Kapitel 5: Authentifikation und Identifikation

5: Authentifikation und Identifikation

Begriffe "*Authentifikation*", "*Authentisierung*" und "*Identifikation*" bezeichnen ähnliches, aber nicht das Gleiche.

Im IT-Sprachgebrauch wird zwischen diesen Begriffen oft nicht präzise unterschieden.

Begriffsdefinitionen

Authentisierung

Daten werden hinsichtlich ihrer Echtheit (Unversehrtheit) überprüft.

Durch digitale Signaturen kann die Echtheit von Dokumenten bestimmt werden.

Authentifikation

Eine Entität weist Berechtigungen nach.

Client authentisiert sich beim Server.

Identifikation

Eine Entität weist ihre Identität nach.

Person weist ihre Identität mittels eines Reisepasses nach.

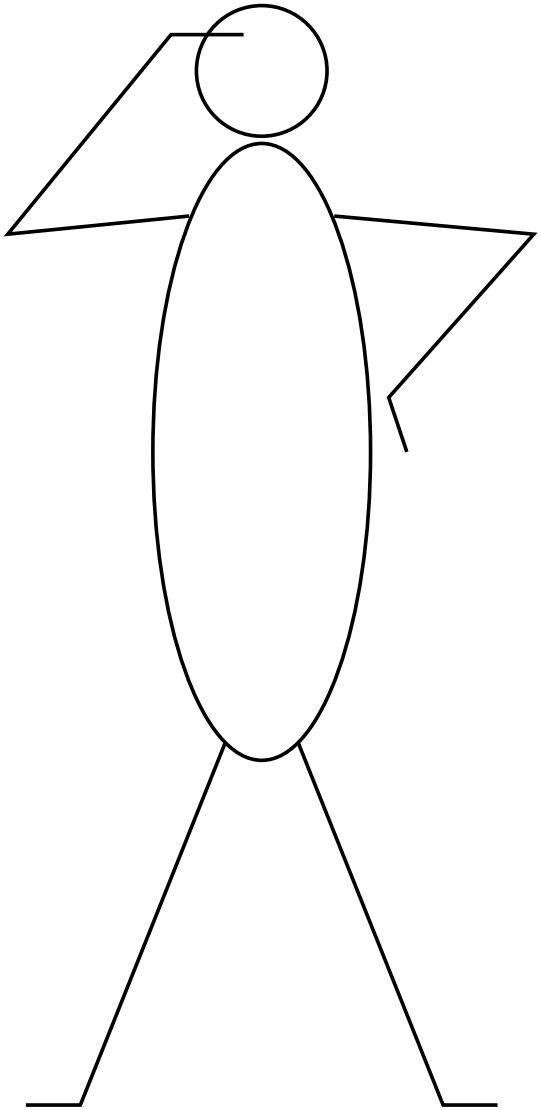
Identifikation

- Einseitig (Authentifikator $\rightarrow$ Authentisator)
  - Mensch $\rightarrow$ Mensch (*"Den Ausweis bitte!"*)
  - Mensch $\rightarrow$ Rechner (Geld-Automat)
  - Rechner $\rightarrow$ Rechner
    (SSL-Protokoll zwischen Server und Browser)

- Gegenseitig
  (*"Kriminalpolizei! . . . Darf ich 'mal bitte Ihren Ausweis sehen?"*)
  - Mensch $\leftrightarrow$ Mensch (Spione)
  - Rechner $\leftrightarrow$ Rechner (IP-Sec)
  - Mensch $\leftrightarrow$ Rechner (eher selten)

5.1: Benutzer-Identifikation

Identifikation durch



körperliche Eigenschaften (Biometrie)

Besitz von Gegenständen

Kenntnis von Geheimnissen

Merkmale

Biometrische Merkmale:

Möglichst eindeutiges, unveränderliches und schwierig zu fälschendes körperliches Merkmal
(Gesicht, Stimme, Fingerabdruck, Retina-Muster, DNA, . . .)

Besitz von Gegenständen:

Eindeutig und schwierig zu duplizieren
(Ausweis, Metallschlüssel, EC-Karte, TANs/Einmalpassworte, . . .)

Kenntnis von Geheimnissen:

(Antworten auf Fragen, Passwort, PIN, Passphrase, . . .)

Anmerkungen

Jede der drei Gruppen von Authentifikationsmerkmalen hat, bezogen auf die Sicherheit, spezifische Schwächen.

