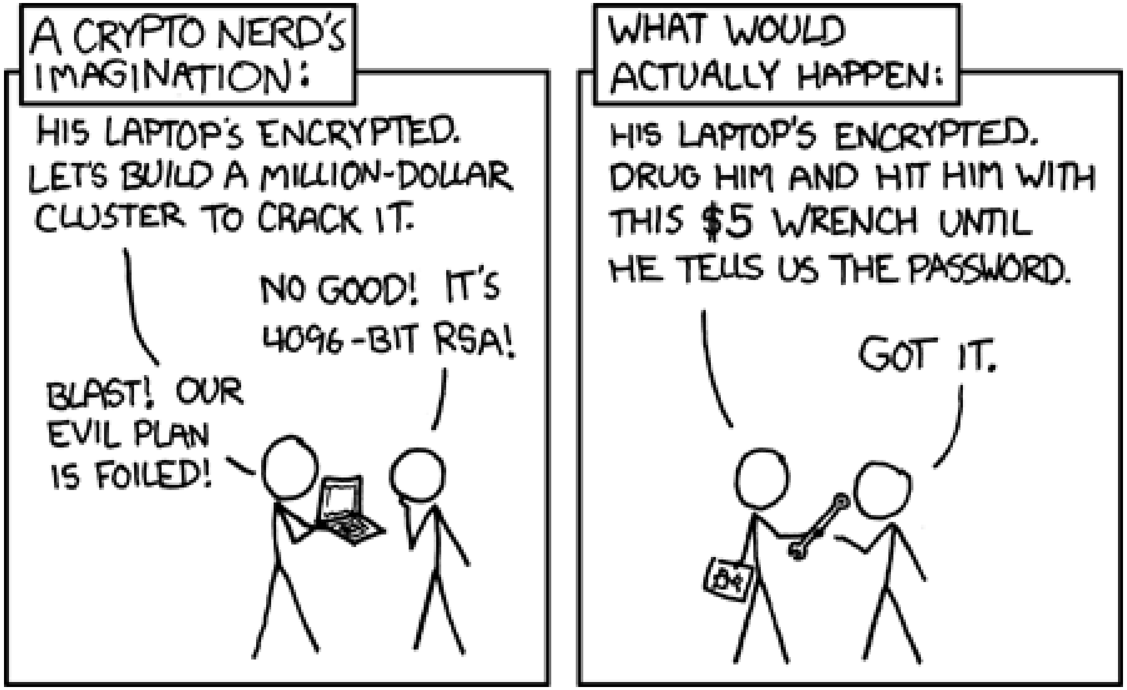

A CRYPTO NERD'S IMAGINATION:
HIS LAPTOP'S ENCRYPTED. LET'S BUILD A MILLION-DOLLAR CLUSTER TO CRACK IT.
NO GOOD! IT'S 4096-BIT RSA!
BLAST! OUR EVIL PLAN IS FOILED!
WHAT WOULD ACTUALLY HAPPEN:
HIS LAPTOP'S ENCRYPTED. DRUG HIM AND HIT HIM WITH THIS $5 WRENCH UNTIL HE TELLS US THE PASSWORD.
GOT IT.

