Sécurité des Applications

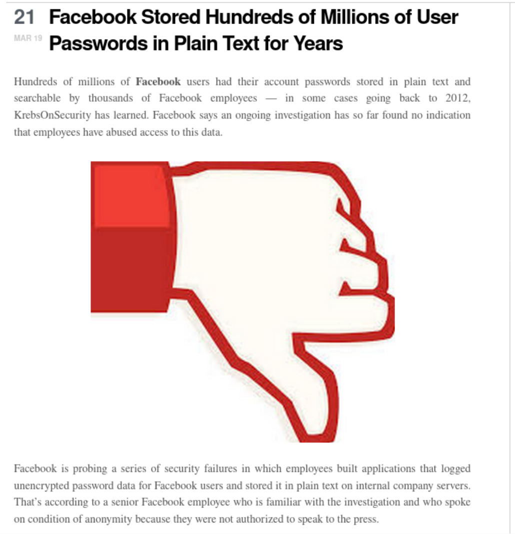
Passwords

Stéphane Küng

January 5, 2021

Passwords

Facebook



21 MAR 19

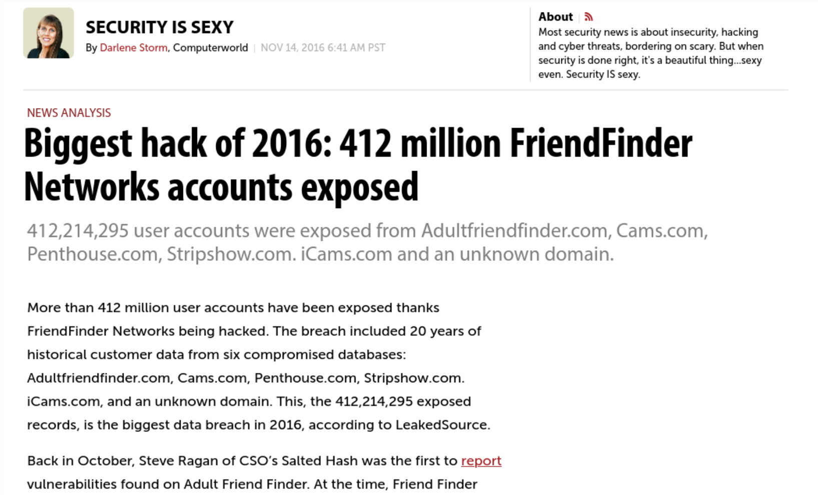
Facebook Stored Hundreds of Millions of User Passwords in Plain Text for Years

Hundreds of millions of **Facebook** users had their account passwords stored in plain text and searchable by thousands of Facebook employees — in some cases going back to 2012, KrebsOnSecurity has learned. Facebook says an ongoing investigation has so far found no indication that employees have abused access to this data.

Facebook is probing a series of security failures in which employees built applications that logged unencrypted password data for Facebook users and stored it in plain text on internal company servers. That's according to a senior Facebook employee who is familiar with the investigation and who spoke on condition of anonymity because they were not authorized to speak to the press.

- May be stored as ClearText

Divers sites de rencontre



SECURITY IS SEXY

By Darlene Storm, Computerworld | NOV 14, 2016 6:41 AM PST

About

Most security news is about insecurity, hacking and cyber threats, bordering on scary. But when security is done right, it's a beautiful thing...sexy even. Security IS sexy.

NEWS ANALYSIS

Biggest hack of 2016: 412 million FriendFinder Networks accounts exposed

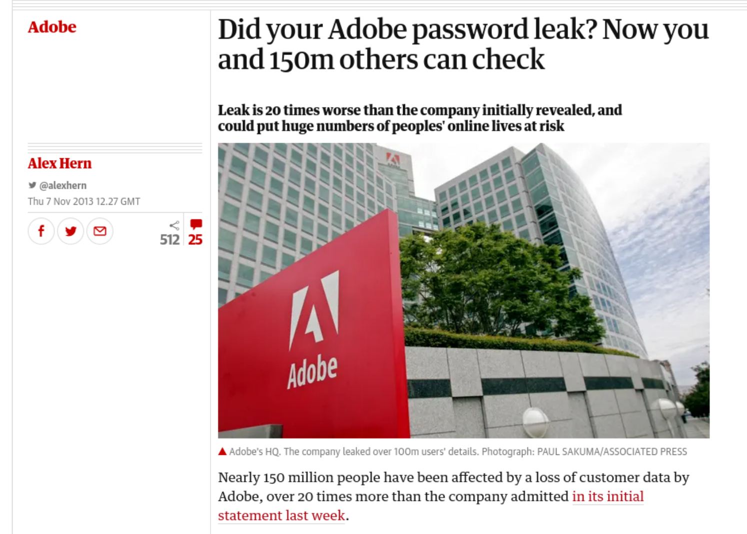
412,214,295 user accounts were exposed from Adultfriendfinder.com, Cams.com, Penthouse.com, Stripshow.com. iCams.com and an unknown domain.

