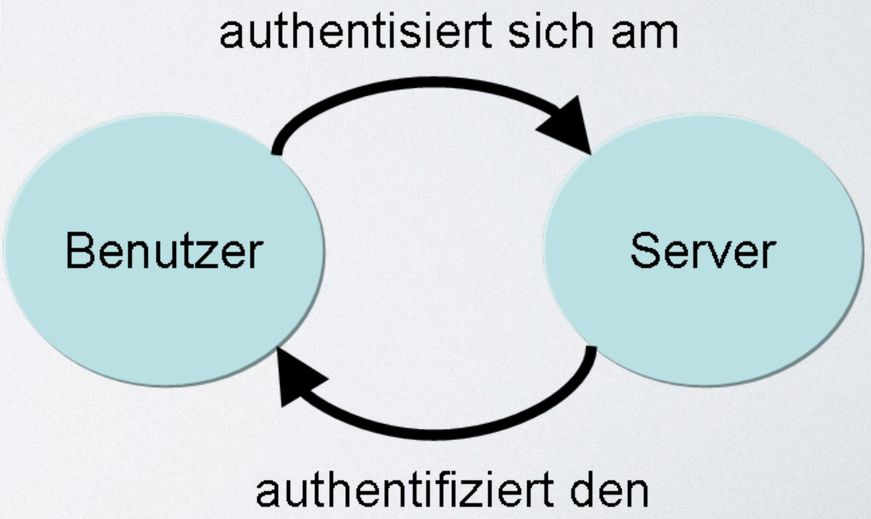
PASSWÖRTER 101

PeP et al. Sommerkademie 2015

Tschagguns
Christophe Cauet

GESCHICHTE

- Passwort, Passphrase, Geheimzahl, Kennwort, Schlüsselwort, Codewort, Losung, Parole
- Freund/Feind Kennung, Authentifizierung
  - "*Bezeugung der Echtheit*"

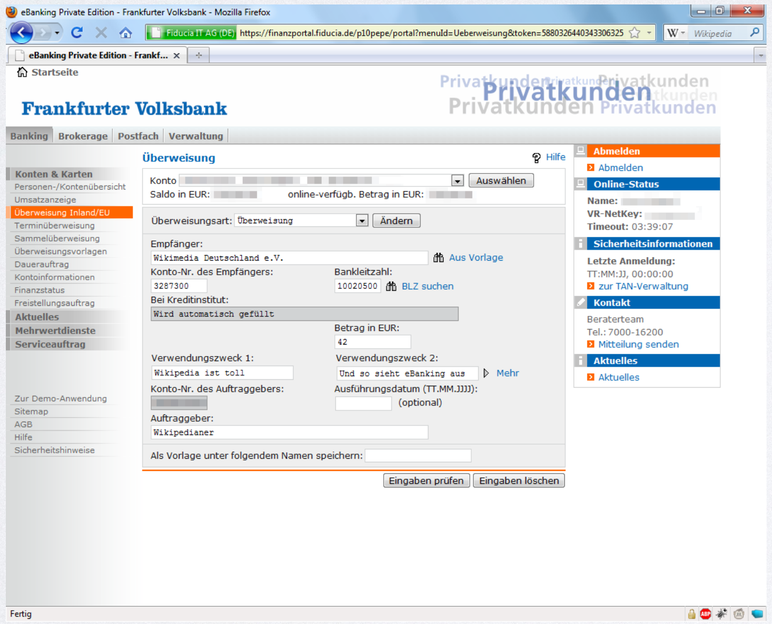


Christoph Probst, https://de.wikipedia.org/wiki/Datei:Authentisieren-authentifizieren.png

BEISPIELE

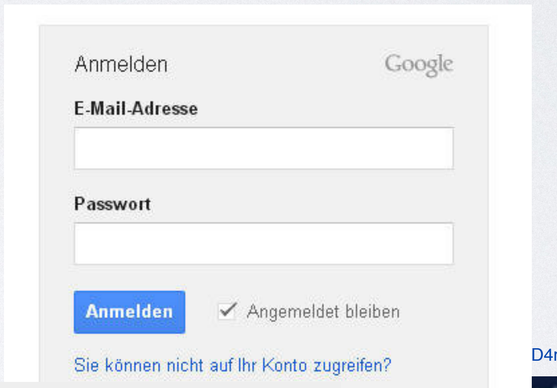
- Geldautomat, Online- und Tele-Banking, Türschloss, allg. Zutrittskontrolle, Militär, allg. Online-Dienste, …

