Lecture 10: Password-Based Authentication

CS 181W

Fall 2022

Classes of Security Countermeasures

- **Authentication:** mechanisms that bind principals to actions
- **Authorization:** mechanisms that govern whether actions are permitted
- **Audit:** mechanisms that record and review actions

Classes of Principals

- **Authentication:** mechanisms that bind principals to actions

  - Authenticating Machines
  - Authenticating Programs
  - Authenticating Humans

Authentication of humans

- Something you are

  biometrics (e.g., fingerprints)

- Something you know

  secret information (e.g., a password)

- Something you have

  possession of a physical device (e.g., a particular phone)

Exercise: Authentication Mechanisms

- What are different ways you have authenticated yourself to a machine? How should we classify them?

Something you are

Something you know

Something you have

Multi-factor Authentication

- Two-factor authentication: authenticate based on two independent methods
  - ATM card plus PIN
  - password plus registered mobile phone
- Multi-factor authentication: two or more independent methods
- Best to combine separate categories, not reuse categories
  - non-example: requiring two passwords from a single human: arguably not independent
  - non-example: requiring single password from each of two humans: authenticates two humans then makes *authorization* decision

PASSWORDS

Password lifecycle

1. **Create:** user chooses password
2. **Store:** system stores password with user identifier
3. **Use:** user supplies password to authenticate
4. **Change/recover/reset:** user wants or needs to change password

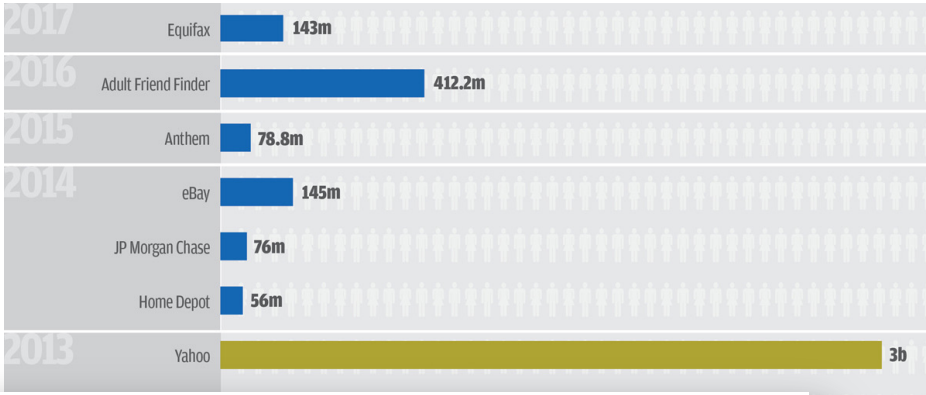
2. PASSWORD STORAGE

Password Storage

- Passwords typically stored in a file or database indexed by username
- **Strawman idea:** store passwords in plaintext
  - requires perfect authorization mechanisms
  - requires trusted system administrators
  - ...

Threat Model: Offline Attack

- Adversary can read files from disk



Hackers steal 46 million Animal Jam user

ALERT: 1,583,193 Breached Accounts At VPN Provider ActMobile

One of the biggest Android VPNs hacked? Data of 21 million users from 3 Android VPNs put for sale

Cybernews Team | Updated on: 06 June 2022

Password Storage

- **Want:** a function f such that...
  1. easy to compute and store f(p) for a password p
  2. hard given disclosed f(p) for attacker to recover p
  3. hard to trick system by finding password q s.t. q != p yet f(p) = f(q)

- Encryption would work, but then the key has to live somewhere

- Cryptographic hash functions suffice!
  - one-way property gives (1) and (2)
  - collision resistance gives (3)

Hashed passwords

- Each user has:
  - username uid
  - password p
- System stores: uid, H(p)

Exercise: Hashed Passwords

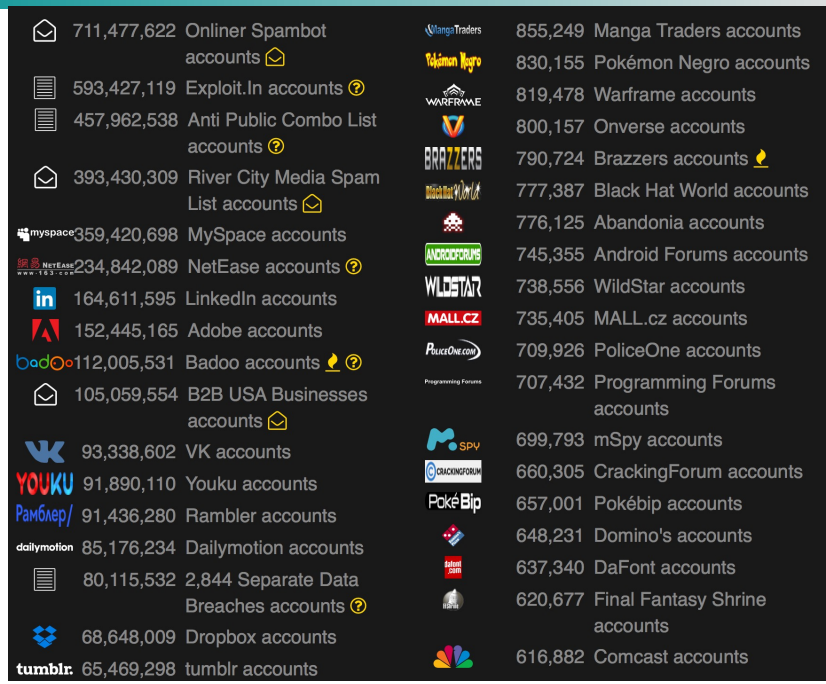
- Consider an alternative authentication protocol where user sends uid, H(p) and the service compares H(p) to the stored hash. Would this be more or less secure than sending the plaintext password? Why?

Hashed passwords are still vulnerable

Assume: attacker does learn password file *(offline guessing attack)*

- Hard to invert: i.e., given H(p) to compute p
- But what if attacker didn't care about inverting hash on arbitrary inputs?
  - i.e., only have to succeed on a small set of p's: p1, p2, ..., pn
- Then attacker could build a **dictionary**...

Dictionary attacks



Dictionary:

- p1, H(p1)
- p2, H(p2)
- ...
- pn, H(pn)

- Dictionary attack: lookup H(p) in dictionary to find p

Salted hashed passwords

- **Vulnerability:** one dictionary suffices to attack every user
- **Vulnerability:** passwords chosen from small space
- **Countermeasure:** include a **unique system-chosen nonce** as part of each user's password

Salted hashed passwords

- Each user has:
  - username uid
  - unique salt s
  - password p
- System stores: uid, s, H(s, p)

3. PASSWORD USAGE

Authenticating to a remote server

- Each user has:
  - username uid
  - unique salt s
  - password p
- System stores: uid, s, H(s, p)

```
1. Hu->L: uid, p
2. L and S: establish secure channel
3. L->S: uid, p
4. S: let h = stored hashed password for uid;
      let s = stored salt for uid;
      if h = H(s, p)
      then uid is authenticated
```

Threat Model: Online Attack

- Adversary can interact with the server as a user

When authentication fails

- **Guiding principle:** the system might be under attack, so don't make the attacker's job any easier
- Don't leak valid usernames:
  - Prompt for username and password in parallel
  - Don't reveal which was bad
- Record failed attempts and review
  - Perhaps in automated way by administrators
  - Perhaps manually by user at next successful login
- Lock account after too many attempts
- Rate limit login

Rate limiting

- **Vulnerability:** hashes are easy to compute
- **Countermeasure:** hash functions that are slow to compute
  - Slow hash wouldn't bother user: delay in logging hardly noticeable
  - But would bother attacker constructing dictionary: delay multiplied by number of entries
  - Ideally, enough to make constructing a large dictionary prohibitively expensive
- Examples: bcrypt, scrypt, Argon2,...

Slowing down fast hashes

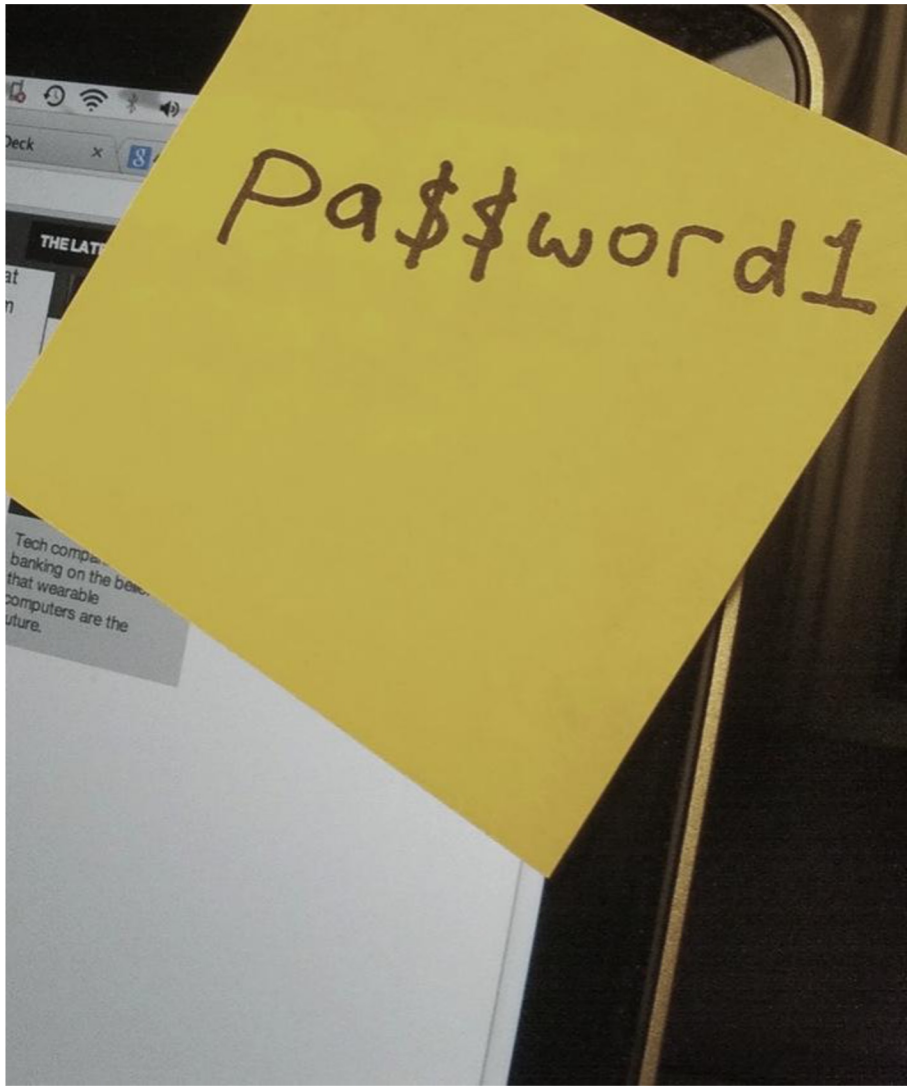
- Given a fast hash function...
- Slow it down by iterating it many times:

```
z1 = H(p);
z2 = H(p, z1);
...
z1000 = H(p, z999);
output z1 XOR z2 XOR ... XOR z1000
```