More than 412 million user accounts have been exposed thanks FriendFinder Networks being hacked. The breach included 20 years of historical customer data from six compromised databases: Adultfriendfinder.com, Cams.com, Penthouse.com, Stripshow.com. iCams.com, and an unknown domain. This, the 412,214,295 exposed records, is the biggest data breach in 2016, according to LeakedSource.

Back in October, Steve Ragan of CSO's Salted Hash was the first to report vulnerabilities found on Adult Friend Finder. At the time, Friend Finder

- SHA-1

Adobe



Adobe

Did your Adobe password leak? Now you and 150m others can check

Leak is 20 times worse than the company initially revealed, and could put huge numbers of peoples' online lives at risk

Alex Hern

@alexhern

Thu 7 Nov 2013 12.27 GMT

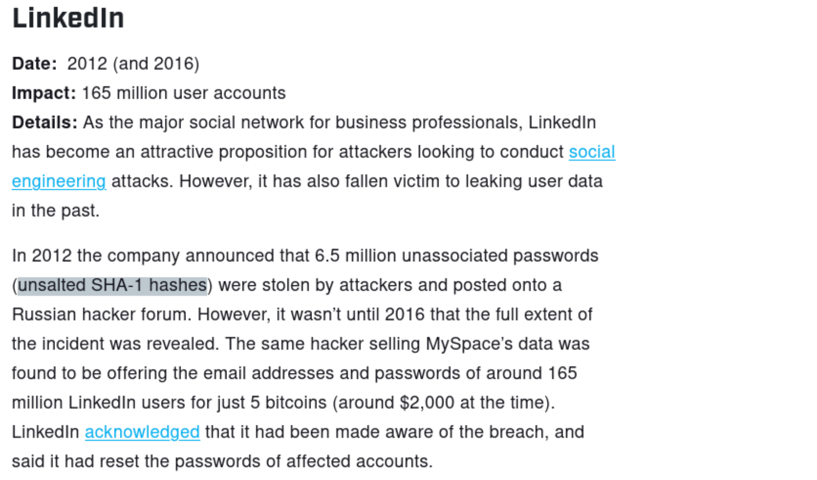
512 25

Adobe's HQ. The company leaked over 100m users' details. Photograph: PAUL SAKUMA/ASSOCIATED PRESS

Nearly 150 million people have been affected by a loss of customer data by Adobe, over 20 times more than the company admitted in its initial statement last week.

- Encrypted (ECB)

LinkedIn



LinkedIn

Date: 2012 (and 2016)

Impact: 165 million user accounts

Details: As the major social network for business professionals, LinkedIn has become an attractive proposition for attackers looking to conduct social engineering attacks. However, it has also fallen victim to leaking user data in the past.

In 2012 the company announced that 6.5 million unassociated passwords (unsalted SHA-1 hashes) were stolen by attackers and posted onto a Russian hacker forum. However, it wasn't until 2016 that the full extent of the incident was revealed. The same hacker selling MySpace's data was found to be offering the email addresses and passwords of around 165 million LinkedIn users for just 5 bitcoins (around $2,000 at the time). LinkedIn acknowledged that it had been made aware of the breach, and said it had reset the passwords of affected accounts.

Password Collection

Password Collection

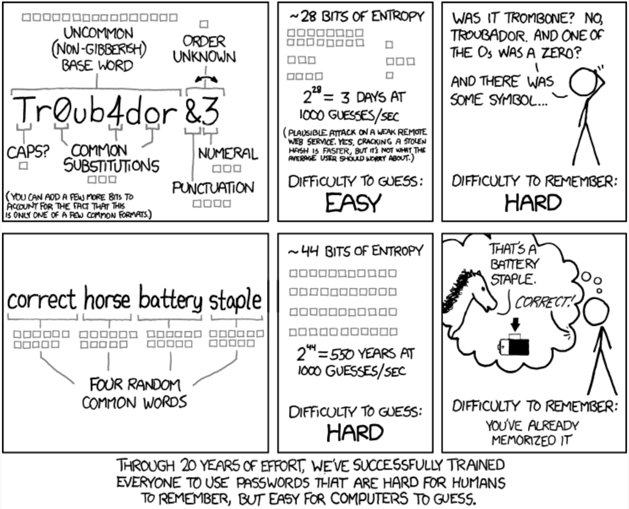
- Educative Purpose Only
- **Using thoses passwords is illegal**