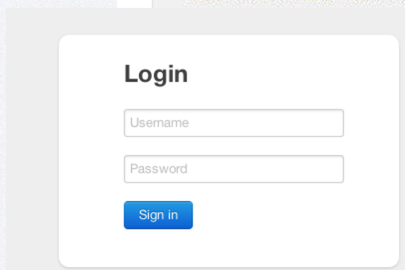
Neitram, https://de.wikipedia.org/wiki/Datei:ATM_pinpad_in_german.jpg

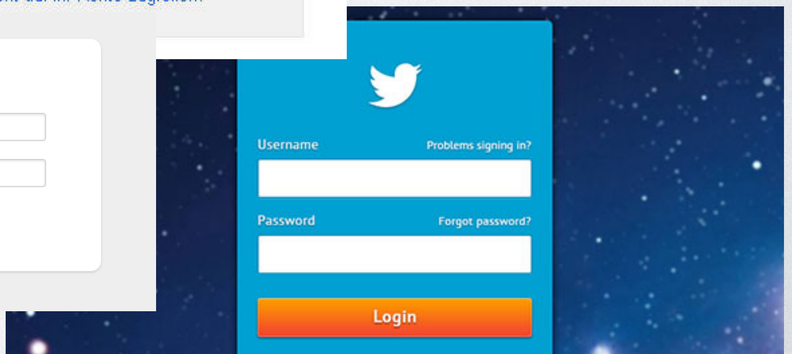


Benji, https://commons.wikimedia.org/wiki/File:Screenshot_eBanking_bei_der_Frankfurter_Volksbank.png

D4m1en, https://commons.wikimedia.org/wiki/File:Digicode.JPG







FLASH AND THUNDER

- Speicherung im Klartext
  - einfach & unsicher
- Speicherung des "Passwort-Hash"

„Eine Hashfunktion ist eine Funktion, die eine Zeichenfolge beliebiger Länge auf eine Zeichenfolge mit fester Länge abbildet. Mathematisch ist diese Funktion nicht injektiv (linkseindeutig) und nicht notwendigerweise surjektiv (rechtstotal).“

Kryptologische Hashfunktion, Wikipedia

MD5

- "*Message-Digest Algorithm 5*", Ronald L. Rivest, MIT (1991)
- 128-bit Hash
- schnell, unsicher (*common-prefix collisions*), häufig verwendet
  - Beispiele

`SoAk15`	→ MD5 →	`85ce6b0153934771ce8817f7b5d51d26`
`SoAk14`	→ MD5 →	`d4875b3c5fb2780f18efd7c126c02a2c`
`password`	→ MD5 →	`5f4dcc3b5aa765d61d8327deb882cf99`

VORGEHEN

1. Passwort-Hash erzeugen

2. Passwort-Hash abgleichen

VORTEILE

- Passwort wird nicht mehr im Klartext gespeichert
- Hash kann nicht zu Passwort zurückgerechnet werden

NACHTEILE

- Berechnung des Hash u.U. langwierig

TRADEOFF

- langwieriger Hash-Algorithmus sicherer
- erhöht Gefahr von DDoS-Angriffen

MAC MINI 2014

- Intel x86_64 i5-4260U CPU @ 1.40GHz w/ 4 threads

Hash	words/sec
MD5	30M
SHA1	15.5M
SHA256	8.4M
SHA-3 (Keccak)	2.6M
bcrypt	1.9k
scrypt	75
OSX 10.8+	25
osCommerce	22.5M

FAZIT

- Speichern von Hashes vs. Klartext
  - Passwörter können nicht einfach ausgelesen werden
  - Hash-Funktion sicher → Passwort geschützt

ANGRIFFS-VEKTOREN

- Abfangen von Passwörtern (**gerichtet**)
  - Man-in-the-middle, Keylogger, Phishing
- Diebstahl von Nutzer-Credentials (**ungerichtet**)
  - Zugang zu Servern

DIEBSTAHL VON DATEN

- Missbrauch der Nutzer-Daten
  - Email-Anbieter, Online-Banking, Kreditkarten-Diebstahl, Social-Engineering, Online-Versand, Identitätsdiebstahl

ROCKYOU.COM (2009)