- Number of iterations is a parameter to control slowdown
  - originally thousands
  - current thinking is 10s of thousands
- Aka key stretching

Password vulnerabilities

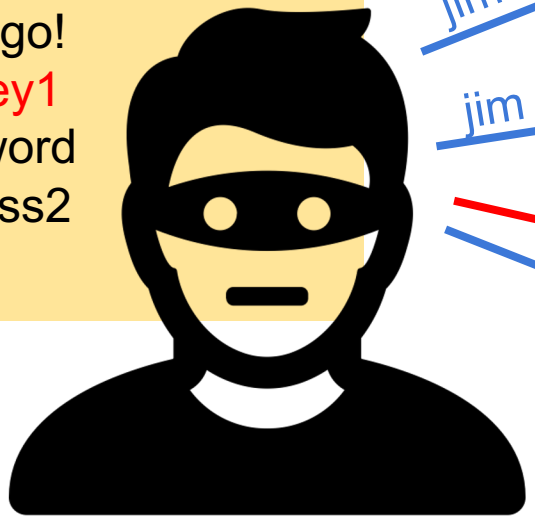
- Shoulder surfing attacks
- Online attacks
- Offline attacks



Attackers exploit password reuse

CRACKED PASSWORDS

UserID	Password
jane	iloveyou89
jami	godoggo!
jim	monkey1
kar	pa$$word
katie	princ3ss2



jim monkey1 → Online Store

jim monkey1 → Bank

jim monkey2 → Employer

jim monkey1 → Employer

1. PASSWORD CREATION

Strong passwords

- How to characterize strength?
- **One Approach:** Difficulty to brute force—"strength" or "security level"
  - if 2^X guesses required, strength is X
- Suppose passwords are L characters long from an alphabet of N characters
  - Then N^L possible passwords
  - Solve for X in 2^X = N^L
  - Get X = L $\log_2$ N
  - This X is aka **entropy** of password
    - Assuming every password is equally likely, X is the *Shannon entropy of the probability distribution* (cf. Information Theory)

Exercise: Entropy of passwords

- Option A: 8 character passwords chosen uniformly at random from 26 character alphabet

- Option B: 1 word chosen at random from entire vocabulary
  - average high-school graduate: 50k word vocabulary

Exercise: Entropy of passwords

- Option A: 8 character passwords chosen uniformly at random from 26 character alphabet
  - entropy of $8 \log_2 26 \approx 37$ bits
  - but that means abcdefgh equally likely as ifhslgqz

- Option B: 1 word chosen at random from entire vocabulary
  - average high-school graduate: 50k word vocabulary
  - entropy of $\log_2 50k \approx 16$ bits
  - but that assumes all words are equally likely

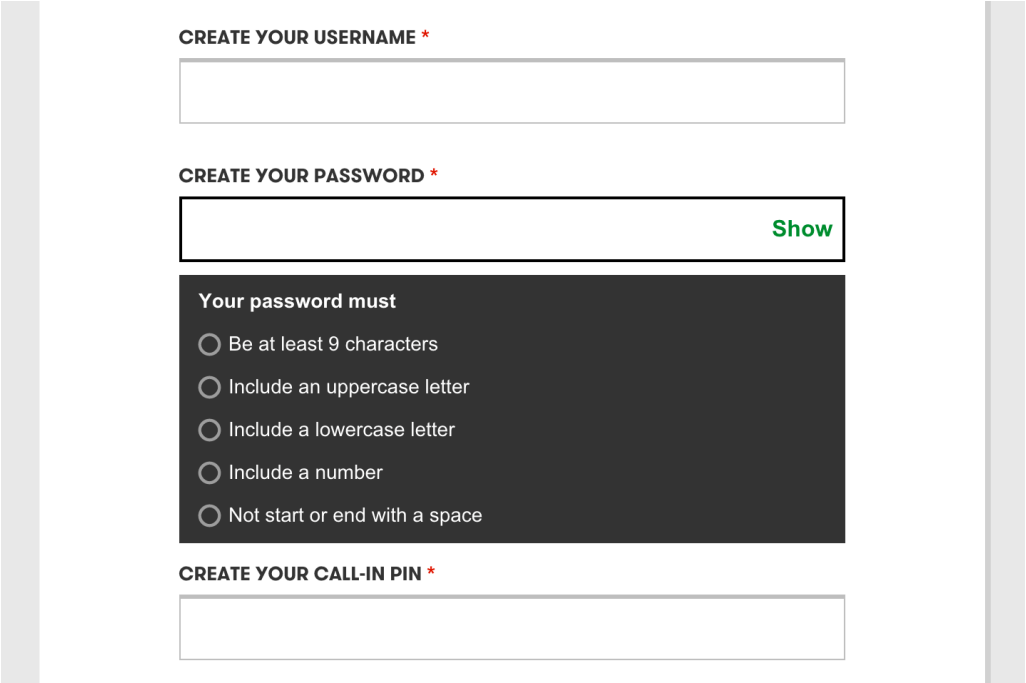
Where can you get lots of passwords to study?

- Real passwords
  - Stolen passwords
  - Surveys
  - Legitimate access to actual passwords
- Passwords created for experiments
  - Lab studies
  - Online studies

Dumb attacker	Smart attacker
aaaaaaaa	123456789
aaaaaaab	password
aaaaaaac	iloveyou
aaaaaaad	princess
aaaaaaae	12345678
…	…

Password Policies

- **Problem:** guide users into choosing strong passwords
- **Solution:** **password policies** are rules for composing passwords
  - e.g., must have at least one number and one punctuation symbol and one upper case letter

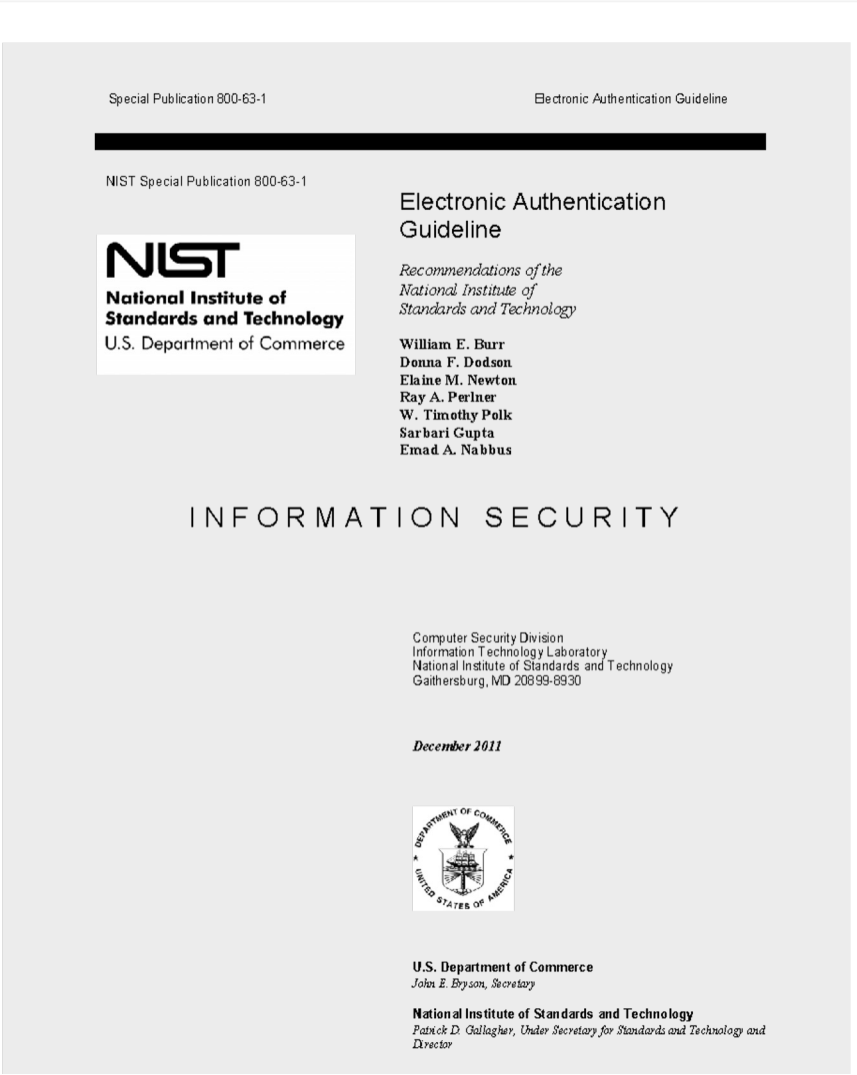


Entropy estimation

- Entropy estimates [NIST 2006 based on experiments by Shannon]:
  - (assuming English and use of 94 characters from keyboard)
  - 1st character: 4 bits
  - next 7 characters: 2 bits per character
  - characters 9..20: 1.5 bits per character
  - characters 21+: 1 bit per character
  - user forced to use lower & upper case and non-alphabetics: flat bonus of 6 bits
  - prohibition of passwords found in a 50k word dictionary: 0 to 6 bits, depending on password length

Entropy estimation

"Unfortunately, **we do not have much data** on the passwords users choose under particular rules.... NIST would like to obtain more data on the passwords users actually choose, but … system administrators are understandably reluctant to reveal password data to others."



Special Publication 800-63-1 Electronic Authentication Guideline

NIST Special Publication 800-63-1

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

Electronic Authentication Guideline

Recommendations of the National Institute of Standards and Technology

William E. Burr
Donna F. Dodson
Elaine M. Newton
Ray A. Perlner
W. Timothy Polk
Sarbari Gupta
Emad A. Nabbus

INFORMATION SECURITY

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8930

December 2011

U.S. Department of Commerce
John E. Bryson, Secretary

National Institute of Standards and Technology
Patrick D. Gallagher, Under Secretary for Standards and Technology and Director

amazonmechanical turk beta
Artificial Artificial Intelligence

Your Account | HITs | Qualifications

Already have an account?
Sign in as a Worker | Requester

Introduction | **Dashboard** | **Status** | **Account Settings**

Mechanical Turk is a marketplace for work.