Aufgabe: Überlegen Sie sich in Gruppen Vor- und Nachteile der vorgestellten Merkmale.

Bei geringen Sicherheitsanforderungen genügt i.d.R. ein Merkmal ("Single-Factor Authentication").

Bei mittleren bis höheren Sicherheitsanforderungen ist es empfehlenswert, mehrere Merkmale aus *mindestens zwei verschiedenen Merkmalsklassen* einzusetzen.
Man spricht dann von einer "**Two-Factor Authentication**".

Beispiel: PIN und EC-Karte am Geldautomaten.

5.2: Passwörter

- *PIN* (“Personal Identification Number”) typischerweise Ziffernfolge
- *Passwort*, *Passphrase*, *Mantra*: Buchstaben und Sonderzeichen
- *Passphrase* synonym zu "Passwort", impliziter Hinweis auf eine *lange* Zeichenfolge
- *Mantra* merkwürdige Eindeutschung von “Passphrase”

Häufige Fehler beim Umgang mit Passwörtern

- Passwort aufgeschrieben ("PostIt Zettel auf Monitor")
- Passwort mangelhafter Qualität (zu kurz / aus dem Wörterbuch)
- “Social Engineering”
- "über die Schulter schauen"
- Passwort bei Fehleingabe preisgegeben
- Das gleiche Passwort in verschiedenen Systemen

Kurze Diskussion

1. Frage: Wie schlimm ist es, ein sehr kurzes Passwort zu verwenden?
   (Die PIN für den Geldautomaten besteht ja auch nur aus 4 Dezimalziffern.)

2. Frage: Wie schlimm ist es, das gleiche Passwort in verschiedenen Systemen zu verwenden?

3. Frage: Wie schlimm ist es, sich das Passwort aufzuschreiben?

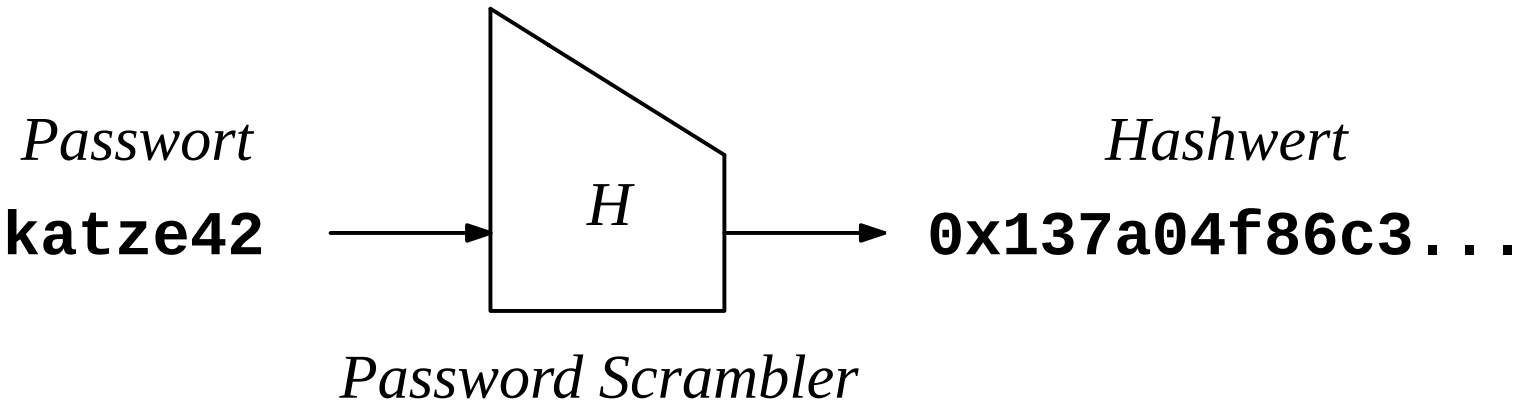
Top 1000 Passwörter

(Quelle: `http://xato.net`)

Passwort Dilemma

- *Schlechtes* Passwort (einfach zu *erraten*)
- *Gutes* Passwort (schwer zu merken)

Speicherung von Passwörtern



- System speichern normalerweise nur den Fingerabdruck eines Passworts.
- Fingerabdruck eines Passworts = Hashwert = Passworthash
- Beim Anmelden werden daher nur die Fingerabdrücke verglichen.
- Eine *Hashfunktion* berechnet aus einem Passwort dessen Fingerabdruck (`DEMO: sha1sum, /etc/shadow`).