- SQL injection vulnerability
- 32 Millionen Passwörter aus Datenbank ausgelesen
- Im Klartext!
- 14 Millionen einzigartige, **echte** Passwörter
- Bonus: Email-Adressen und Account-Infos von Drittenanbietern (my-space, facebook, …)

```
Data UserAccount [32603388]
===============
1|jennaplanerunner@hotmail.com|mek*****|myspace|0|bebo.com
2|phdlance@gmail.com|mek*****|myspace|1|
3|jennaplanerunner@gmail.com|mek*****|myspace|0|
5|teamsmackage@gmail.com|pro*****|myspace|1|
6|ayul@email.com|kha*****|myspace|1|tagged.com
7|guera_n_negro@yahoo.com|emi*****|myspace|0|
8|beyootifulgirl@aol.com|hol*****|myspace|1|
9|keh2oo8@yahoo.com|cai*****|myspace|1|
10|mawabiru@yahoo.com|pur*****|myspace|1|
11|jodygold@gmail.com|att*****|myspace|1|
12|aryan_dedboy@yahoo.com|iri*****|myspace|0|
13|moe_joe_25@yahoo.com|725*****|myspace|1|
14|xxxnothingbutme@aol.com|1th*****|myspace|0|
15|meandcj069@yahoo.com|too*****|myspace|0|
16|stacey_chim@hotmail.com|cxn*****|myspace|1|
17|barne1en@cmich.edu|ilo*****|myspace|1|
18|reo154@hotmail.com|ecu*****|myspace|1|
19|natapappaslie@yahoo.com|tor*****|myspace|0|
20|ypiogirl@aol.com|tob*****|myspace|1|
21|brittanyleigh864@hotmail.com|bet*****|myspace|1|myspace.com
22|topenga68@aol.com|che*****|myspace|0|
23|marie603412@yahoo.com|cat*****|myspace|0|
24|mellowchick41@aol.com|chu*****|myspace|0|
25|baiko0o@aol.com|may*****|myspace|0|
26|indahamzah84@hotpop.com|lov*****|myspace|0|
...
```