We give businesses and developers access to an on-demand, scalable workforce.
Workers select from thousands of tasks and work whenever it's convenient.

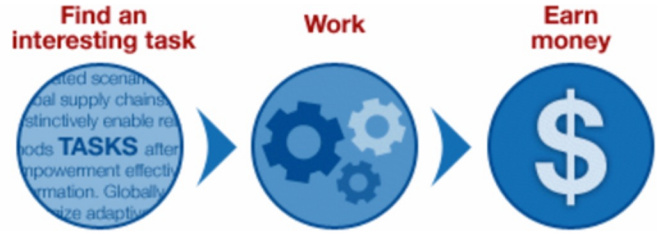
476,446 HITs available. View them now.

Make Money
by working on HITs

HITs - *Human Intelligence Tasks* - are individual tasks that you work on. Find HITs now.

As a Mechanical Turk Worker you:

- Can work from home
- Choose your own work hours
- Get paid for doing good work



Find HITs Now

or learn more about being a **Worker**

Get Results
from Mechanical Turk Workers

Ask workers to complete HITs - *Human Intelligence Tasks* - and get results using Mechanical Turk. Register Now

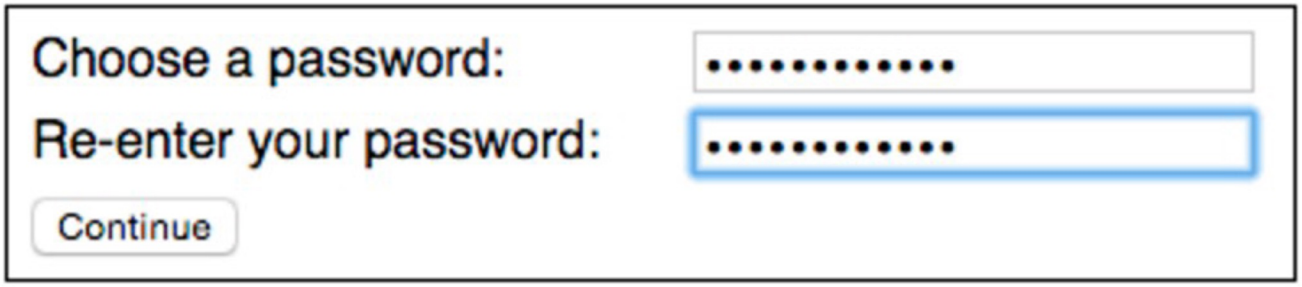
As a Mechanical Turk Requester you:

- Have access to a global, on-demand, 24 x 7 workforce
- Get thousands of HITs completed in minutes
- Pay only when you're satisfied with the results

Get Started

Participant tasks

- Create password under a randomly assigned condition
- Take a survey
- Recall password
- Return 2 days later to recall password and take survey



Password policies

Policy	Example password
Basic8	**password**
Dictionary8	**sapsword**
Comprehensive8	**Sapsword1!**
Basic16	**passwordpassword**

S. Komanduri, R. Shay, P.G. Kelley, M.L. Mazurek, L. Bauer, N. Christin, L.F. Cranor, and S. Egelman. Of passwords and people: Measuring the effect of password-composition policies. CHI 2011.

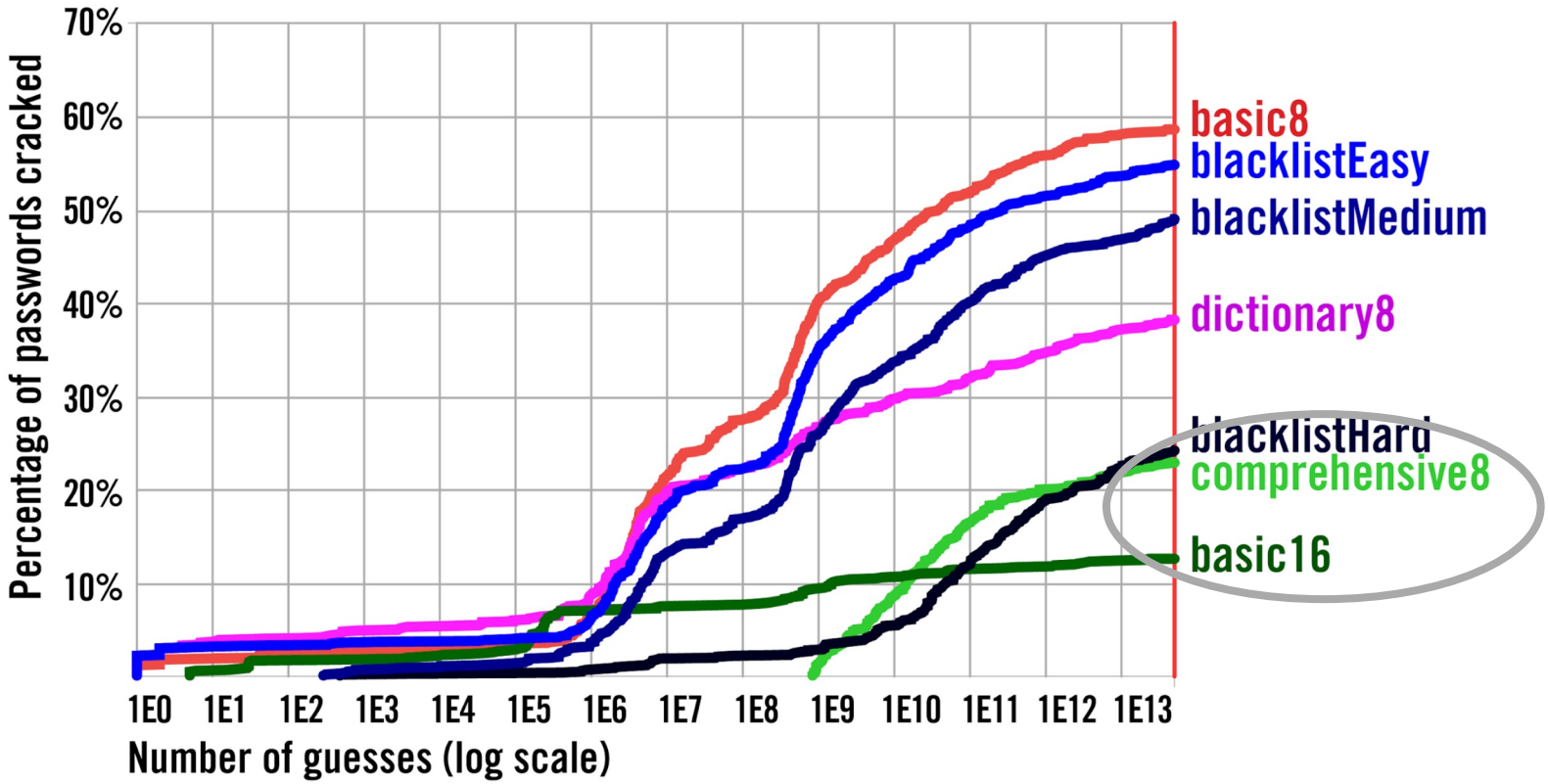
Password strength metric

Guessability

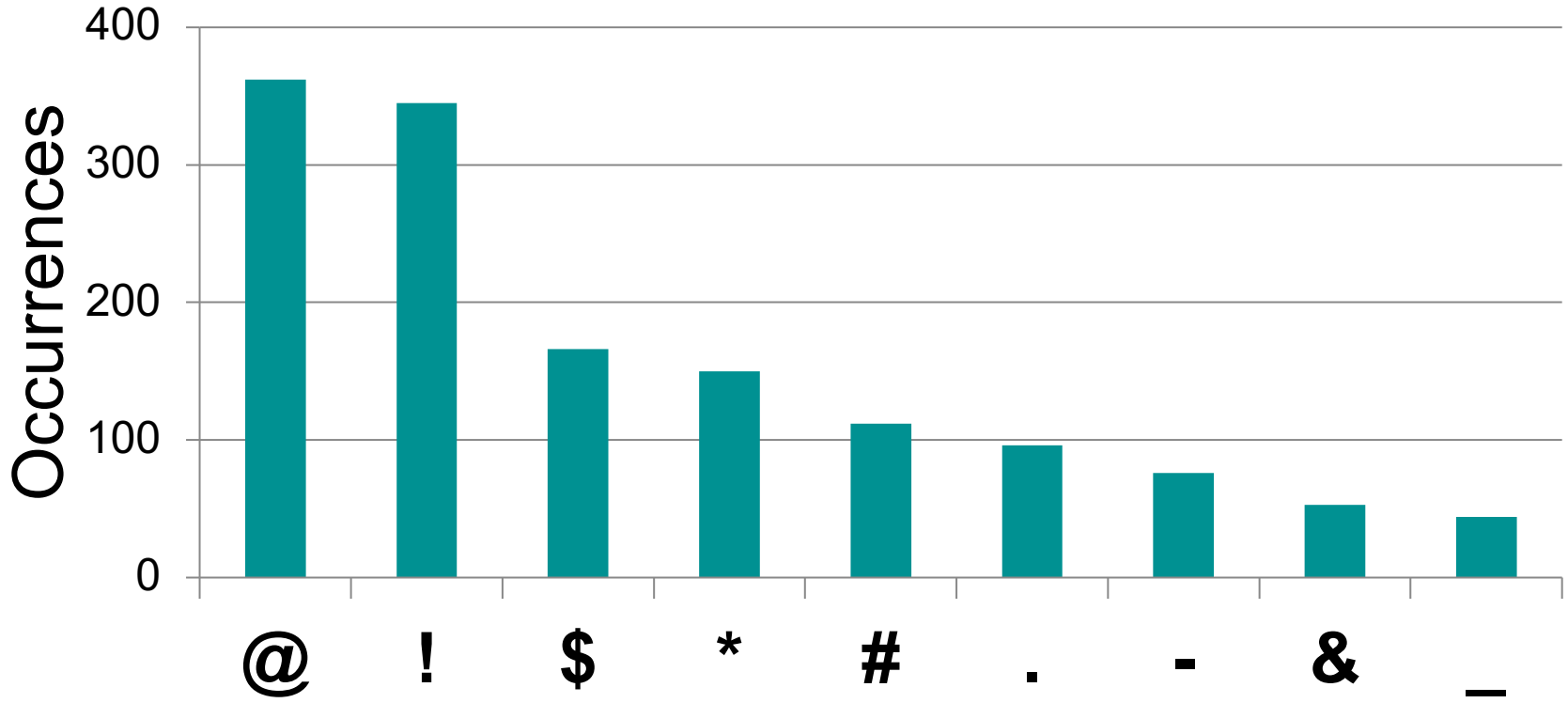
Estimate of how many guesses a sophisticated attacker will need to guess a password