Einwegfunktion

- (Kryptographische) Hashfunktionen sind Einwegfunktionen.
- Dies bedeutet, dass es praktisch unmöglich ist für gegeben Hashwert ein passendes Urbild zu finden.
- Sei $H : \{0,1\}^* \rightarrow \{0,1\}^n$ eine kryptographische Hashfunktion.
- Gegeben: y mit $y \leftarrow H(x)$ für zufälliges x.
- Gesucht: Preimage z für y ($H(z) == y$)
- Aufwand für die Preimagesuche ist unpraktikabel (ca. $O(2^n)$)

Preisfrage

Warum ist das mit der Einwegfunktion eine gute Idee?

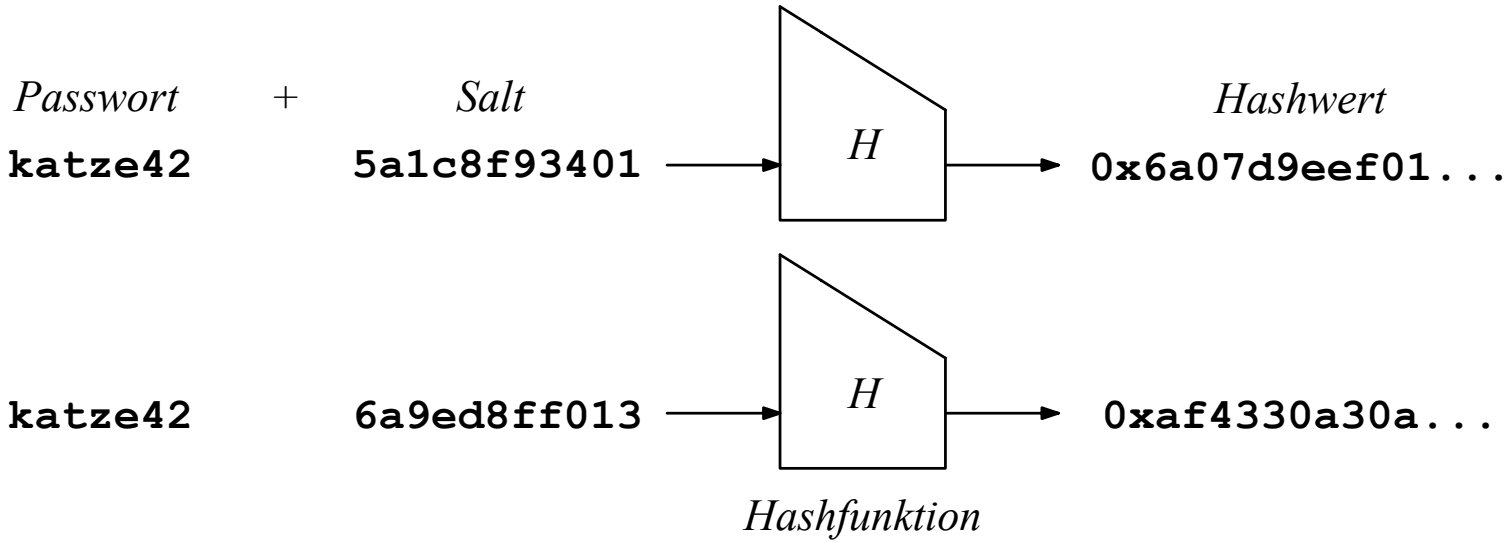
- Datei mit den Fingerabdrücken kann *verloren* gehen.
  - Adobe (verliert ca. 150 Millionen Passworthashes)
  - LinkedIn (verliert ca. 6.5 Millionen Passworthashes)
  - Yahoo (verliert ca. 0.5 Millionen Klartextpasswörter)
  - ...
- Angreifer wäre dann in der Lage die dazugehörigen Passwörter zu berechnen.

Was bleibt dem Angreifer noch übrig?

Angreifer probieren alle möglichen Passworte durch.

Salt

Password Scrambler (Verfahren zum Passwort hashen) verwenden Salt (zusätzlicher zufälliger Input).



- Server speichert sich den Salt und den Hashwert.
- Saltbits die nicht gespeichert werden nennt man Pepper.
- **Frage:** Warum ist Salt eine gute Idee?

Wie arbeiten Angreifer typischerweise?