Source: https://xkcd.com/538/

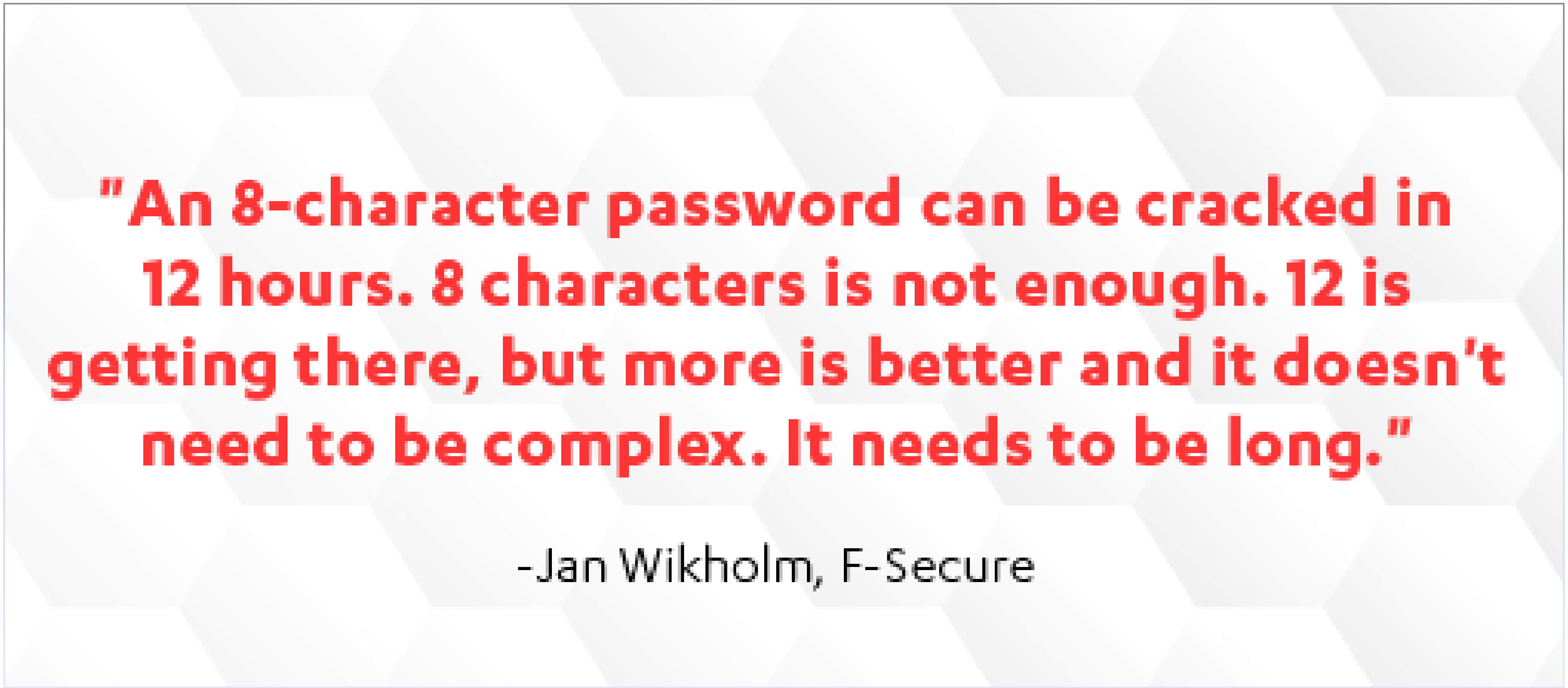
Is That Password Long Enough?

Or why we should stop focusing on password length

REFRESHING MEMORIES

Tilman Runge

35c3 Day 2, 20:30 @ M1



"An 8-character password can be cracked in 12 hours. 8 characters is not enough. 12 is getting there, but more is better and it doesn't need to be complex. It needs to be long."

-Jan Wikholm, F-Secure

Long enough to achieve what …?

- What means secure?
- Which threat (=who) are we defending against?
- How does a longer password help you with … what?

Focus on

- Online services (social networks, online banking)
- Password to de/encrypt data

What do we need passwords for?

Protect our information and processes

- **Confidentiality**: Access to sensitive data in our email account or decrypt an encrypted email
- **Integrity**: online banking
- **Authenticity**: e-government, signed emails

Different technical implementations:

- Passwords for **authentication**: user name + password („account“)
- Passwords to **encrypt/decrypt** data: Veracrypt, email, laptop, etc.

Common threats to passwords

Include:

- Phishing („Please confirm your Paypal credentials…“)
- Shoulder surfing
- Keylogging / access to unlocked user device
- Giving up passwords at the border
- Guessing (brute forcing the password)

<u>Main reason </u>for long passwords: make it harder to guess/brute force

Passwords: Technical implementations

Encryption

Veracrypt, Hard-Disk-Encryption, PGP, WPA2, passwordmanager

- Only one field: password
- Password is used as an **encryption key** for encrypting/decrypting data
- When forgotten, access to data is lost (forgotten password=forgotten decryption key)

Authentication

Online services

- Input fields for username *and* password
- Password is used as **proof** – only the user knows it
- The password given by the user is compared to the password in the user database
- Password can be changed/resetted when forgotten
- Intrusion detection systems to avoid brute forcing may be used