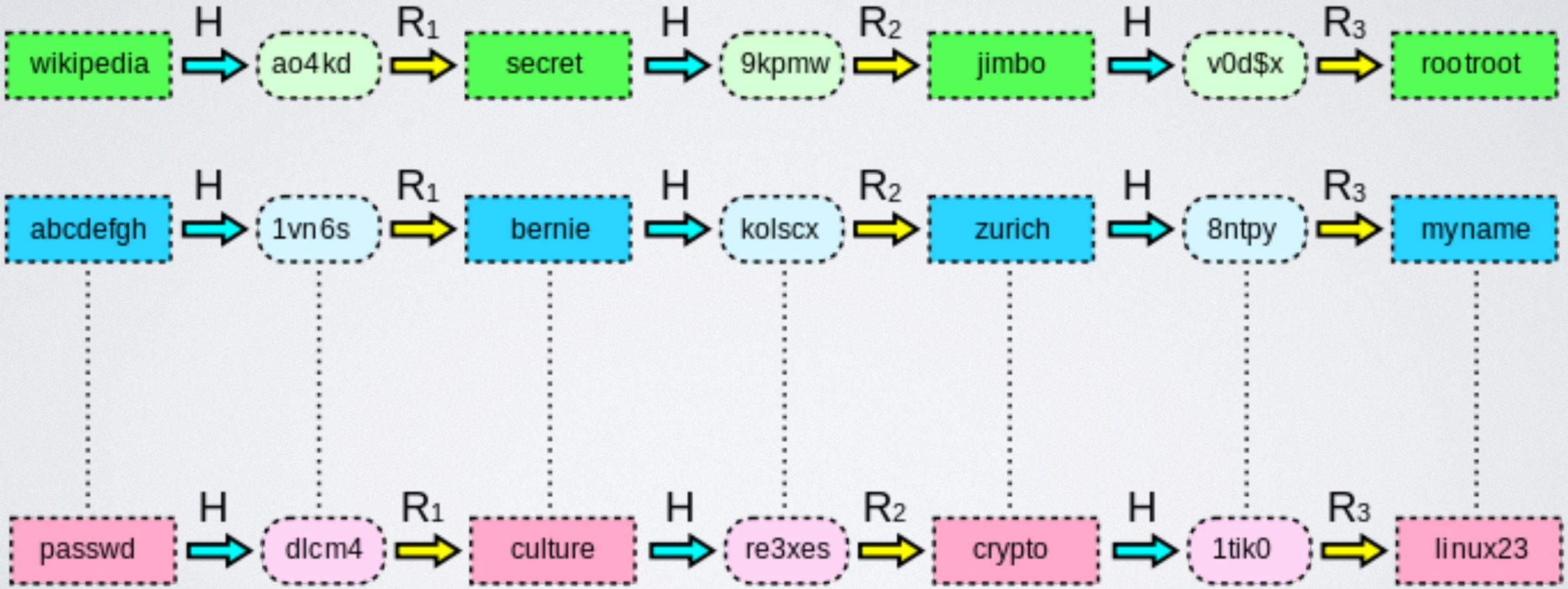
WORST CASE SZENARIO

- Speicherung im Klartext
- Üblicherweise Speicherung von Passwort-Hashes
  - Mögliche Angriffe?

RAINBOW TABLES

- Schnelle, probabilistische Suche nach Urbild eines Hashes
- Einmalige Berechnung, dann schnelles Nachschlagen
  - pro Hash-Funktion
  - abhängig von "*Reduktions-Funktion*"

RAINBOW TABLES



KENNWORTLÄNGE & ITERATIONEN

- lange Kennwörter
- Iterationen der Hash-Funktion
- RT werden unwirtschaftlich
  - Stromkosten
  - CPU Dauer
  - Speicherplatz

SALT & PEPPER

- Salt
  - zufällige Zeichenkette für jedes Passwort
  - assoziierte Speicherung
- Pepper
  - feste Zeichenkette für jedes Passwort
  - Speicherung an anderer Stelle

SCHUTZ?

- Server schützen (**Anbieter**)
- Starke Passwörter wählen (**Nutzer**)
  - Was ist ein starkes Passwort?