Recommendations

NIST

Recommentations for password and authentication:

- Nist



UNCOMMON (NON-GIBBERISH) BASE WORD
ORDER UNKNOWN
Tr0ub4dor&3
CAPS?
COMMON SUBSTITUTIONS
NUMERAL
PUNCTUATION
(YOU CAN ADD A FEW MORE BITS TO ACCOUNT FOR THE FACT THAT THIS IS ONLY ONE OF A FEW COMMON FORMATS.)
~28 BITS OF ENTROPY
2^28 = 3 DAYS AT 1000 GUESSES/SEC
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)
DIFFICULTY TO GUESS: EASY
WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE Os WAS A ZERO?
AND THERE WAS SOME SYMBOL...
DIFFICULTY TO REMEMBER: HARD
correct horse battery staple
FOUR RANDOM COMMON WORDS
~44 BITS OF ENTROPY
2^44 = 550 YEARS AT 1000 GUESSES/SEC
DIFFICULTY TO GUESS: HARD
THAT'S A BATTERY STAPLE.
CORRECT!
DIFFICULTY TO REMEMBER: YOU'VE ALREADY MEMORIZED IT
THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

Other considerations

- Allow special characters and spaces (Emoji, Unicode, . . .)
- No Password hints
- Screen new passwords against:
  - commonly used or compromised passwords
  - Dictionary words
  - Sequential or repetitive (“aaaa”, “4567”)
  - Context-specific words (Service name, username, . . .)
- Min 8 chars, Max 64 chars (around 340 bits of entropy)
- Limit rate submission
- Use approved encryption and an authenticated protected channel in order to provide resistance to eavesdropping and MitM attacks.
- Remove periodic password change requirements (but force a change if evidence of compromise)

Other considerations 2

- Should be permit to past (password manager)
- Should offer an option to display the secret
- Should provide meter
- Should not impose composition limite (chars maj min num special,)

Do NOT use

- ClearText Password
- Hash (SHA-2, SHA-3, . . .)
- Reversible Encryption
- Salt + Hash only
- Home made function *(md5(sha1(mdp)))*

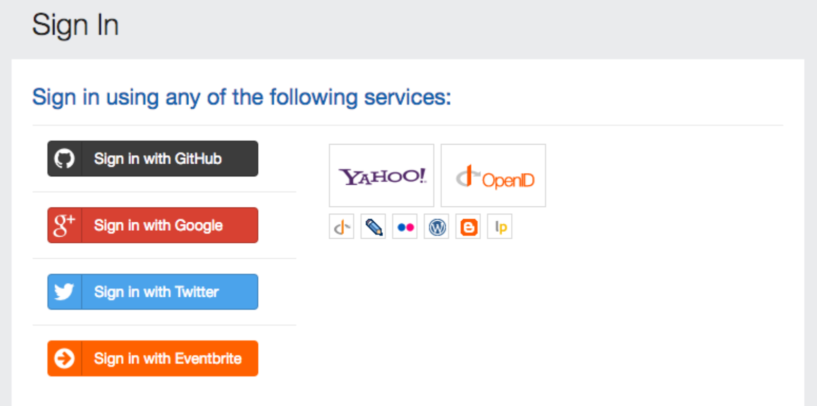
Use a Key derivation function

- PBKDF
- Argon2
- bcrypt
- scrypt
- Lyra2
- Balloon

NIST recommendations for Key derivation function

- Verifiers SHALL store memorized secrets in a form that is resistant to **offline attacks**. Memorized secrets SHALL be **salted** and **hashed** using a suitable one-way **key derivation function**. Key derivation functions take a **password**, a **salt**, and a **cost factor** as inputs then generate a password hash.
- Use at least 10,000 iterations
- The salt SHALL be at least **32 bits** in length and be chosen arbitrarily so as to minimize salt value collisions among stored hashes.
- verifiers SHOULD perform an additional iteration of a key derivation function using a salt value that is secret and known only to the verifier. (also known as pepper)
- A **memory-hard** function SHOULD be used because it increases the cost of an attack.

PasswordLess ?



Sign In

Sign in using any of the following services:

- OpenID Connect, SAML, OAuth2, PKCE

Multi Factor Authentication

- Something you know (password)
- Something you have (device, paper, token)
- Something you are (biometrics)

Authenticator Assurance Level (by NIST)

- Authenticator Assurance Levels