- Wir unterscheiden zwischen Online- und Offline-Angreifern
- Online-Angreifer:
  - Benötigt Interaktion mit einem Webserver oder Gerät.
  - Nur Server/Gerät kann Passworthash berechnen.
  - Gut: Nur so viele Versuche wie Server/Gerät zulässt.
  - Gut: Potentiell hohe Wartezeit nach einigen Versuchen.
- Offline-Angreifer:
  - Kann Passworthashes selbst testen
  - Wird nicht gedrosselt!
  - Hat beliebig viele Versuche!

Offline-Angreifer

- Problem: Hashfunktionen sind viel zu effizient (nicht für Passwörter entwickelt)
- Passwörter lassen sich sehr schnell durchprobieren, z.B.:
  - ca. 1,2 Mrd./s für aktuelle Hashfunktionen (SHA2)
  - ca. 11 Mrd./s für Windows-XP-Passworthashes
- Derzeit nutzen Angreifer i.d.R. viele Grafikkarten

Quelle: `http://hashcat.net/`

Passwort-Crack-Programme

- Freie gute Software für diverse Hashfunktionen:
  - John the Ripper (für CPUs)
    `http://www.openwall.com/john/`
  - DaveGrohl (für CPUs)
    `http://davegrohl.org/`
  - oclHashcat (für Grafikkarten)
    `http://hashcat.net/oclhashcat-plus/`
- Werden immer besser:
  - Testen erst Wörterbücher
  - Testen auch beliebte Verfremdungen
  - Sortieren Passworte nach Wahrscheinlichkeit

$\Rightarrow$ Nur ausreichend lange und zufällige Passwörter sind sicher!

5.3: Exkurs Entropie

Die Entropie, soll den *Informationsgehalt* einer Nachrichtenquelle angeben.

Entropie

Anzahl Bits die mindestens gebraucht werden, um die Information einer Nachrichtenquelle darzustellen.

Kurz: Entropie = Mindestanzahl Bits zur Speicherung

- Eine Quelle, die permanent immer die gleiche Information versendet, verbreitet keine Information.
- Eine Quelle, die das Ergebnis "Kopf" oder "Zahl" eines fairen Münzwurfs liefert, hat 1 bit Entropie.

Was hat die Länge der (komprimierten) Darstellung mit der Sicherheit eines Passwortes zu tun?

Wenn man zum Speichern eines unbekannten Wertes mindestens b bit braucht, und ich versuche, diesen Wert zu erraten, dann kann ich statistisch bestenfalls erwarten, dies nach 2^{b-1} Versuchen zu schaffen.

Die Entropie einer Informationsquelle (für uns insbesondere eines Verfahrens, Passwörter zu generieren!) gibt an, welchen Aufwand man im Durchschnitt treiben muss, um ein Passwort aus dieser Quelle zu *knacken*.

Beispiel

Die Informationsquelle erzeugt die Buchstaben "a", bis "d", und zwar

- "a" mit der Wahrscheinlichkeit $\Pr[\text{"a"}] = 0.5$,
- "b" mit der Wahrscheinlichkeit $\Pr[\text{"b"}] = 0.25$, und
- "c" und "d" mit $\Pr[\text{"c"}] = \Pr[\text{"d"}] = 0.125$?

Frage: Wie viele Bits werden benötigt, um die Wörter dieser Informationsquelle zu speichern?

Shannon-Entropie

Definition 6.1 (Shannon-Entropie)

Sei Q eine Informationsquelle mit $|Q| = n$ und seien $p_i \in Q, 1 \leq i \leq n$ die Wahrscheinlichkeiten des Auftretens der Einzelereignisse in Q. Dann ist die Shannon-Entropie einer Quelle Q gegeben durch:

$$H_1(Q) = -\sum_i p_i \cdot \log_2(p_i).$$

Die Shannon-Entropie gibt den mittleren Informationsgehalt einer Quelle Q an.

Beispiel

Gegen Informationsquelle Q (gezinkter Würfel) erzeugt Zahlen 1 bis 6 mit der folgenden Wahrscheinlichkeit.

- $\Pr[1] = 4/16$
- $\Pr[2] = 1/16$
- $\Pr[3] = 1/16$
- $\Pr[4] = 4/16$
- $\Pr[5] = 2/16$
- $\Pr[6] = 4/16$

Frage: Wie hoch ist die Entropie der Informationsquelle Q?

Frage: Ist die Entropie eines fairen Würfels höher?