Passwords used for encryption

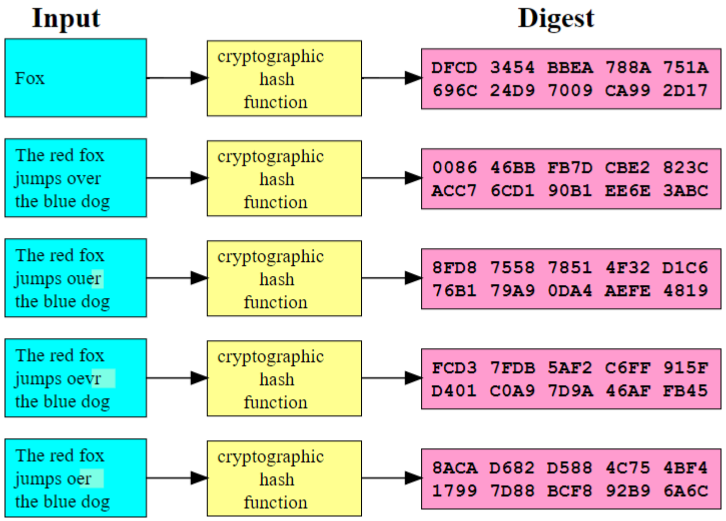
- Password is used as an encryption key
- **P**assword **b**ased **k**ey **d**erivation **f**unction (PBKDF) first transforms the password into key

  Why? To ensure some properties needed for cryptographic keys such as:

  - Length (e.g. 128bit)
  - Entropy
  - And: time to compute

… lets look at PBKDF more in depth: Hashing

Hashing


Input
Digest
Fox
cryptographic hash function
DFCD 3454 BBEA 788A 751A
696C 24D9 7009 CA99 2D17
The red fox jumps over the blue dog
cryptographic hash function
0086 46BB FB7D CBE2 823C
ACC7 6CD1 90B1 EE6E 3ABC
The red fox jumps ouer the blue dog
cryptographic hash function
8FD8 7558 7851 4F32 D1C6
76B1 79A9 0DA4 AEFE 4819
The red fox jumps oevr the blue dog
cryptographic hash function
FCD3 7FDB 5AF2 C6FF 915F
D401 C0A9 7D9A 46AF FB45
The red fox jumps oer the blue dog
cryptographic hash function
8ACA D682 D588 4C75 4BF4
1799 7D88 BCF8 92B9 6A6C

Source: https://en.wikipedia.org/wiki/File:Cryptographic_Hash_Function.svg

Characteristics:

- Any type of input (password, text, file, image, …) will result in the same *type* of output
- One-way function: can only be computed from input to output
- Output does not reveal anything about input

Breaking Passwords: The Math

Brute force (trying all possible combinations) is possible, but takes a while

- But how long? For each password we need to
  1. Derive cryptographic key from password (using key derivation function)
  2. Try if cryptographic key unlocks the encrypted data
  3. Unsuccessful? Use next password and start at 1.

- So how long until we have found the password?
  - Depends on number of tries necessary to find the right password (→length of password important)

    aaaa, aaab, aaac, aaad, …. bbba, bbbb, bbbc, …. xxxx
  - Depends on speed of the key derivation function