ANGRIFFS-VEKTOREN AUF HASHES

Brute-Force

Masken

Wörterbuch

Hybrid

Permutation

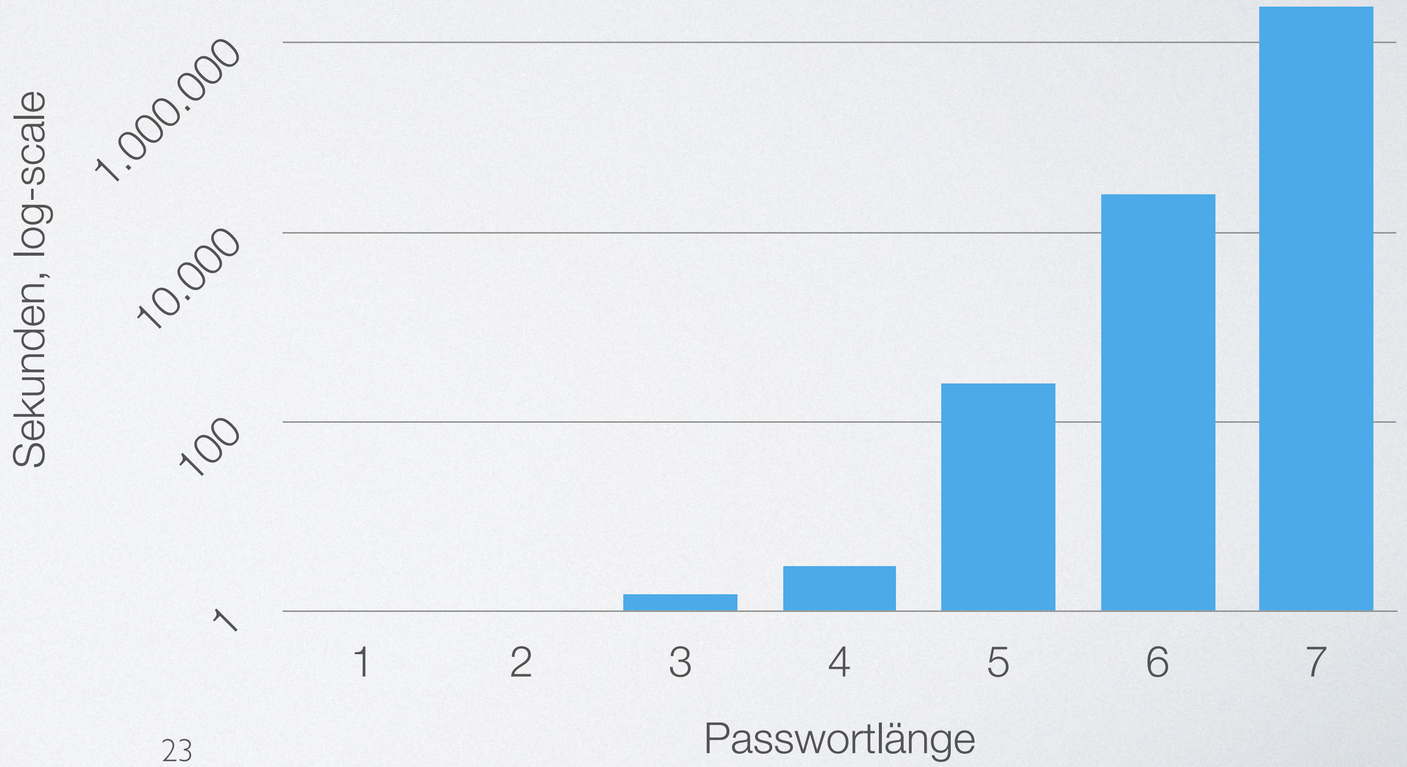
Regeln

PRINCE

BRUTE-FORCE

- Passwortlänge *n*
- # Zeichen *a*
  - abcdefghijklmnopqrstuvwxyz
  - ABCDEFGHIJKLMNOPQRSTUVWXYZ
  - 0123456789
  - «space»!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
- a^n
  - $64^6 = 68\ 719\ 476\ 736$
    ~ 25k Sekunden/7h @ 30M/s
- stößt schnell an Grenzen

Intel x86_64 i5-4260U CPU @ 1.40GHz w/ 4 threads
Case-sensitive alpha-numerisch plus Sonderzeichen



MASKEN

- Verfeinerung der Brute-Force-Methode
- Vorgabe des Zeichensatzes
- Passwort: julian1984
  md5 Hash: `0c6407c02eebe4d2713a3ff27db6486e`
  Maske: ?l?l?l?l?l?l?d?d?d?d
- $26^6 + 10^4 = 300M$, geknackt in etwa 10 Sekunden @ 30M/s
  vs. $64^{10} = 10^{18}$

WÖRTERBUCH

- Tatsächliche Wörterbücher
  - Deutsch, Englisch, Namen, Städte, Nachschlagewerke
- Echte Passwortlisten
  - Rockyou, MySpace, Anonymous, LinkedIn

DIE 50 SCHLECHTESTEN PASSWÖRTER

123456	letmein	6969	batman	hockey
password	baseball	jordan	trustno1	killer
12345678	master	harley	thomas	george
1234	michael	ranger	tigger	sexy
pussy	football	iwantu	robert	andrew
12345	shadow	jennifer	access	charlie
dragon	monkey	hunter	love	superman
qwerty	abc123	fuck	buster	asshole
696969	pass	2000	1234567	fuckyou
mustang	fuckme	test	soccer	dallas

HYBRID

- Wörterbücher + Brute-Force
- `${Wörter im Wörterbuch} + ?d?d?d?d`
  - `julian1984`

KOMBINATIONEN

- Kombiniere Wörterbucheinträge aus einem oder mehreren Wörterbücher
- `${Wort aus Buch1} + ${Wort aus Buch2}`

PERMUTATIONEN

- Erzeuge alle möglichen Zeichen-Permutationen für die Wörterbucheinträge
- `ABC, ACB, BAC, BCA, CAB, CBA`

REGELN

- Basiert ebenfalls auf Wörterbüchern
- Substitutionen
  - `a=@, s=$, e=3, o=0` (133t$p3@k there you go…)
  - 1=1, 1=2, 1=3, 1=4, …

BEISPIEL

- Wort: `word`
- Regeln: `o=o, o=0, o=0, o=., 1=1, 1=2, 1=3, 1=9`
- Ergebnis: `word1, w0rd1, w0rd1, w.rd1, word2, w0rd2, w0rd2, w.rd2, word3, w0rd3, w0rd3, w.rd3, word9, w0rd9, w0rd9, w.rd9`

FORTGESCHRITTEN

- Regel: `cT1$!^.fso0`

  `pA$$word`

  → `Pas$$word`

  → `PA$$word`

  → `PA$$word!`

  → `.PA$$word!`

  → `.PA$$word!!drow$$AP.`

  → `.PA$$w0rd!!dr0w$$AP.`

PRINCE

- **Pr**obability **In**finite **C**hained **E**lements
- erweiterte Kombinations-Attacke
  - 1 Wörterbuch
  - Ketten von Einträgen