Beobachtungen: Shannon-Entropie

Satz 6.1

Ist Q eine gleichverteilte Quelle von n Elementen, dann

$$H_1(Q) = \log_2(n)$$

Satz 6.2

Ist Q nicht gleichverteilt, dann gilt

$$H_1(Q) < \log_2(n).$$

Folgerung

Egal wie Q verteilt ist, stets gilt

$$0 \leq H_1(Q) \leq \log_2(n).$$

Beispiel Fairer Würfel

Bei dem fairen Würfel gilt: $p_i = 1/6$ und $\log_2(1/6) \approx -2.58$.

$$\begin{aligned} H_1(\textbf{Fairer Würfel}) &= -(6 \cdot (1/6 \cdot \log_2(1/6)) \\ &= -\log_2(1/6) \\ &= \log_2(6) \\ &= 2.58 \end{aligned}$$

Entropie bei unabhängigen Zufallsquellen

Für zwei Quellen Q und R bezeichnet QR die Menge zufälliger Paare (q, r), von Elementen q aus Q und r aus R.

Satz 6.3

Sind Q und R zwei voneinander unabhängige Quellen, dann ist

$$H_1(QR) = H_1(Q) + H_1(R).$$

Es bezeichne Q^i die Quelle, die durch i-faches und unabhängiges Ziehen eines Elements aus Q definiert ist.

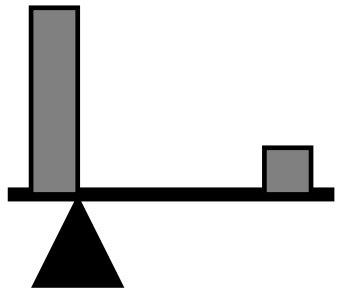
Folgerung

$$H_1(Q^i) = i * H_1(Q).$$

Die Min-Entropie

- Die Shannon-Entropie fragt nach der *durchschnittlichen* Anzahl an Bits, die man braucht, um die Wörter aus einer Quelle zu speichern.

- Eine Eigenschaft des *Durchschnitts* ist jedoch, dass es Konstellationen gibt, bei denen fast alle Werte unterdurchschnittlich und wenige überdurchschnittlich sind (siehe Abb. rechts).



- Bezogen auf die Passwort-Sicherheit kann das zu optimistischen Schlussfolgerungen führen.

- Deshalb betrachtet man alternativ zur Shannon-Entropie auch die Min-Entropie.

Die Min-Entropie

als “pessimistische” Alternative zur Shannon-Entropie

Sei Q eine Quelle ...

Definition 6.2 (Min-Entropie)

Die **Min-Entropie** von Q ist

$$H_\infty(Q) = \min_i \{-\log_2(\Pr[q_i])\} = -\log_2(\max(\Pr[q_i])).$$

Satz 6.4

Allgemein gelten

$$0 \leq H_\infty(Q) \leq H_1(Q) \leq \log_2(n) \quad \textit{und} \quad H_\infty(Q^i) = i \cdot H_\infty(Q).$$

Ist Q gleichverteilt, gilt sogar $H_\infty(Q) = H_1(Q) = \log_2(n)$.

Beispiel: Min Entropie

Gegen Informationsquelle Q (gezinkter Würfel) erzeugt Zahlen 1 bis 6 mit der folgenden Wahrscheinlichkeit.

- $\Pr[1] = 4/16$
- $\Pr[2] = 1/16$
- $\Pr[3] = 1/16$
- $\Pr[4] = 4/16$
- $\Pr[5] = 2/16$
- $\Pr[6] = 4/16$

Frage: Was ist die Min-Entropie der Informationsquelle Q?

Frage: Ist die Min-Entropie eines fairen Würfels höher?

5.4: Passwortentropie

Was sind ausreichend lange Passwörter?

- 96 druckbare Zeichen
  - Klein-/Großbuchstaben, Ziffern, Satz- und Sonderzeichen

- Beispiel: Passwort aus 6 Zeichen
  - Eine von 96^6 Möglichkeiten
  - Informatik rechnet in 2er-Potenzen: $96^6 \approx 2^{40}$ Möglichkeiten

  $\Rightarrow$ 40 Bit **Entropie**

Hashcat

- Hashcat ist ein *Advanced Password Reovery Tool*
- Es verwendet GPUs um Passwörter *wieder herzustellen*.
- Hashcat unterstützt klassische Verfahren um Passworthashes zu erstellen.
- Geschwindigkeiten für GeForce RTX 3090