Password	Guess number
12345678	4
Password178	1.4×10^6
jn%fKXsl!8@Df	Beyond cutoff

Password policy strength

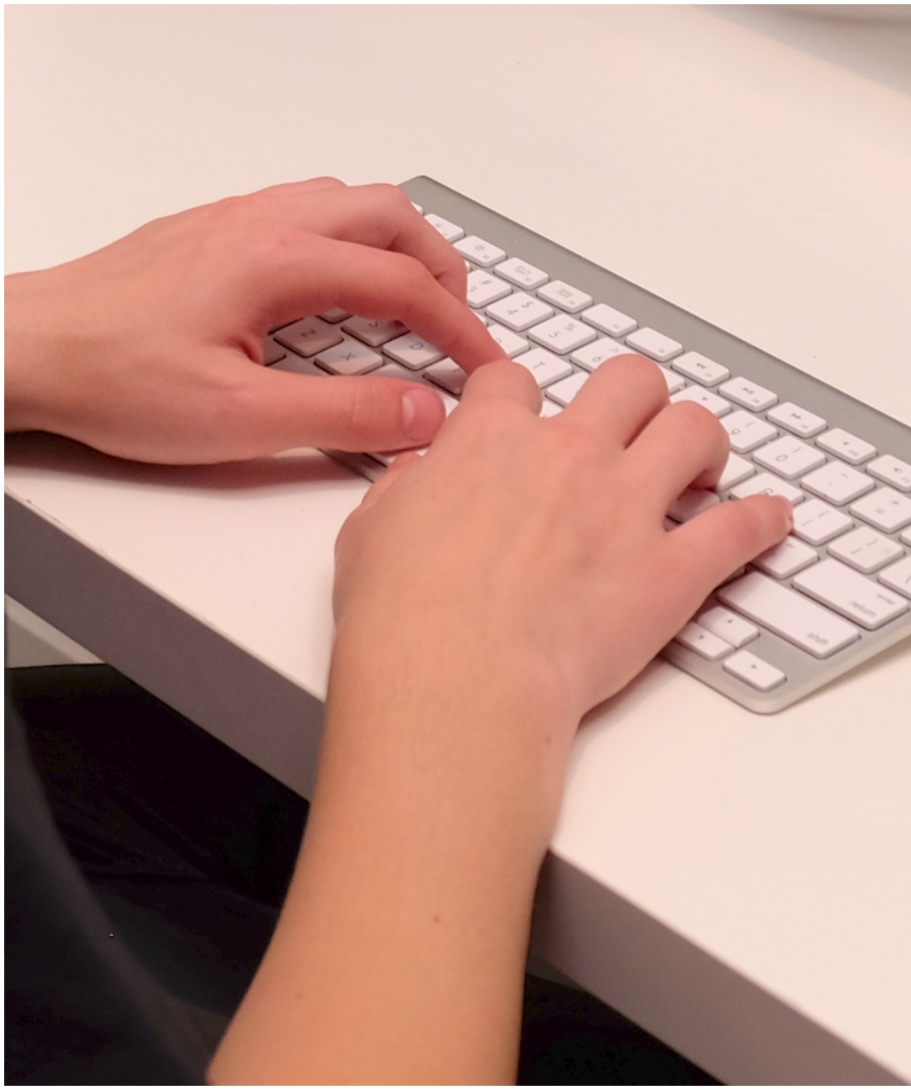


We all like to use the same symbols

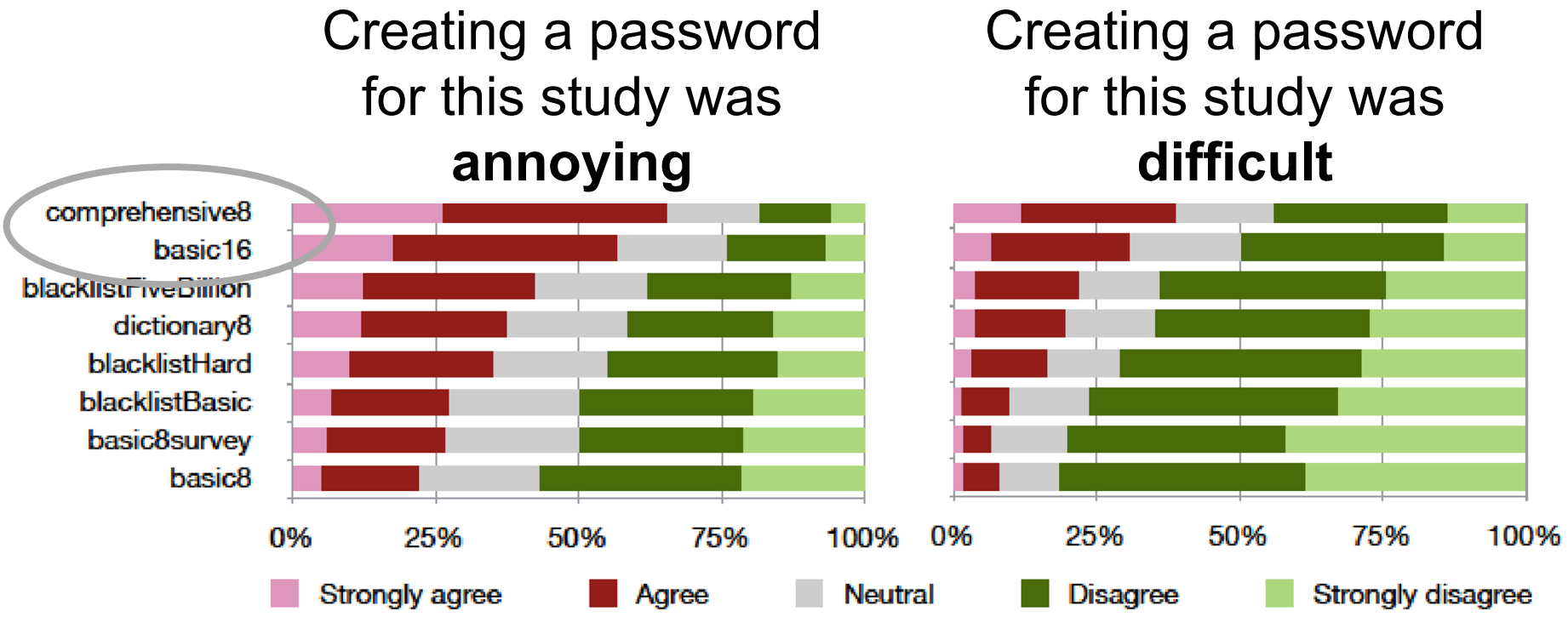


Usability metrics

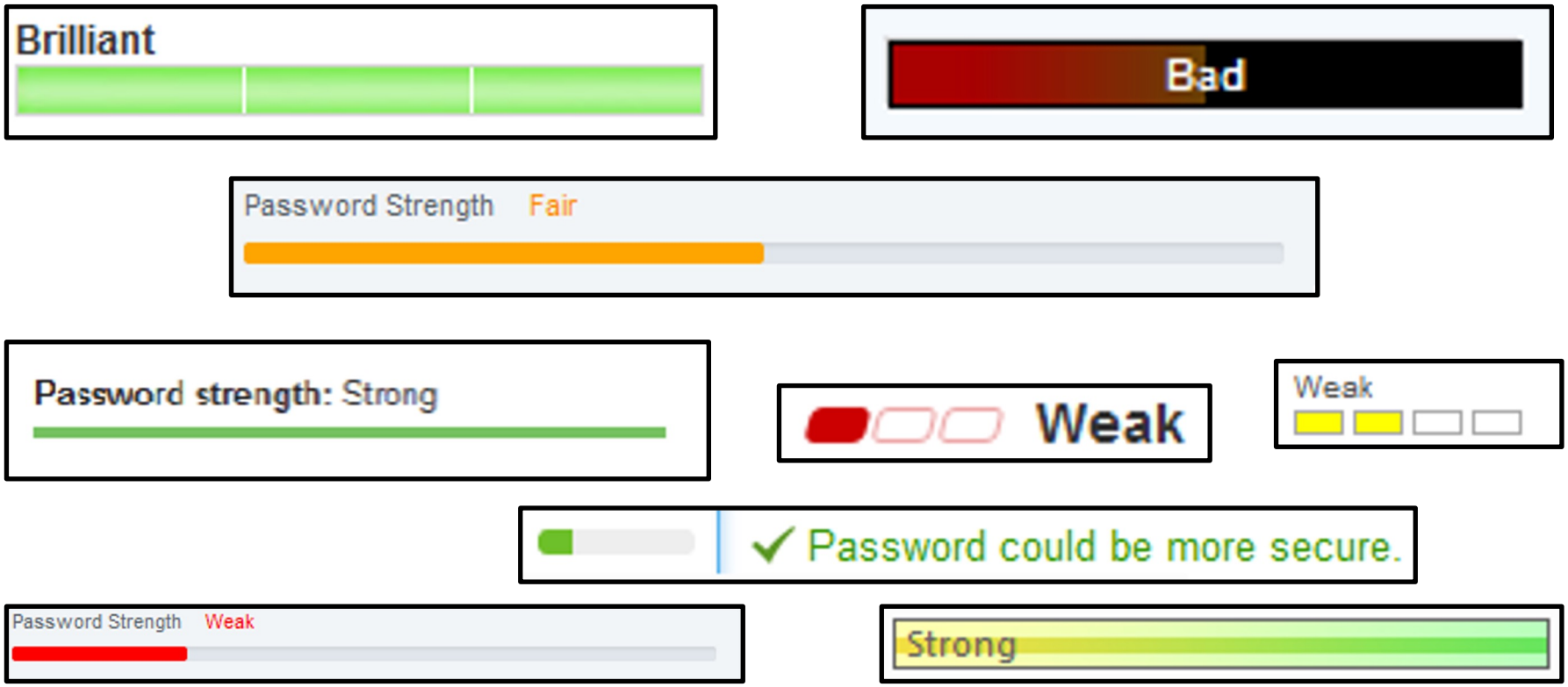
- Creation attempts and time
- Recall attempts
- Reported sentiment
- Write-down rate
- Study drop-out rate