PBKDF: Repeated Hashing

We want purposefully increase time needed to compute encryption key

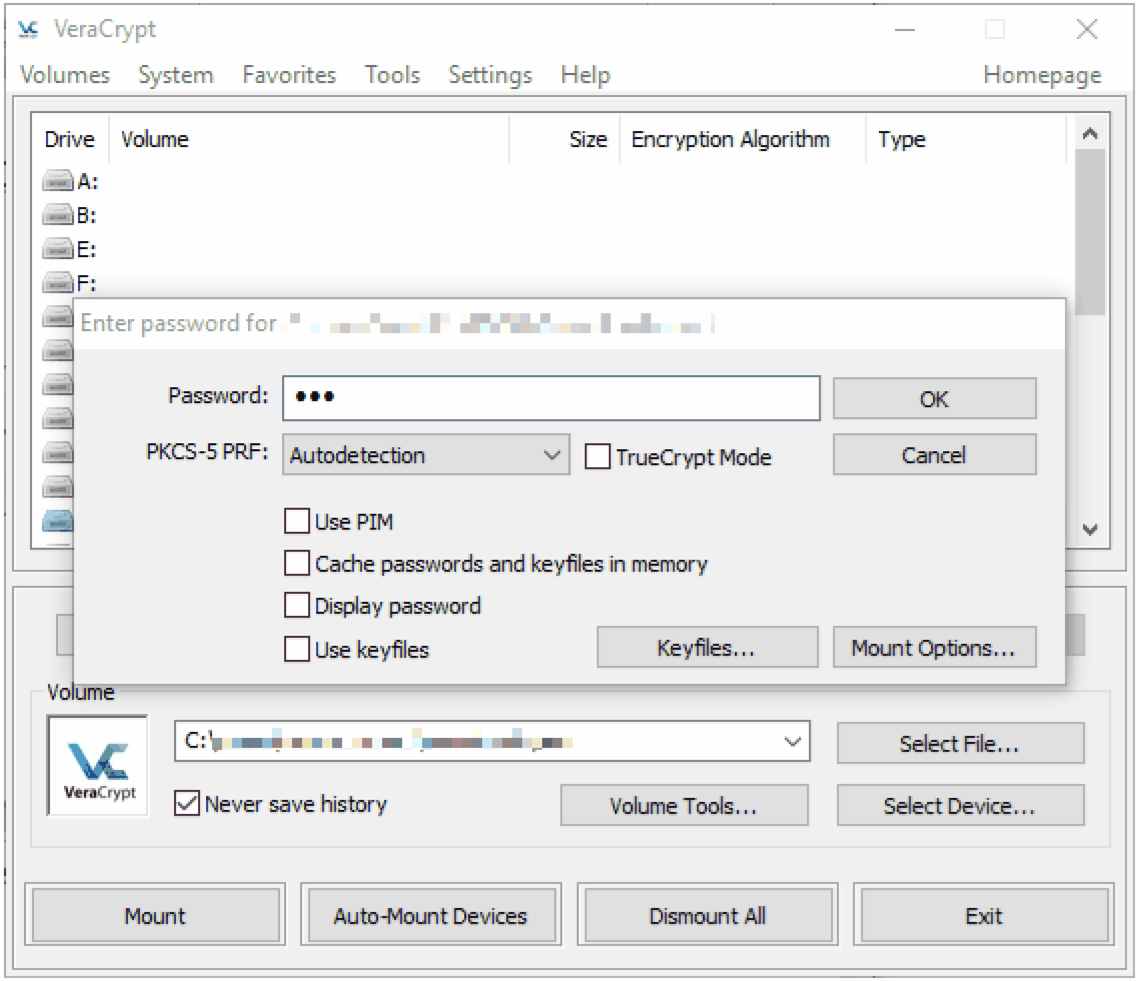
- Use the PBKDF on its output iteratively:

1. PKDF(„secret“) = 29mca48a
2. PKDF(„29mca48a“) = c4m03cdk
3. PKDF(„c4m03cdk“) = iod492dk

…

100.000. PKDF („aplckm340“) = cw3ld02c ← the encryption key

Key derivation functions in real life



Enter password for [redacted]