Hashverfahren	**Geschwindigkeit**
MD5	65 GH/s ($\approx 2^{36}$ Test/s)
SHA-1	22 GH/s ($\approx 2^{35}$ Test/s)
MD5Crypt	32 MH/s ($\approx 2^{25}$ Test/s)
SHA512Crypt	483 KH/s ($\approx 2^{19}$ Test/s)
PBKDF2-SHA256	129 KH/s ($\approx 2^{17}$ Test/s)
bcrypt	102 KH/s ($\approx 2^{17}$ Test/s)
scrypt	2 KH/s ($\approx 2^{11}$ Test/s)

(Quelle: `https://gist.github.com/Chick3nman/e4fcee00cb6d82874dace72106d73fef`)

- Hashverfahren ist signifikant für die Passwortsicherheit.

Leaks

- 2008 MySpace: ca, 360 Millionen Accounts (SHA1)
- 2012 Dropbox: ca. 68 Millionen Accounts (SHA1 und bcrypt)
- 2012 Last.fm: ca. 37 Millionen Accounts (MD5)
- 2012 YouPorn: ca. 1,3 Millionen Accounts (Plaintext)
- 2013 Badoo: ca. 112 Millionen Accounts (MD5)
- 2014 Domino's: ca. 0,5 Millionen Accounts (MD5)
- 2014 Kickstarter: ca. 5 Millionen Accounts (SHA1)
- 2015 Patreon: ca. 2,5 Millionen Accounts (bcrypt)
- 2017 LinkedIn: ca 165 Million Accounts (SHA1)
- 2018 Creative: ca. 0.5 Millionen Accounts (MD5)

(Quelle: `https://haveibeenpwned.com/PwnedWebsites`

Password Recovery Zeiten (GeForce RTX 3090)

Passwortlänge (Entropie)	Recovery Zeiten MD5	bcrypt
6 (40 Bit)	16 Sekunden	3 Monate
7 (46 Bit)	17 Minuten	17 Jahre
8 (52 Bit)	18 Stunden	1104 Jahre
9 (59 Bit)	3 Monate	2^{17} Jahre
10 (65 Bit)	17 Jahre	2^{23} Jahre
12 (79 Bit)	2^{18} Jahre	2^{37} Jahre

Alter des Universums: ca. 14 Mrd. Jahre $\approx 2^{34}$ Jahre.

Password Hashing Competition (PHC)

- 2012: Alle Password Hashing Verfahren bis auf `scrypt` und `bcrypt` sind anfällig gegen GPU Angriffe
  - `scrypt` hatte einen eingebauten GPU Schutz (speicherintensiv + nicht parallelisierbar)
  - `bcrypt` läuft auf heutigen GPUs zufällig langsam
- 2012: Ankündigung des PHC mit dem Ziel der Standardisierung eines modernen (GPU-resistenten) Password Scramblers
- Q1 2013: Call for Submission: 24 Einreichungen
- Q2 2015: Argon2 wird zum Gewinner ernannt
- Es gab 4 Finalisten mit besonderer Auszeichnung
  1. Catena
  2. Lyra2
  3. Makwa
  4. yescrypt

Faktor Mensch

- Die Sicherheit reduziert sich drastisch
  wenn Passwörter keine Zufallskombinationen sind!
  - “`A0!94%l+5_`” = mehrere Wochen auf Tausenden Rechnern
  - “`G3h31m007!`” = wenige Minuten auf einer (!) Grafikkarte

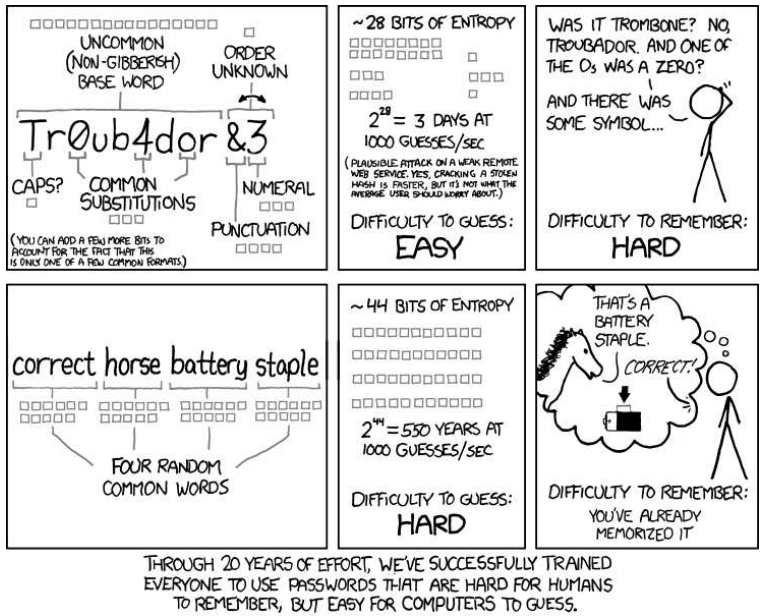
- Problem: Menschen sind schlechte Passwortgeneratoren