KETTEN

- Zeichenlänge: 4
  - 4

    2 + 2

    1 + 3

    3 + 1

    1 + 1 + 2

    1 + 2 + 1

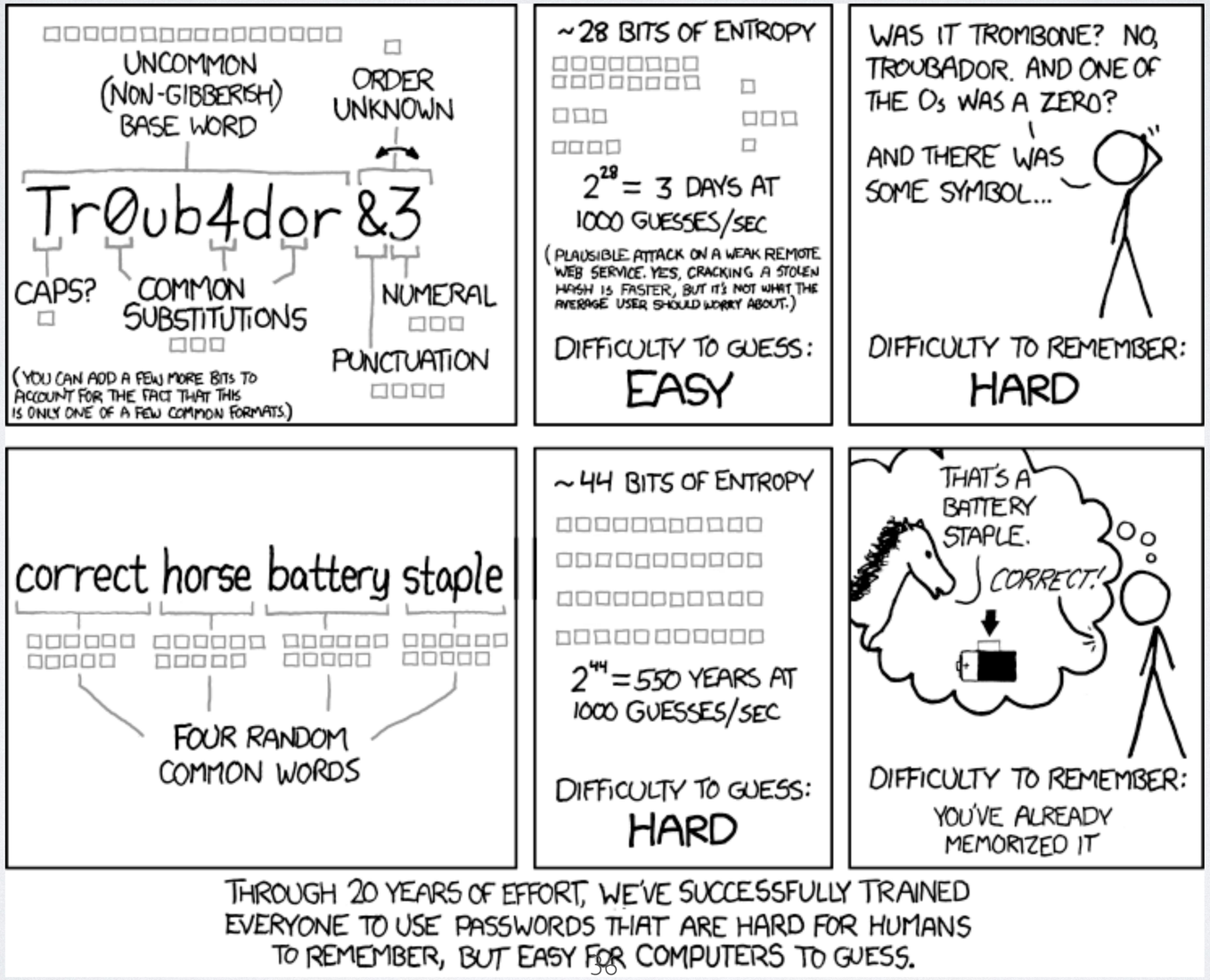
    2 + 1 + 1

    1 + 1 + 1 + 1

- Zerlegen des Wörterbuchs in Zeichenketten, dann erstellen aller möglichen Kombinationen
- "schlaue Kombinationen":
  - Schlüsselraum
  - Zeichenhäufigkeit