Password: •••

PKCS-5 PRF: Autodetection

□ TrueCrypt Mode

□ Use PIM

□ Cache passwords and keyfiles in memory

□ Display password

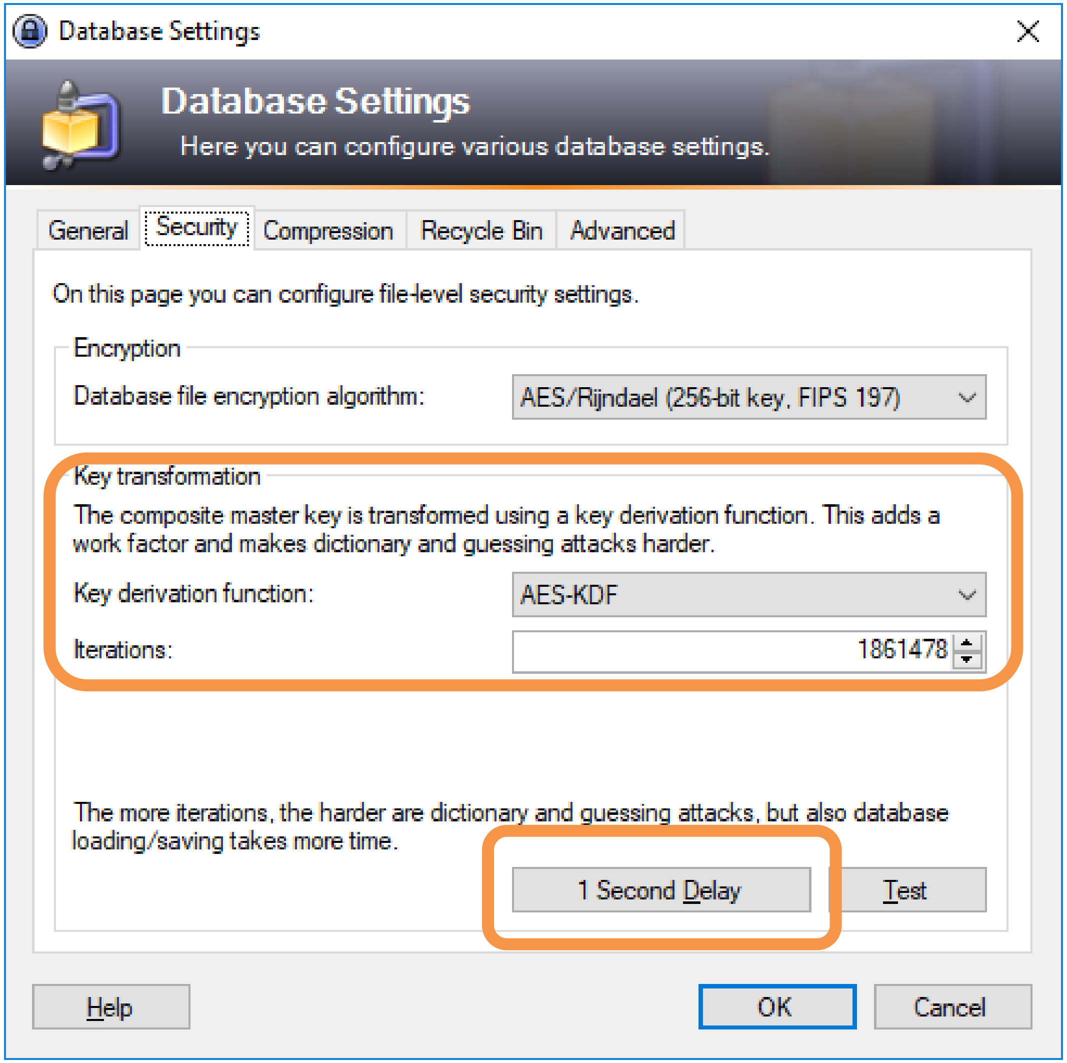
□ Use keyfiles

Volume

C:\[redacted]

☑ Never save history

Adjust key derivation function delay



Database Settings

Here you can configure various database settings.

On this page you can configure file-level security settings.

Encryption

Database file encryption algorithm: AES/Rijndael (256-bit key, FIPS 197)

Key transformation

The composite master key is transformed using a key derivation function. This adds a work factor and makes dictionary and guessing attacks harder.

Key derivation function: AES-KDF

Iterations: 1861478

The more iterations, the harder are dictionary and guessing attacks, but also database loading/saving takes more time.

Breaking Passwords: The Math

How long until we have found the password?