Schlechte Passwort-Regeln

- Erratbare Begriffe:
  - “`BHT`”, “`Hochschule`”
- Namen oder Stichtage:
  - “`Helga`”, “`20sep1993`”
- Wort aus dem Wörterbuch, auch Verfremdung hilft nicht:
  - “`Lichtgeschwindigkeit`”, “`Sh3ttl4nd-TerrIer`”, “`Tr0ub4dour`”
- Wortkombinationen:
  - “`Adam2+7Eva`”
- Bekannte Begriffe:
  - “`supercalifragilisto5`” (aus dem Musical Mary Poppins)

Gute Passwort-Regeln (1/2)

- Zufällig Passwörtern mit 13 Zeichen
  - “`as8%4,&xn9?`”
  - $\approx$ 85 Bit Entropie
  - Problem: Wie merke ich mir eine solche Kombination?
- Einfacher: Kombination aus mind. fünf zufällig gewählten Wörtern
  - “`korrekt Pferd Batterie Büroklammer Magnet`”
  - $\approx$ 85 Bit Entropie



Quelle `https://xkcd.com/936/`

Gute Passwort-Regeln (2/2)

- Besser: Merksätze mit mind. 19, besser 22 und mehr Zeichen
  - "`IbhdVIT-SadBHTulsvNüP`" = Ich besuche heute die Vorlesung IT-Sicherheit an der BHT und lerne sehr viel Neues über Passwortsicherheit.
    - 19 Zeichen: > 60 Bit Entropie
    - 22 Zeichen: > 70 Bit Entropie
- Noch besser: Bei den Merksätze *der/die/das* ignorieren (weniger vorhersagbar)

Gemeine Preisfrage

Warum ist "`IbhdVIT-SadBHTulsvNüP`" jetzt kein gutes Passwort mehr?

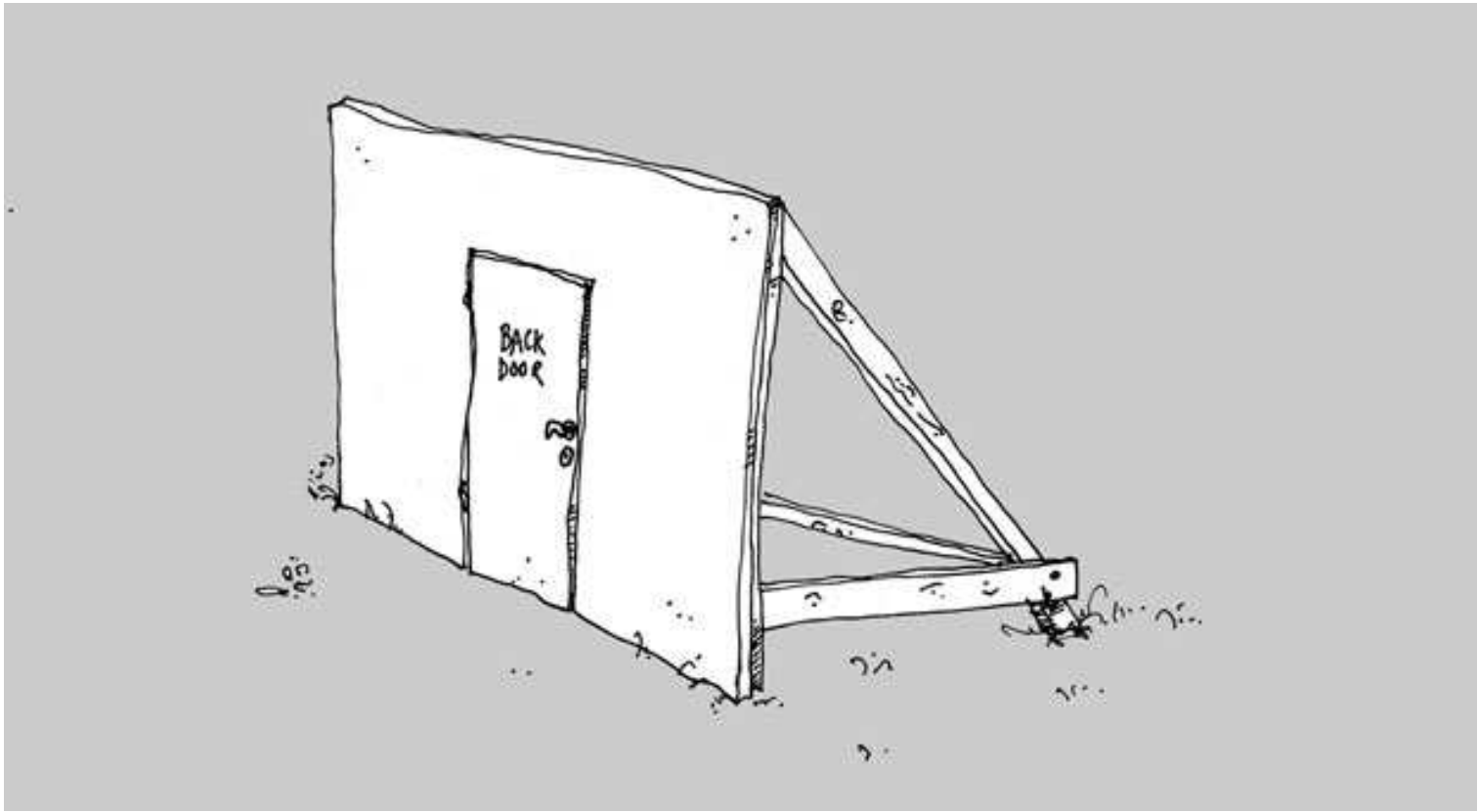
Guter Umgang mit Passwörtern (1/2)