Password policy usability



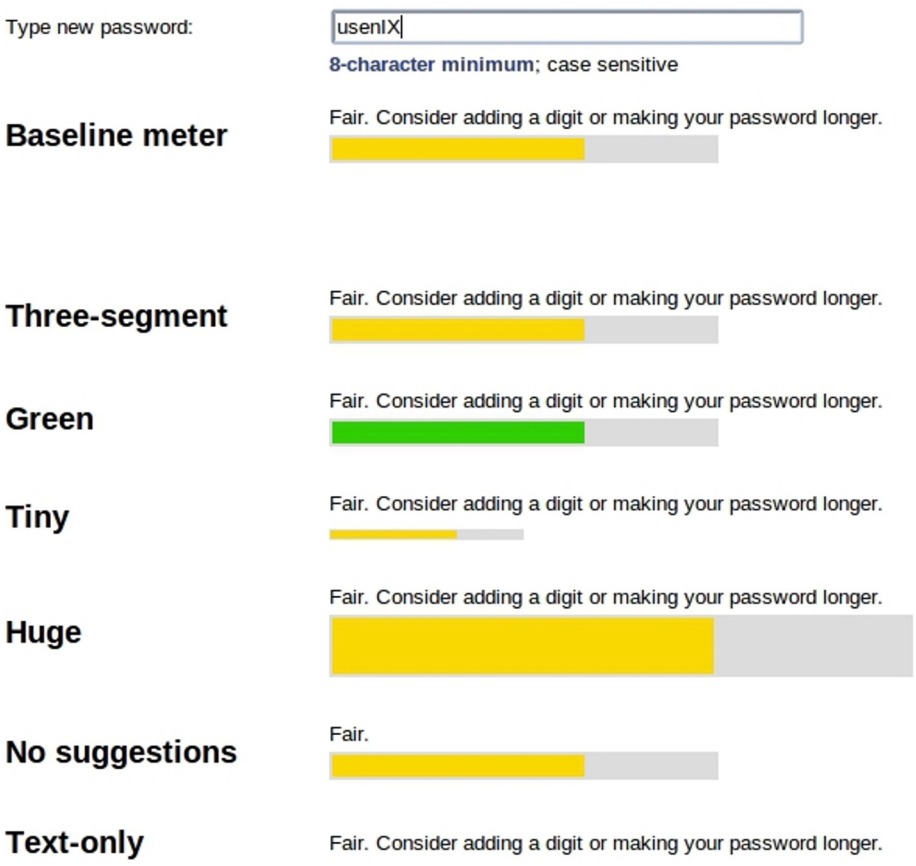
Do password meters help?