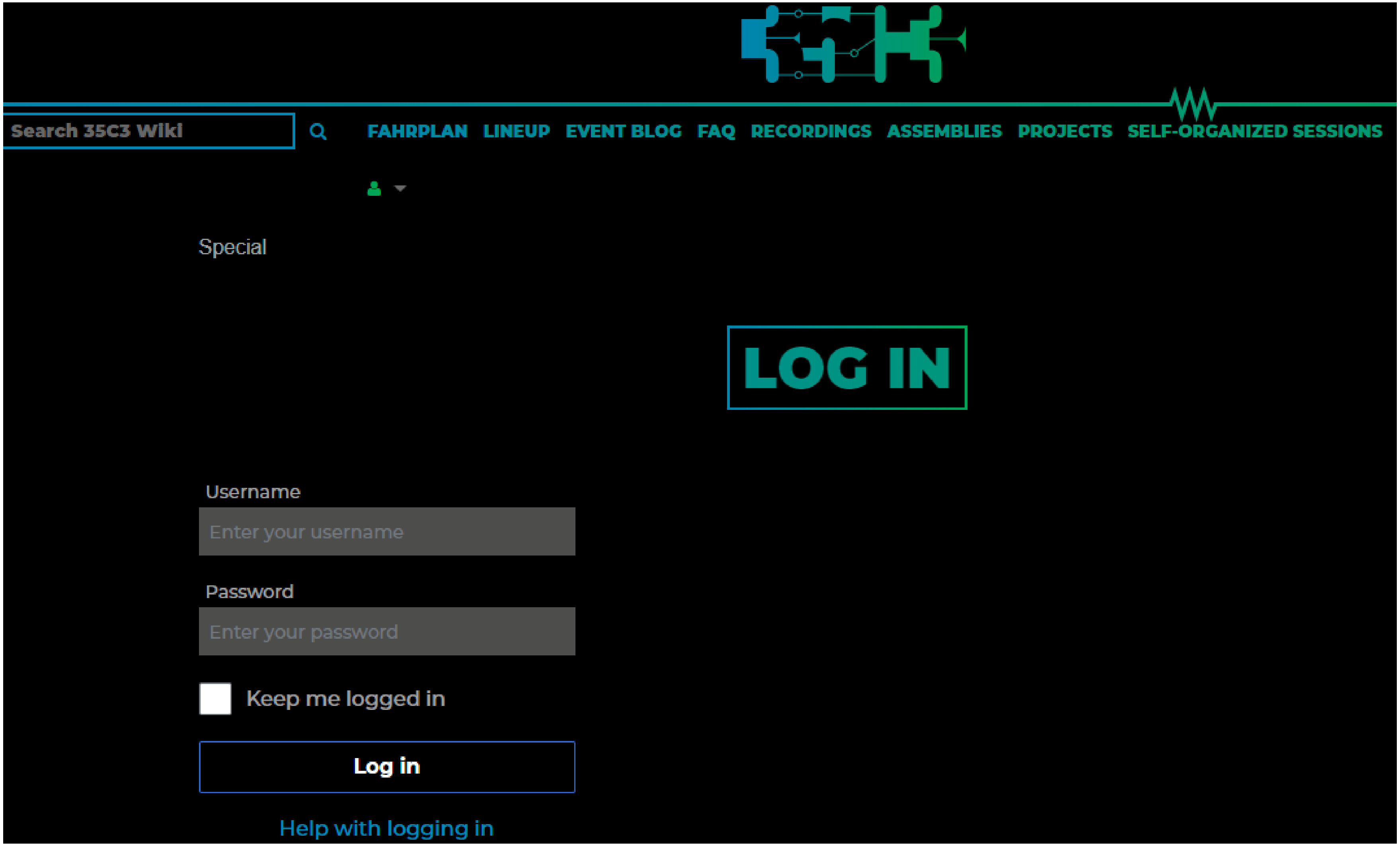
- Depends on speed of the key derivation function
- Depends on number of tries necessary to find the right password

Total time = number of tries * time necessary/try

Example: 8 characters (a-z, A-Z, 0-9, special ch.) with ~262 billion combinations

Password guesses per second	**Time needed in years**
100.000	0,1
1000	8,3
1	8.312,5
0,2	41.562,7

Passwords: authentication



Search 35C3 Wiki
FAHRPLAN LINEUP EVENT BLOG FAQ RECORDINGS ASSEMBLIES PROJECTS SELF-ORGANIZED SESSIONS

Special

LOG IN

Username

Enter your username

Password

Enter your password

□ Keep me logged in

Log in

Help with logging in

Passwords for authentication

- Online service needs to identify its users

  → database with usernames and passwords

- One-way encryption (hashing) is used to save passwords in the database

  → bcrypt(„secret_passwort") = 29mca48a

- How can the user authenticate if the online service only knows the hash?

  bcrypt(user_password)=saved_password_in_database

Passwords for authentication

So passwords are hashed and we are safe, right?

- Are passwords really hashed?
- Password hashed in the database but in clear text in a logfile
- Is the hash algorithm safe against guessing/brute-forcing?

→ We simply cannot know

→ Assume that your passwords have been leaked

Have you ever used …

KICKSTARTER

YAHOO!

imgur

YOUPORN

tumblr

Dropbox

myspace

Adobe

last.fm

EPIC GAMES

LinkedIn®

Source:
https://haveibeenpwned.com/PwnedWebsites

Threat modell: Untargeted attack

Seqences of events

1. The user signs up at $onlineservice
2. Someone breaks into $onlineservice, steals user database containing user names and (hashed) passwords
   - In a perfect world, hashed & salted passwords would be unusuable
   - In our world: consider your passwords leaked in clear text
3. Someone (else) uses leaked credentials to break into other web services: Facebook, Google, Paypal, …
4. ???
5. Profit!!