- Nutzen Sie ruhig Passwortgeneratoren für zufällig gewählte Passwörter
  - Z.B. PWGen: `http://pwgen-win.sourceforge.net/`
  - Vertrauen Sie Online-Programmen nicht!
- Passwörter aufschreiben?
  - Ja, wenn man sie sicher verwahrt
- Alternative: Passwortsafe-Programme
  - KeyPass: `http://keepass.info/`
  - Password Safe: `https://pwsafe.org/`
  - Problem: Synchronisierung zwischen mehreren Geräten

Guter Umgang mit Passwörtern (2/2)

- Nutzen Sie zufällige Salts mit mind. 64 Bits
- Nutzen Sie aktuelle “Key-Derivation Functions” (KDF) anstelle reiner Hashfunktionen.
  - `PBKDF2`, `sha512crypt`, `scrypt`, `Argon2`
  - Verwenden Tausend(e) Iterationen einer Hashfunktion
  - Stört den Nutzer nicht (0.1 Sekunden)
  - Bremst aber Angriffe aus (bis Faktor 1.000.000)
  - Für Betriebssysteme schon lange Standard
- Beispiele:
  - Debian GNU/Linux: `sha512crypt` (konfigurierbar)
  - Windows (seit NT 4.0 SP4): `NTLanManagerv2`
  - OS X (seit 10.8): `PBKDF2`

Gute Passwörter alleine reichen nicht (1/2)



Quelle: `http://www.raumlabor.net/?p=502`

Auch das beste Passwort nützt nichts, wenn man den Schutz einfach umgehen kann.

Gute Passwörter alleine reichen nicht (2/2)

- Schließen Sie potentielle Hintertüren
  - Lügen Sie bei PW-Wiederherstellungsfragen (“erstes Auto”, “Mädchenname der Mutter”)
- Verwenden Sie Passwörter nicht mehrfach
  - Wegwerf-Passwörter für unwichtige Seiten
  - Master-Passwort für versch. Dienste variieren, solange nicht nachvollziehbar für Andere
- Speichern Sie Passwörter niemals online
  - Dropbox und co. lesen Ihre Dateien (ja, wirklich!)

Zusammenfassung

Sie sollten . . .

- . . . die drei Merkmale der Benutzer-Identifikation kennen.
- . . . die häufigen Fehler beim Umgang von Passwörtern kennen.
- . . . wissen wie man Passwörter speichern sollte.
- . . . wissen was man unter Entropie von Passwörtern versteht.
- . . . die Regeln für gute und schlechte Passwörter kennen.