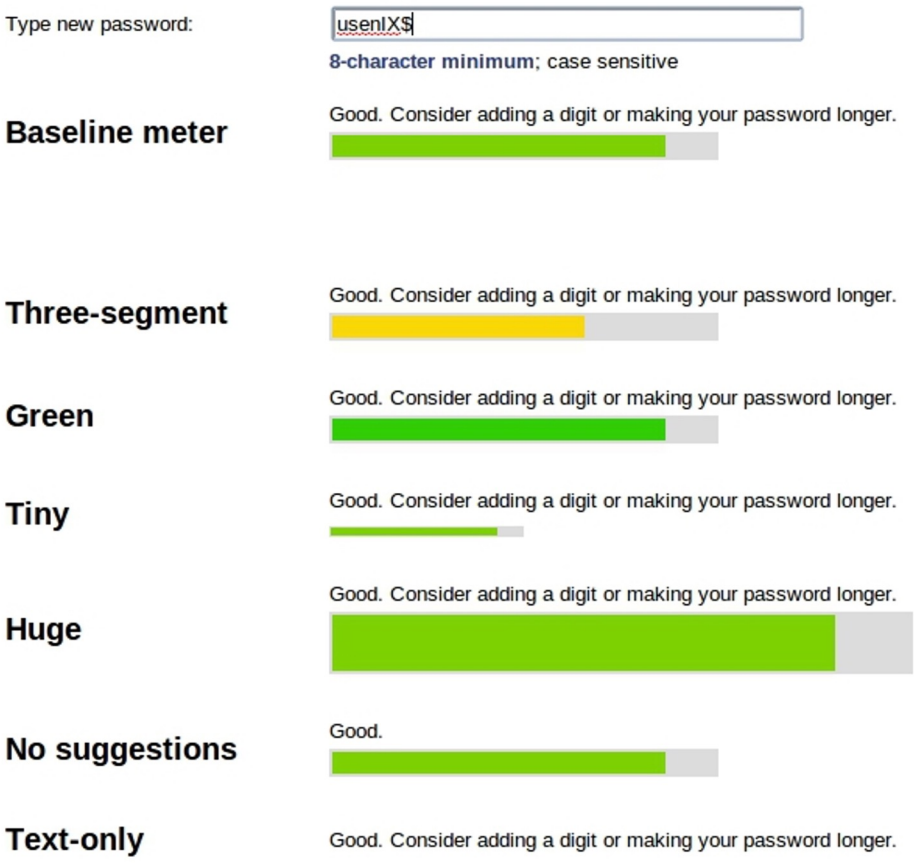
Conditions with visual differences

Conditions with visual differences

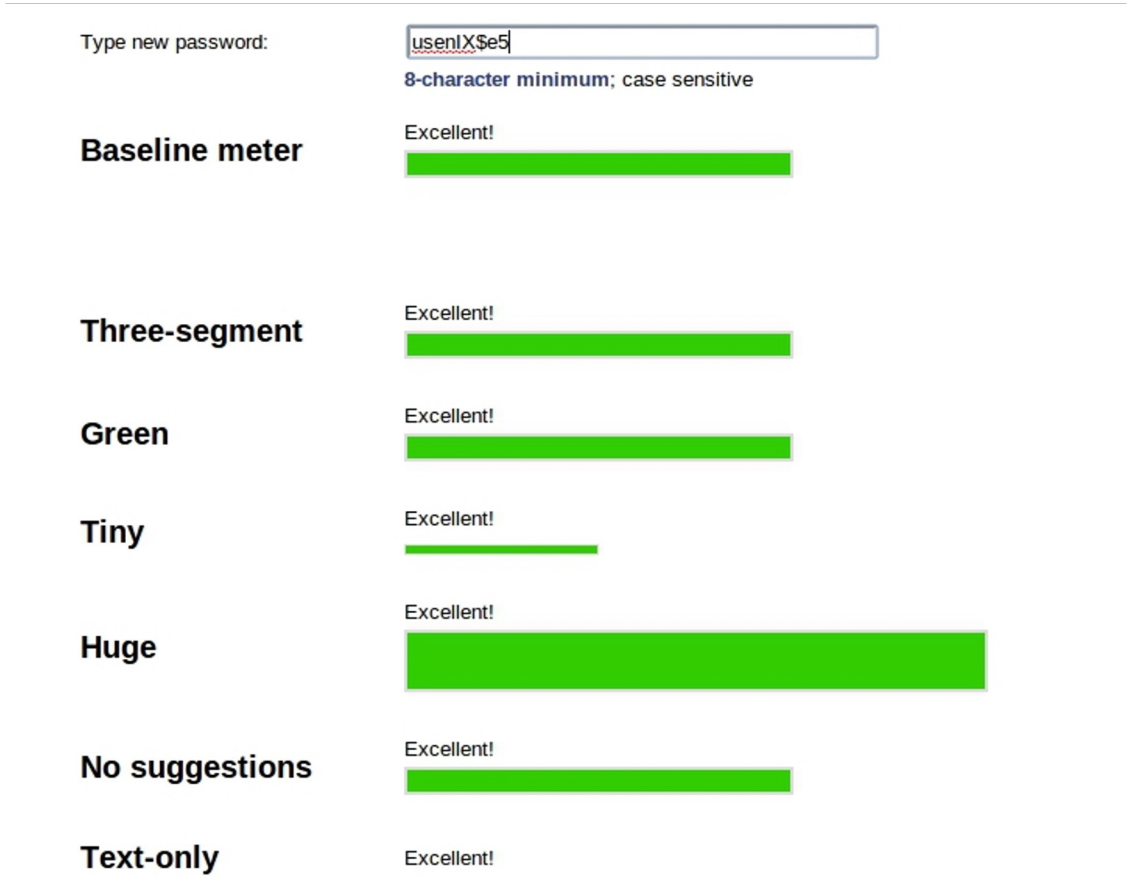
Conditions with visual differences



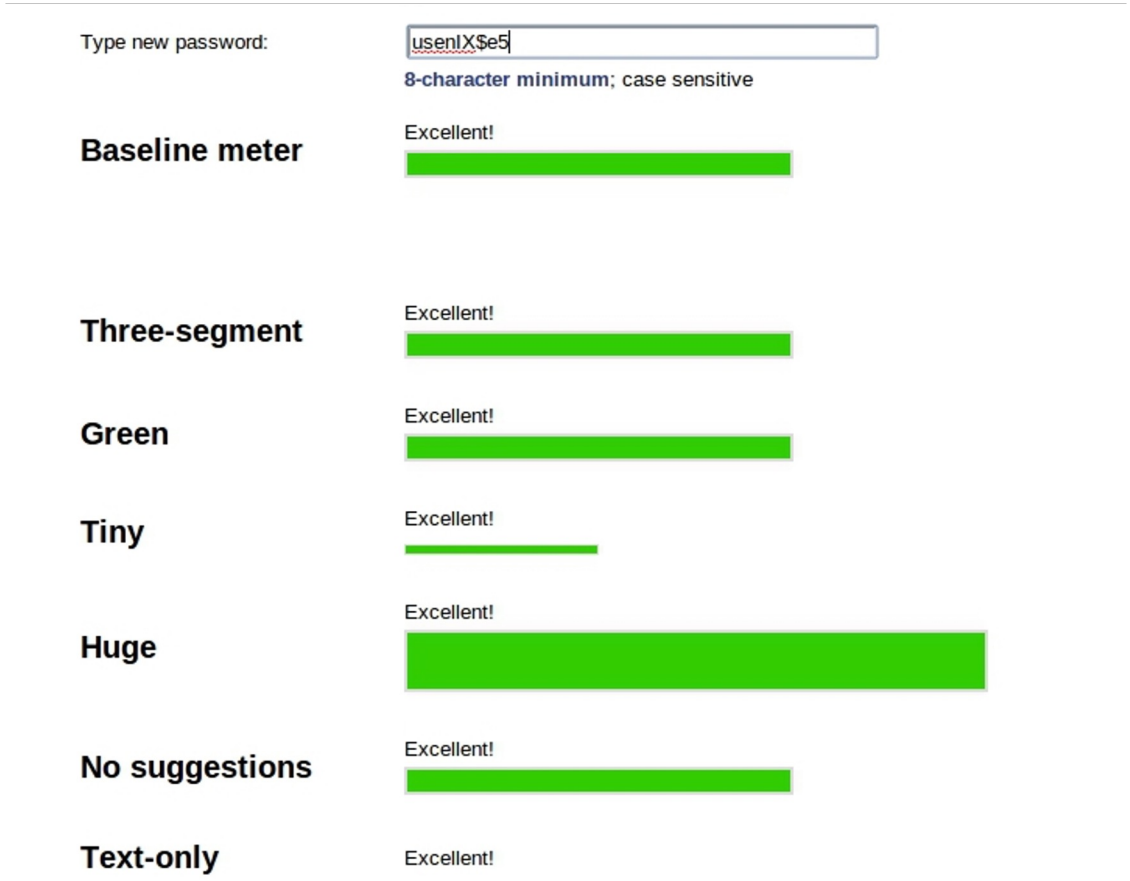
Conditions with visual differences



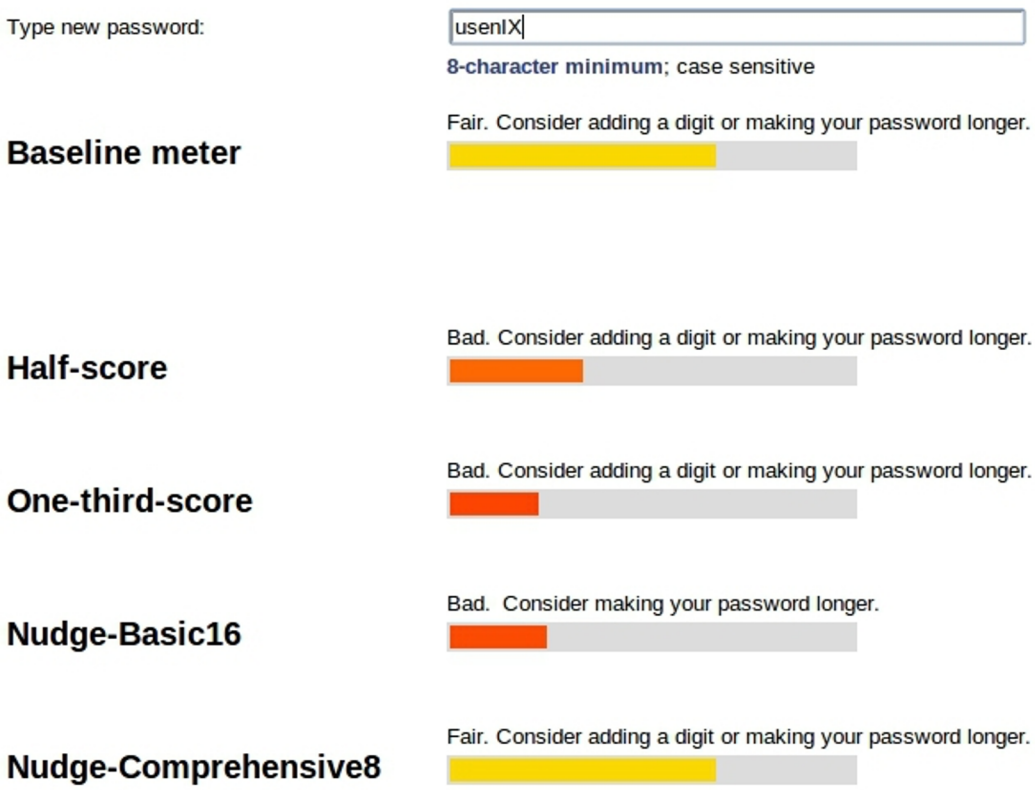
Conditions with visual differences



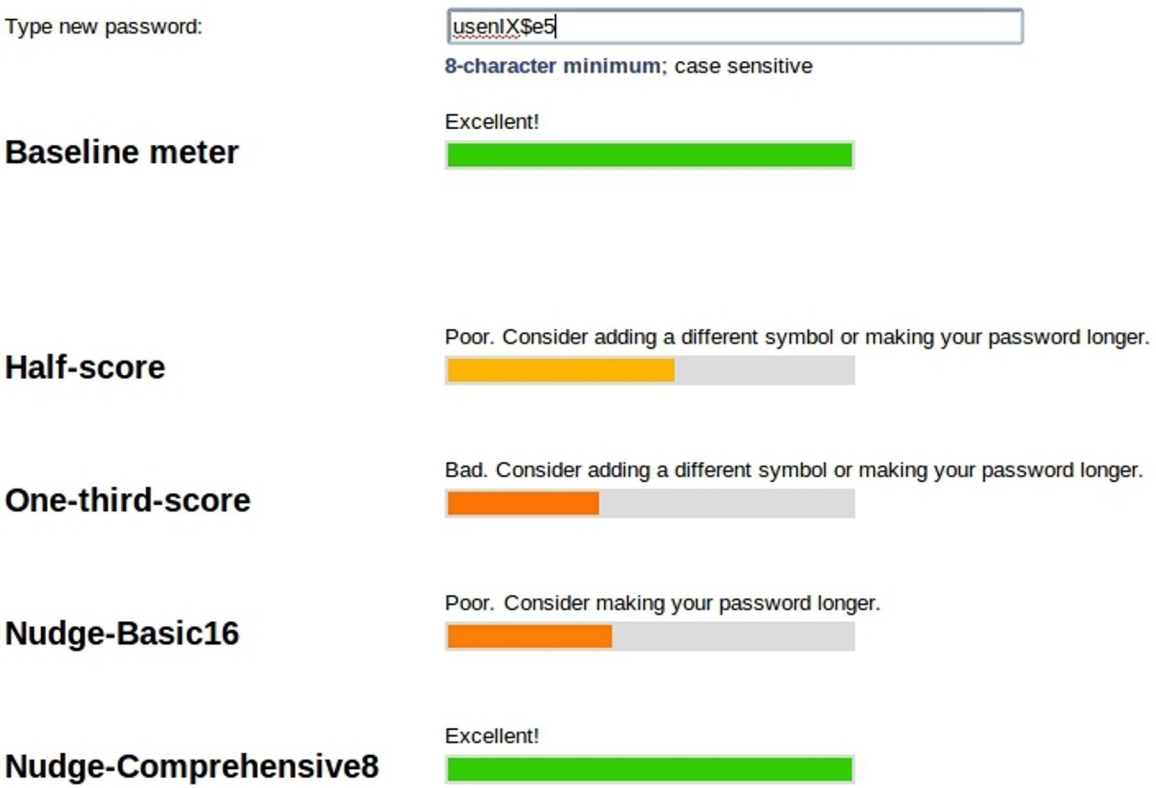
Conditions with visual differences



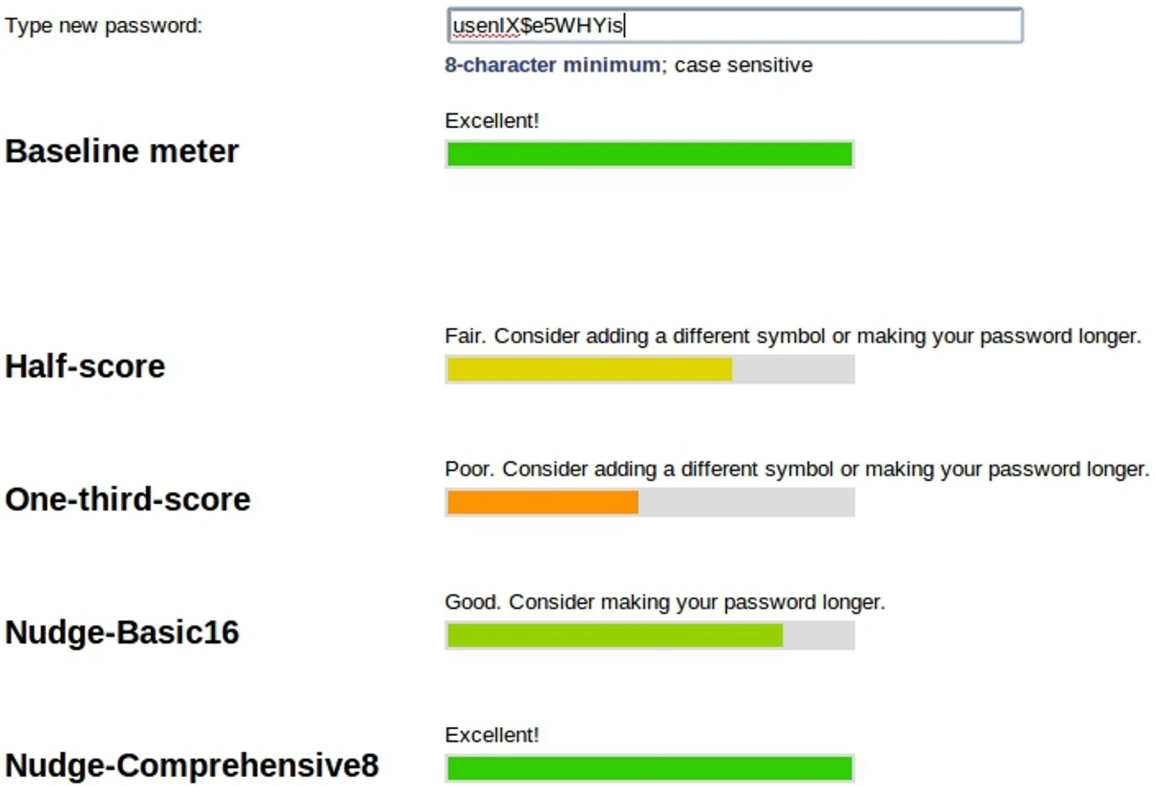
Conditions with scoring differences



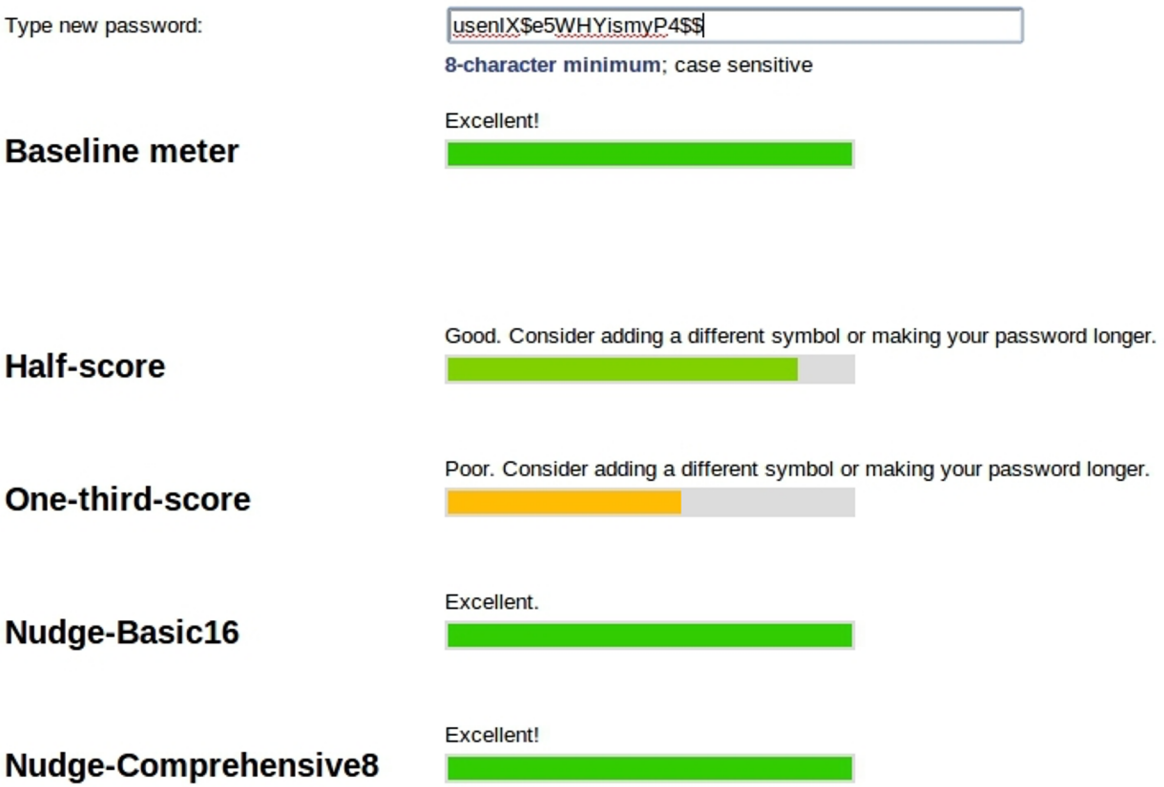
Conditions with scoring differences



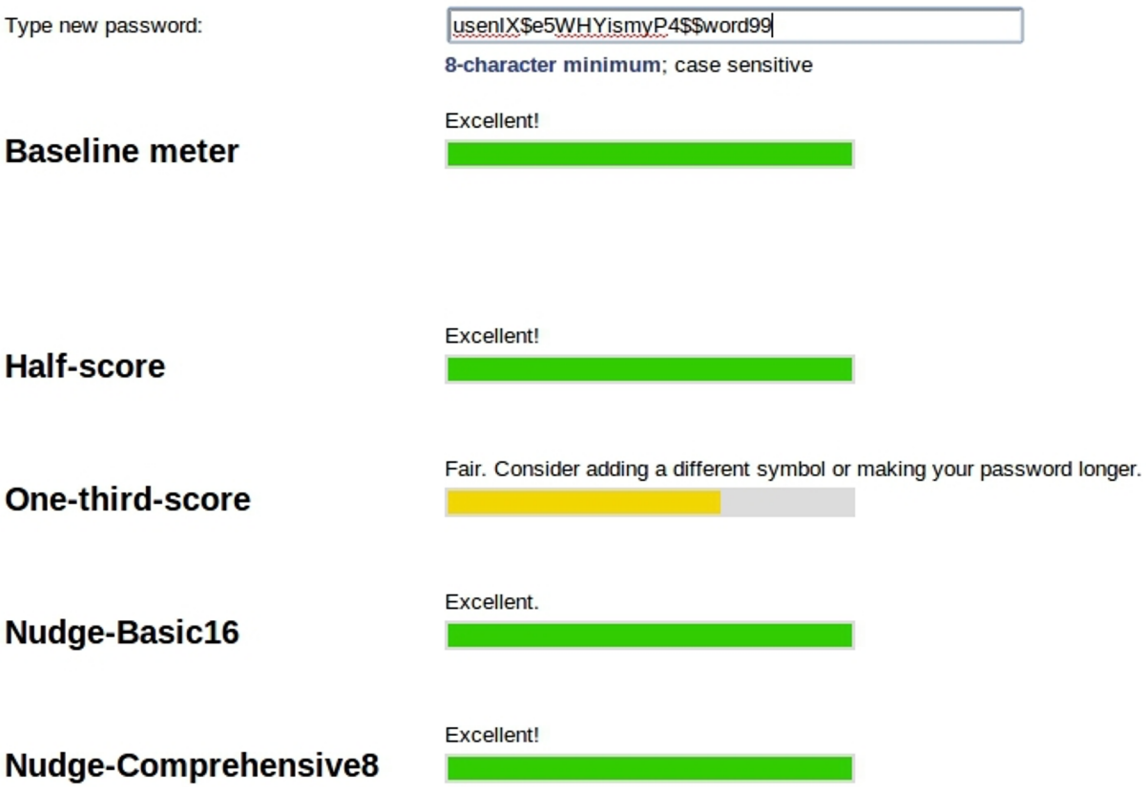
Conditions with scoring differences



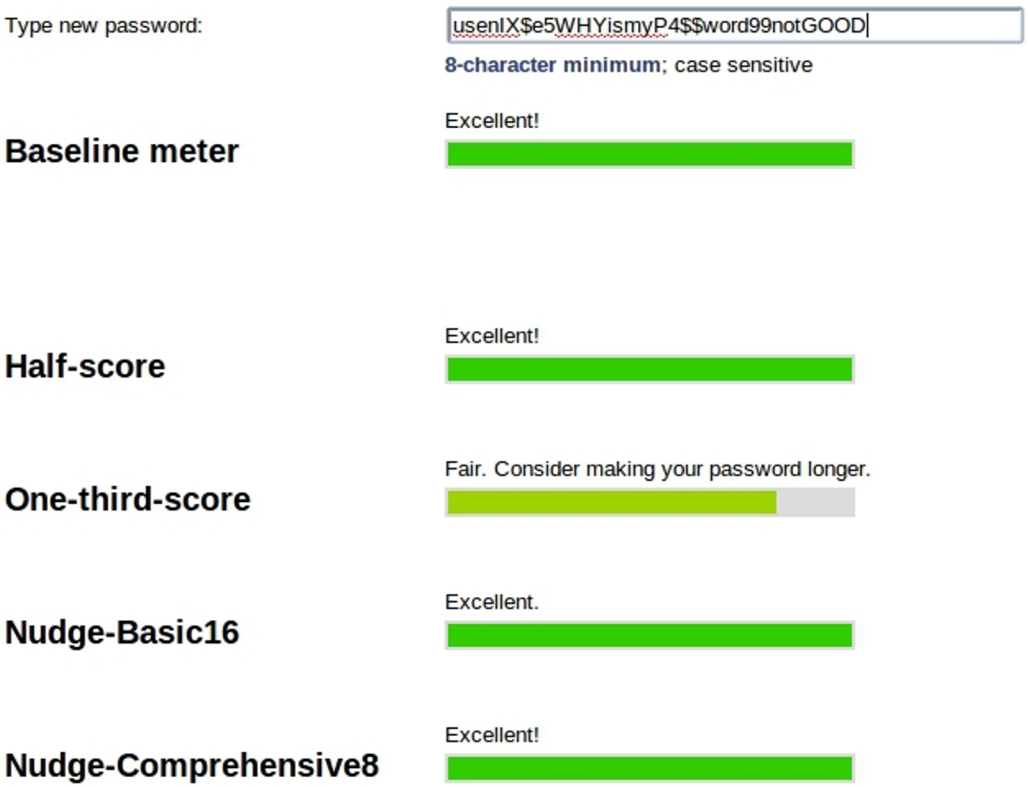
Conditions with scoring differences



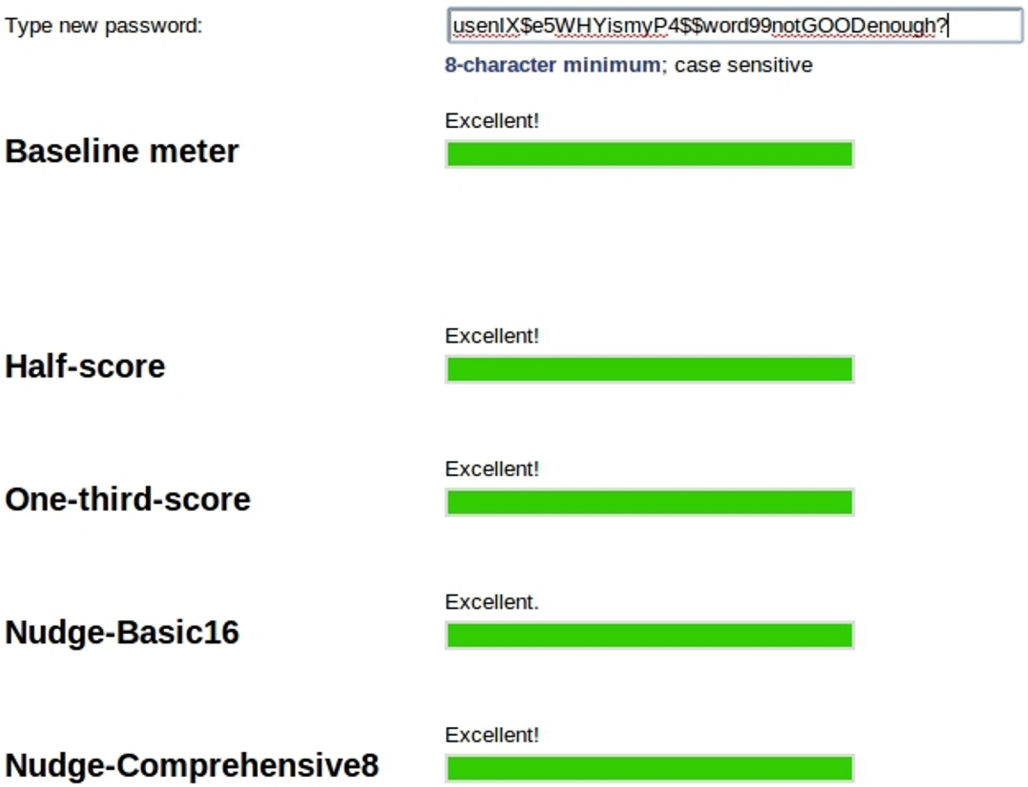
Conditions with scoring differences

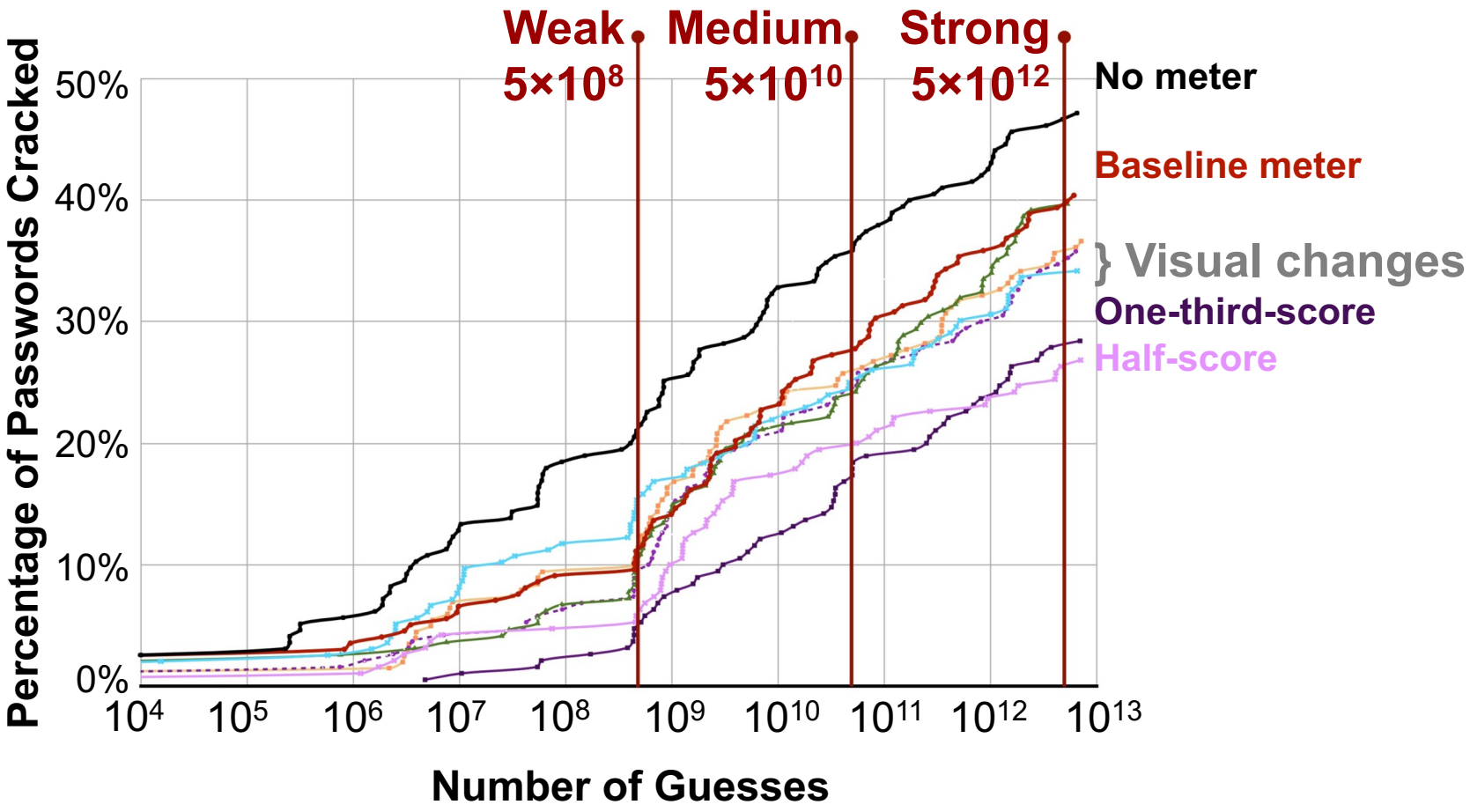


Conditions with scoring differences



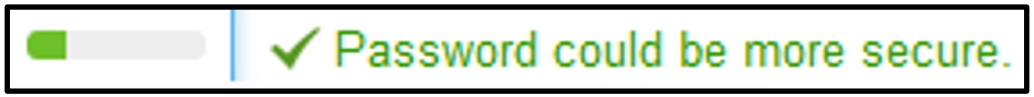
Conditions with scoring differences




Weak
5×10^8
Medium
5×10^10