Threat modell: Untargeted attack

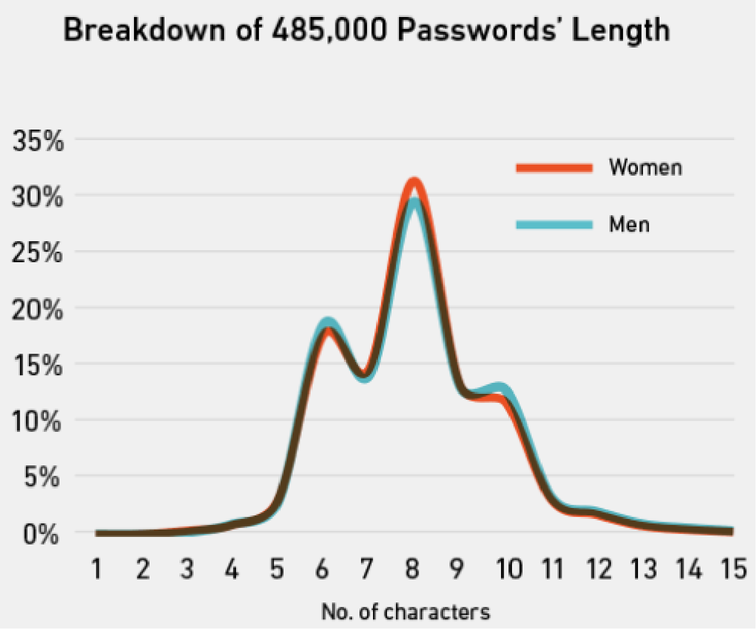
Some characteristics

- Individuals are not the target
- Motivation: $$$, fame
- Only online accounts targeted
- Division of labour:
  1. Someone steals user database, later sells it
  2. Someone else cracks hashed password, sells it
  3. Again someone else
     - Finds and sells valuable accounts (Paypal, Steam, Fortnite, World of Warcraft…)
     - Uses the credentials to get access theemailaccount to send malware
     - Uses the credentials to find out more about you to start a spear phishing campaign
- It is inefficient to put much effort into cracking *every* password
- The most prevalent type of threat: makes up ~90% of retail logins attempts

Threat modell: Untargeted attack

How to defend:

- Do not reuse passwords
- Use passwords that are long enough so that it becomes inefficient to crack them
- Remember: Some online services save passwords in plain text



https://wpengine.com/unmasked/

Threat modell: targeted attack

Characteristics:

- Someone is interested in you & willing to spend time
- Motivation: can be anything
- Target: Online accounts *and* personal devices:
  - May try to decrypt your encrypted data on your devices
  - Gain access to online accounts

How to defend:

- Avoid phishing (NoPhish by Secuso)
- Do not install software of unknown origin
- Not by using longer passwords but different passwords
- Your email account is your most valuable account

Password managers

Provide you with unknown comfort:

- Database: You never have to remember a password again*
- PW-Generation: You never have to think of a password again
- Auto Input: You never have to type a password again
- Phishing: You are unlikely to be phished (again)

Some issues to think about:

Needs to be always accessable:

- How to keep them in sync with all your devices? Cloud sync?
- Privacy versus useability
- Not all Password managers

Need for backup:

- Several places
- Forgot your password?

*Except the password to open the password manager

Admins, Developers: Help your users

How can you as a developer or sysadmin help:

- Implement systems that don not rely on passwords or stop password proliferation:
  - Single Sign On
  - Active Directory as identity provider
- Do not rely solely on one factor for successful authentication
  - 2-Factor-Authentication
  - Password reset needs to be hard
- Help users stop using bad passwords
  - Give them Password managers
  - Stop pestering them with password rules (complexity, change frequency)
  - Upon password change: do not accept passwords that have already been leaked elsewhere (→check against HaveIBeenPwned.com password list)

Conclusion

- Attack type
  - Targeted: It is about you
  - Untargeted: It is about large numbers of credentials
  - Easy to randomly target masses, hard to hack a specific account of a specific person
- Two types of uses for passwords
  - Encryption
  - Authentication
- Password length doesn‘t matter once your password was leaked, which is likely.
  - If you can, do use long passwords
  - More important: use different passwords
  - Use a password safe

Q&A