HANDS ON

STARKES PASSWORT



https://xkcd.com/936/

BEST PRACTICE

- min. 12 - 14 Zeichen
- case-sensitive alpha-numeric mit Sonderzeichen
- zufällig generieren
- Passwörter nicht mehrfach verwenden
- Vermeiden:
  - Zeichenwiederholung, Tastatur-Muster, Wörterbucheinträge, Buchstaben- oder Zifferfolgen, Benutzernamen, persönliche Informationen
  - persönliche Referenzen als Bestandteil
  - Kombinationen aus Vorangegangenem

PASSWORD MANAGER

- speichern, verwalten und generieren von Passwörtern
- abgesichert durch Verschlüsselung
- Master-Passwort

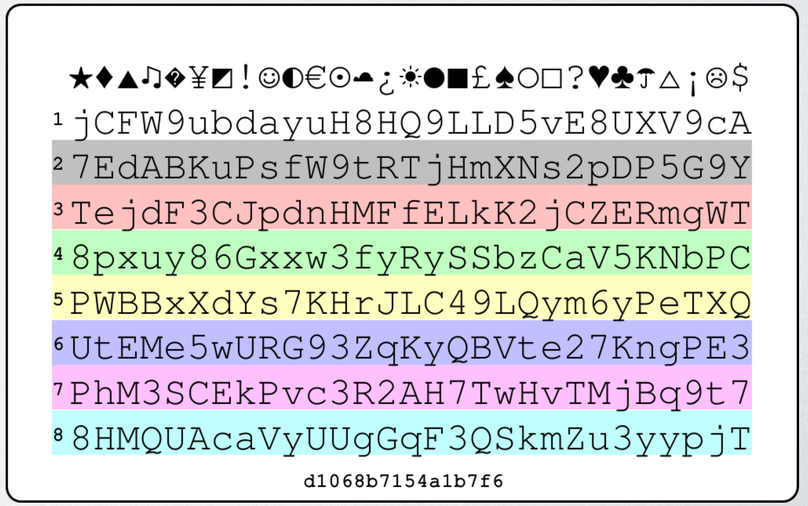
ANGEBOT

	OS	Browser	License
1 Password	OSX, iOS, Win, Android	✓	proprietär
KeePassX	OSX, Win, Linux	✓	GPLv2
LastPass	IE, Firefox, Chrome, Safari, Opera; (iOS, Android)	✓	proprietär
pass	Linux, OSX, Android	✓	GPLv2

ALTERNATIVEN

- Password Manager ohne Datenbank https://saltthepass.com
  - `md5(pAssw0rd + amazon.com)`
    `= ad08b81bc243e0e17d1d03392611a42a`
  - `md5(pAssw0rd + ebay.com)`
    `= 0cfad7aed6167461b85981858bef8ea9`
- PasswordCard

http://www.passwordcard.org/en



MULTI-FAKTOR-AUTHENTIFIZIERUNG

- authentifizierung anhand mehrerer unabhängiger Faktoren
  - z.B. Wissen + Besitz (PIN + Bankkarte)
    - **Wissen** Benutzername, Kennwort, TAN
    - **Besitz** Token, Bankkarte, Schlüssel
    - **Biometrisch** Fingerabdruck, Iris, Stimme

ONLINE 2FA

- Häufig: Passwort + Time-based-One-time-Password

  (z.B. https://github.com/google/google-authenticator)

- Passwort + SMS TAN o.ä.

UNTERSTÜTZER 2FA

- App.net
- Buffer
- Facebook
- Google+
- LinkedIIn
- Tumblr
- Twitter
- WordPress.com
- Apple
- eBay
- Etsy
- TeamViewer
- AWS
- PayPal
- Stripe
- Kickstarter
- IFTTT
- LastPass
- Rackspace
- Linode
- DigitalOcean
- Steam
- Origin
- Humble Bundle
- EVE Online
- Blizzard
- YouTube
- FastMail
- Gmail
- Outlook.com
- Yahoo Mail
- Yandex.Mail
- Gandi
- Name.com
- easyDNS
- CloudFlare
- GitHub
- Slack
- Mailchimp
- Heroku
- Google Cloud Platform
- OneDrive
- Dropbox
- Box
- iCloud

https://twofactorauth.org

FRAGEN?

https://github.com/ccauet/soak-passwords