Une méthode efficace si

- l'attaque est la recherche exhaustive (force brute)
- et les lettres sont choisies uniformément

Une méthode peu efficace si

- les humains ne choisissent pas aléatoirement les lettres
- des attaques bien plus efficaces que la force brute (Modèles de Markov, modèles basées sur les grammaires, ...) car le choix des lettres non uniforme

Bilan

Politiques peu efficaces, voire même contre-productives si elle donnent des indications à l'attaquant, réduisant ainsi l'espace de recherche.

Faut-il bannir les politiques de composition ?

Bannir les politiques de composition ?

On s'intéresse ici à 2 études dont le but est de mesurer l'efficacité des politiques de composition des mots de passe :

- Can long passwords be secure and usable ? [3]
- Diversify to Survive : Making Passwords Stronger with Adaptive Policies [2]

Can long passwords be secure and usable ? [3] I

Objectifs

- Mesurer l'impact des politiques de composition de mot de passe sur la sécurité et l'utilisabilité
- est-ce que les politiques complexes sont mieux que des politiques très simples ?

Méthodologie

- étude sur 8000 participants
- les participants doivent construire un mot de passe sous différentes contraintes et le réutiliser 3 jours plus tard
- ils doivent rapporter leur ressenti en terme de sécurité et d'utilisabilité

Can long passwords be secure and usable ? [3] II

Résultat sur les politiques

- Bonne pratique : mots de passe longs avec peu de prérequis (16+ chars).
- Mauvaise pratique : politiques restrictives (8+ chars avec 4 classes)

Utilisateurs frustrés

Les utilisateurs sont frustrés des règles trop restrictives donc les appliquent de manière minimale et prédictible

Exemples de politiques efficaces

- basic20 : mots de longueur 20 et plus
- 2word16 : 2 mots séparés par un espace totalisant au moins 16 caractères

Diversify to Survive : Making Passwords Stronger with Adaptive Policies [2] I

Objectifs

Mesure de l'impact d'une politique adaptative sur la sécurité et l'utilisabilité

Méthodologie

- étude sur 2600 participants
- les participants doivent construire un mot de passe sans aucune indication de robustesse
- le serveur calcule des masques les mots de passe (classe de chaque caractère pour chaque position) : $Masque(P@ssword12) = USLLLLLLDD$
- le serveur bannit un masque trop utilisé
- il rejette les nouveaux mots ayant un masque interdit. Il indique parfois comment les améliorer

Diversify to Survive : Making Passwords Stronger with Adaptive Policies [2] II

Résultats

- les politiques adaptatives offrent une bien meilleure sécurité avec peu d'inconvénients sur l'utilisabilité
- elles doivent être envisagées quand il y a beaucoup d'utilisateurs
- apporter un feedback n'améliore pas la construction des mots de passe. En effet il est suffisant de demander à l'utilisateur de choisir un autre masque

Solutions

Que faire ?

Méthodes préconisées

- des politiques simples : basic20, 2word16
  → facile à mettre en place et plus efficaces que les complexes
- des mesures de robustesse : modèles probabilistes (PCFG, chaînes de Markov)
  → peu pratique à mettre en place
- zxcvbn [4] : mesure de robustesse proposée par Dropbox.
  → facile à mettre en place
- pam_pathwell [1] : module PAM et et mesure adaptative (prend en compte les nouveaux MDP).
  → contrôle plus fin des politiques

Méthodes à éviter

- indiquer comment renforcer son mot de passe car les modifications prévisibles
- les politiques complexes sont à éviter car contre-productives

Mesure de robustesse : zxcvbn

Principe

Estimer la taille des espaces de recherche parcourus par l'attaquant pour trouver le mot

- repose sur la "tokenization" : découpe le mot en "tokens" pour les attaquer plus facilement
  $\rightarrow$ "dropbox", "2016"
- attaques simulées : force brute, keyboard walking, dictionnaire, force brute sur dates, répétitions, substitution l33t.

Mesure de robustesse : zxcvbn

Points forts

- tient compte de plusieurs attaques
- peu coûteux en ressources
- possibilité de personnaliser la puissance de l'attaquant
- code source disponible (dans la plupart des langages) et lisible donc possibilité de personnaliser selon besoins

Point faible

- pas adaptatif

Actuellement une méthode très satisfaisante.

Conclusion

Conclusion

Politiques à proscrire

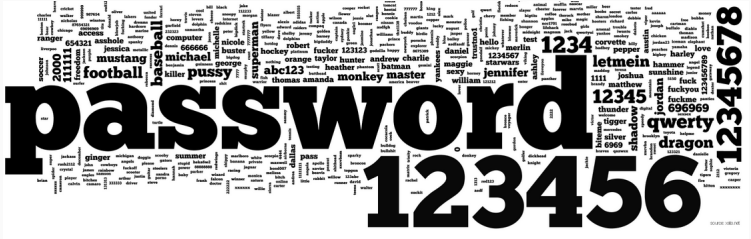
- les politiques trop complexes car contre-productives

Politiques à préconiser

- utiliser une politique simple qui fonctionne bien (basic20, 2word16)
- zxcvbn en exigeant un score minimum sans donner d'indication
- pam_pathwell : solution plus fine et sécurisée à l'échelle mais plus difficile à mettre en oeuvre

Merci

Merci de votre attention. Des questions ?



Références I


KoreLogic.
libpathwell.
https://github.com/KoreLogicSecurity/libpathwell, 2015.


Segreti, S. M.
Diversify to survive : Making passwords stronger with adaptive policies.
USENIX.


Shay, R., Komanduri, S., Durity, A. L., Huh, P. S., Mazurek, M. L., Segreti, S. M., Ur, B., Bauer, L., Christin, N., and Cranor, L. F.
Can long passwords be secure and usable ?
In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2014), ACM, pp. 2927–2936.

Références II


Wheeler, D. L.
Zxcvbn : Low-budget password strength estimation.
In *Proc. USENIX Security* (2016).