Strong
5×10^12
No meter
Baseline meter
} Visual changes
One-third-score
Half-score
Percentage of Passwords Cracked
50%
40%
30%
20%
10%
0%
10^4
10^5
10^6
10^7
10^8
10^9
10^10
10^11
10^12
10^13
Number of Guesses

Meters help, but need improvement

- Color, size, shape, bunnies, don't make much difference
- Most meters on websites don't give accurate information
- Many meters provide praise too soon or don't provide actionable information



B. Ur, P.G. Kelley, S. Komanduri, J. Lee, M. Maass, M. Mazurek, T. Passaro, R. Shay, T. Vidas, L. Bauer, N. Christin, and L.F. Cranor. How does your password measure up? The effect of strength meters on password creation. USENIX Security 2012.

Passwords

NIST (2017, updated 2020) recommends:

- minimum of 8 characters
- up to 64 characters should be accepted
- all printable ASCII characters and Unicode should be accepted
- blacklist compromised values, dictionary words, repetative characters, and context-specific words
- no other security requirements

Should provide guidance on picking a good password (e.g., password meter

Exercise: Choosing Passwords

- Guess the top five most common passwords in 2021

Weak passwords

Top 10 passwords in 2021:

1. 123456
2. 123456789
3. 12345
4. qwerty
5. password
6. 12345678
7. 111111
8. 123123
9. 1234567890
10. 1234567

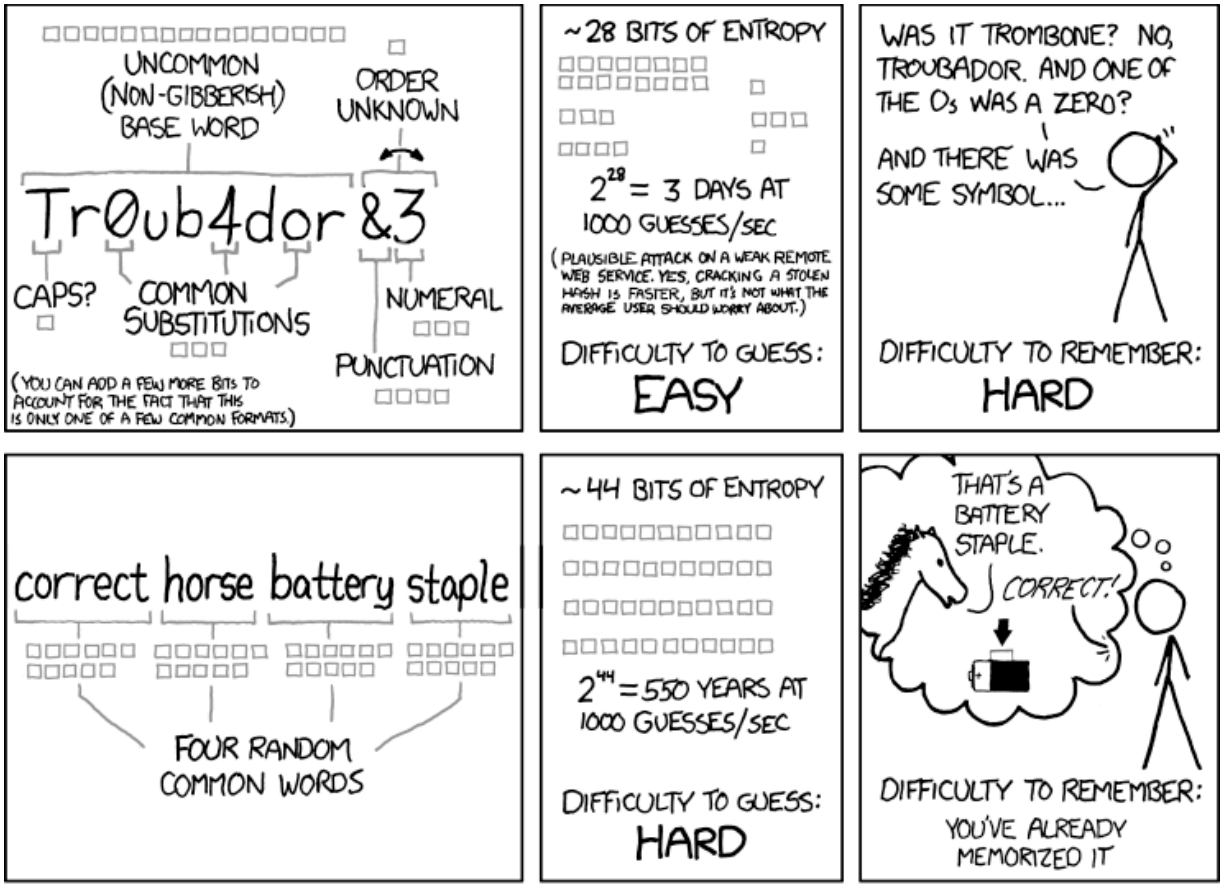
13: 1q2w3e, 31: 1qaz2wsx, 60: football

Top 20 passwords suffice to compromise 10% of accounts

Typical passwords

- 7-9 character root plus a 1-3 character appendage
  - Root typically pronounceable, though not necessarily a real word
  - Appendage is a suffix (90%) or prefix (10%)
- Dictionary of 1000 roots plus 100 suffixes (= 100k passwords) cracks about 24% of all passwords
- More sophisticated dictionaries crack about 60% of passwords within 2-4 weeks
- Given biographical data (zip code, names, etc.) and other passwords of a user...
  - success rate goes up a little
  - time goes down to days or hours

Passwords


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.