Managing password security in 2015

Rolfe Bozier

29-Apr-2015

This work is licensed under a Creative Commons Attribution 4.0 International License.

Agenda

- A bit of background
- How to manage password security – badly
- How to manage password security – properly
- Managing your passwords

This work is licensed under a Creative Commons Attribution 4.0 International License.

Background

- A password is secret shared between you and a second party
- What do passwords achieve?
  - "prove" who you are to some system
  - but really just proves you know a secret that was given to someone
- Why we still use passwords
  - They are cheap
  - They are portable
  - You can easily change them
- What about biometrics?
  - They are expensive
  - You cannot change them
  - What if you are blind, an amputee or physically disabled?
  - Compromise may entail physical coercion

This work is licensed under a Creative Commons Attribution 4.0 International License.

How to manage password security - *badly*

This work is licensed under a Creative Commons Attribution 4.0 International License.

How to manage password security - badly

- Store the wrong thing
  - storing passwords as plaintext
    - only system security can protect them
  - storing using symmetric encryption
    - the decryption password needs to be on the system
    - only system security can protect them
- If a system ever offers to send you your password, you have a big problem
- A system should only ever store a hashed password

This work is licensed under a Creative Commons Attribution 4.0 International License.

How to manage password security - badly

- Not protecting during transmission
  - Sending the password using HTTP not HTTPS
  - if there is no padlock on the browser, it can be visible!
- Unexpected leaks
  - Once upon a time, Unix systems used to log login failures:

```
[...]
Jan 19 12:13:14 server1 login: login failed for user rolfe on tty3
[...]
```

  - What could possibly go wrong?

This work is licensed under a Creative Commons Attribution 4.0 International License.

How to manage password security - badly

- Poor encryption algorithms
  - Rolling your own or “improving” on another one
  - DES password encryption was limited to 56-bit keys
    - Truncate passwords after 8 characters
  - Algorithms that are optimised for speed
  - Algorithms that are easy to implement in parallel
  - Algorithms with known weaknesses

This work is licensed under a Creative Commons Attribution 4.0 International License.

How to manage password security - badly

- Protecting encrypted passwords
  - Before the 1990s, encrypted passwords on Unix were not hidden
  - Rainbow tables contain lists of plaintext + encrypted password pairs
  - Even today, password security requires OS/system integrity
    - Every month it seems another site is hacked
- Front-end vulnerabilities
  - Security needs to be maintained from the keyboard all the way through to the authentication code at the remote end
  - ATM skimming / shoulder surfing
  - Malware key-loggers
  - Storing plaintext passwords in files
  - Social engineering

This work is licensed under a Creative Commons Attribution 4.0 International License.

How to manage password security - *properly*

This work is licensed under a Creative Commons Attribution 4.0 International License.

How to manage password security - properly

- Implementing password encryption
  - Research the options in your deployment environment
  - *Know* which is the right way in your environment
    - Which is best: MD5(), SHA1(), RC4(), crypt(), password_hash() ??
    - if you don't know, don't guess!
  - Salt your passwords
  - Rule of thumb: when in doubt, use bcrypt
    - ensures the calculation is slow
    - does not have a parallel implementation
  - Allow for algorithm upgradability
    - Moore's Law: what's OK today probably won't be in 18 months
- Password verification
  - Overwrite password information in memory ASAP
  - What’s wrong with using strcmp() ?
  - Use rate limiting

This work is licensed under a Creative Commons Attribution 4.0 International License.

How to manage your password

This work is licensed under a Creative Commons Attribution 4.0 International License.

How to manage your own passwords

- Bad news!
  - Assume the bad guy has a large list of encrypted passwords (this happens)
  - Also assume the encryption algorithm is known
- How to crack a password:
  - First, try common passwords
  - Second, apply rule-based variations
  - Exhaustive checking targeting key subspaces:
    - shorter passwords
    - lower-case only
    - upper-case only
    - letters only
    - letters + numbers
  - Combine the two previous steps
  - Finally, try brute force on the rest of the password space
  - Once they've cracked a password, it is added to their list

This work is licensed under a Creative Commons Attribution 4.0 International License.

How to manage your own passwords

- Crackers are *very* good at what they do
  - They spend more time thinking about breaking your password than you spend creating it
  - They have access to fast/parallel hardware
  - They feed their successes back into improving their heuristics
- You might think that the following passwords are secure:
  - `:LOL1313le`
  - `Coneyisland9/`
  - `momof3g8kids`
  - `1368555av`
  - `n3xtb1gth1ng`
  - `qeadzcwrsfxv1331`
  - `m27bufford`
  - `J21.redskin`
  - `Garrett1993*`

This work is licensed under a Creative Commons Attribution 4.0 International License.

How to manage your own passwords

- How to choose a good password?
  - DON'T: simple passwords (see above)
  - DON'T: mnemonic rule-based passwords (you know... i→1, o→0 etc.)
  - DO: choose at least 12 characters
  - DO: choose appropriate security levels
    - your Slashdot password is used often and isn't critical, so it can be easier to remember
    - your PayPal password better be harder to guess!
- You want to force a cracker to fall back to brute force:
  - Include ALL of: upper/lower case, digits, punctuation
- There are sites they will auto-generate passwords that:
  - are easier to remember (e.g. "pronounceable")
  - use a good range of characters

This work is licensed under a Creative Commons Attribution 4.0 International License.

How to manage your own passwords

- Which password is more secure?
  - `D0g.....................`
  - `PrXyc.N(n4k77#L!eVdAfp9`

This work is licensed under a Creative Commons Attribution 4.0 International License.

How to manage your own passwords

- Which password is more secure?
  - `D0g.....................`
  - `PrXyc.N(n4k77#L!eVdAfp9`
- Each extra character introduces ~100 times more effort
- Corollary: password strength measures are generally junk

This work is licensed under a Creative Commons Attribution 4.0 International License.

How to manage your own passwords

- Dealing with password re-use
  - how many passwords have you set up?
  - how many passwords are the same?
  - what would happen if one was cracked?
    - now they are all vulnerable
- All your accounts are vulnerable to the weakest implementation

This work is licensed under a Creative Commons Attribution 4.0 International License.

How to manage your own passwords

- It’s all too hard! - not really
- If two-factor authentication is available, choose it!
- For infrequently-used passwords, specify a long random password
  - use password-reset to create a new password
  - your email address had better be secure
- Consider a password safe
  - Protect all your passwords with 1 strong password
  - **DO** research the software first, some are pretty bad

This work is licensed under a Creative Commons Attribution 4.0 International License.

How I manage passwords

This work is licensed under a Creative Commons Attribution 4.0 International License.

How I manage passwords

- I have around 100 password-protected accounts
- I have 4 main passwords:
  - 2 for low-security sites (forums, registration-required, …)
  - 1 for medium security sites (e-commerce)
  - 1 for high security network services (domains, email etc)
- Plus some other random ones which are infrequent but important (e.g. MySQL password on hosted website)
- Plus a few critical ones (ATM PIN, Internet banking)
- I can’t remember all these (especially in conjunction with different usernames)

This work is licensed under a Creative Commons Attribution 4.0 International License.

How I manage passwords

- I use the KeePass password safe software
- The database is protected by a master password and a key certificate
- I have KeePass installed on my Linux desktop and Android phone
- The database is replicated via a private Dropbox folder
- Database entries can include other text, which is handy
- Really important passwords (ATM PINs, Internet Banking) aren't included.

This work is licensed under a Creative Commons Attribution 4.0 International License.

Summary

- Advances in hardware and cracking techniques mean that passwords are more vulnerable than ever before
  - You probably can't choose an easy-to-remember password that can't be cracked
- System intrusion techniques are becoming more sophisticated
  - You can't rely on the remote system not to leak information
- Re-used passwords are vulnerable to the worst security implementation

This work is licensed under a Creative Commons Attribution 4.0 International License.

Summary

- To ensure your password remains uncrackable:
  - It **must** include upper/lower case, digits and punctuation
  - It **must** be at least 12 characters long

This work is licensed under a Creative Commons Attribution 4.0 International License.