Recall

AAA[A]

Passwords

- "password" files
- dictionary attacks: online vs offline

Cryptographic hash functions

Password hashing

Resisting *offline* dictionary attacks*

Rainbows† and salt

Iterative password hashing (KDFs)

* see, e.g., John the Ripper

† Oeschslin, "Making a Faster Cryptanalytic Time-Memory Trade-Off", CRYPTO 2003: Advances in Cryptology - CRYPTO 2003, 2003. DOI: 10.1007/978-3-540-45146-4_36.

We don't need *any* cryptography to resist an online dictionary attack. Protecting password databases is, instead, all about resisting ____________________, where an adversary has gained access to a password database and they want to get passwords from it. Without any cryptography, they can simply do a database lookup. With cryptography, however, we can make things much harder for them.

As a (very bad!) alternative to password hashing, check out this analysis of a major password database breach at Adobe.

Tools like GPUs are really good at parallel computation. Attackers can use them to try lots and lots of passwords concurrently to see if they can find the correct one (a bit like the Bombes in Bletchley Park!). _______________________________ (KDFs) make life harder for an attacker by forcing computation to be _____________. There is a cost for the user, too, but it's insignificant compared to the benefit of not having your password cracked when a business suffers a data breach!

What makes a good password?

Hard to guess

Complex?

I AM MORDAC, THE PREVENTER OF INFORMATION SERVICES. I BRING NEW GUIDELINES FOR PASSWORDS.
"ALL PASSWORDS MUST BE AT LEAST SIX CHARACTERS LONG... INCLUDE NUMBERS AND LETTERS... INCLUDE A MIX OF UPPER AND LOWER CASE..."
"USE DIFFERENT PASSWORDS FOR EACH SYSTEM. CHANGE ONCE A MONTH. DO NOT WRITE ANYTHING DOWN."
SQUEAL LIKE A PIG!!!
S.Adams E-mail: SCOTTADAMS@AOL.COM
© 1998 United Feature Syndicate, Inc.
4/6/98

Fundamentally, it should be hard for an attacker to guess a password.

We have ideas about what makes guessing harder: not using common words, maybe making passwords long, maybe using funny symbols. Some of these ideas are intuitive, others have been ______________________________. But *why* would those things make it harder to guess a password?

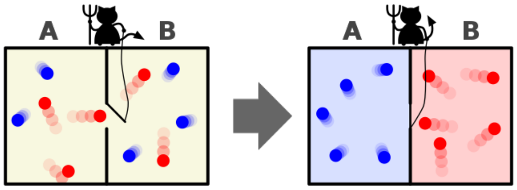
Before talking about sensible password policies, we need to understand just a little bit of information theory. One way of describing hard-to-guess-ness — but one which is often misunderstood — is the information-theoretic concept of *entropy*.

Entropy

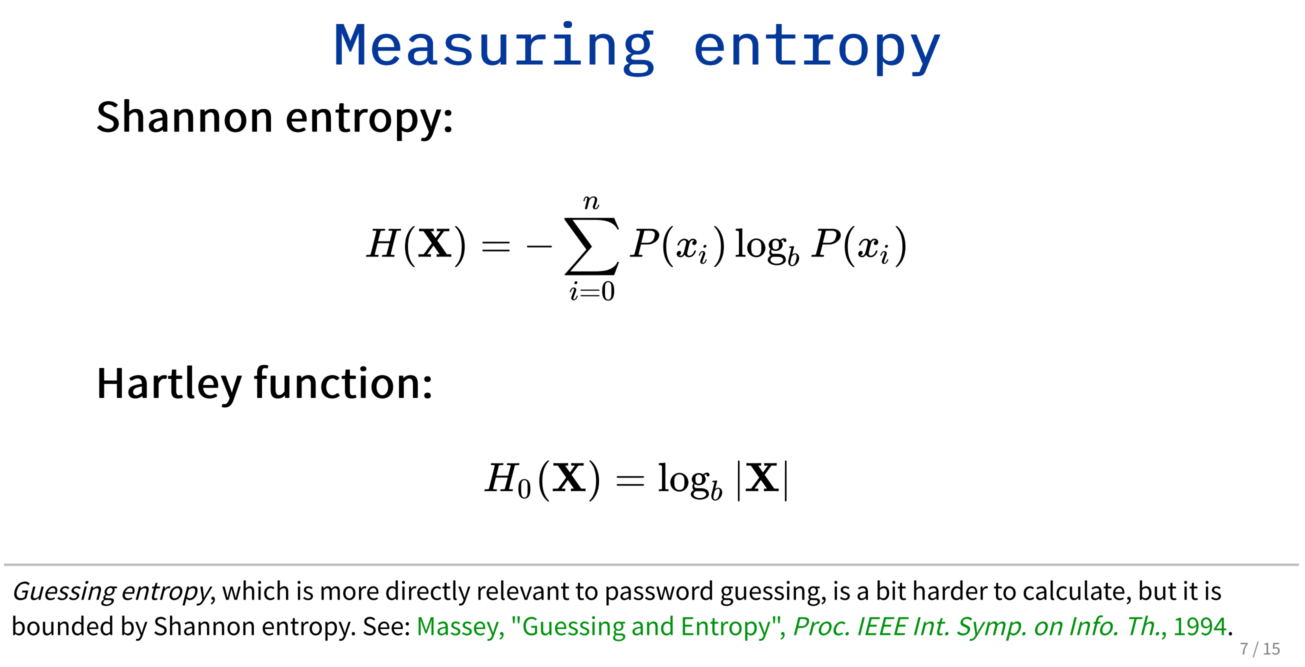
A measure of information

- or disorder, or chaos...
- thermodynamics: Maxwell's demon
- units: *Shannons* (a.k.a., bits!)

The concept of entropy is much older than computing: entropy has been used for a long time in thermodynamics to describe the disordered-ness of systems. In a closed system, entropy is a monotonically non-decreasing quantity, i.e., left alone, a closed system will become less ordered and more chaotic. The "heat death of the universe" doesn't refer to things getting really hot, it refers to an increase of random motion to the point that there is no structure, no separation between hot and cold, so no useful work can be done.



Maxwell's demon refers to a theoretical idea that links thermodynamics and information theory. Normally, if you bring hot and cold things together, their temperature evens out. However, if you could control the interface between two chambers of gas such that you open the door for faster molecules going right and slower molecules going left, but close the door for other molecules, you could make lukewarm gas turn into hot gas in one chamber and cold in the other. Would this violate the second law of thermodynamics? No, because the ________________________________ affects the entropy!



Measuring entropy

Shannon entropy:

$$H(\mathbf{X}) = -\sum_{i=0}^{n} P(x_i) \log_b P(x_i)$$

Hartley function:

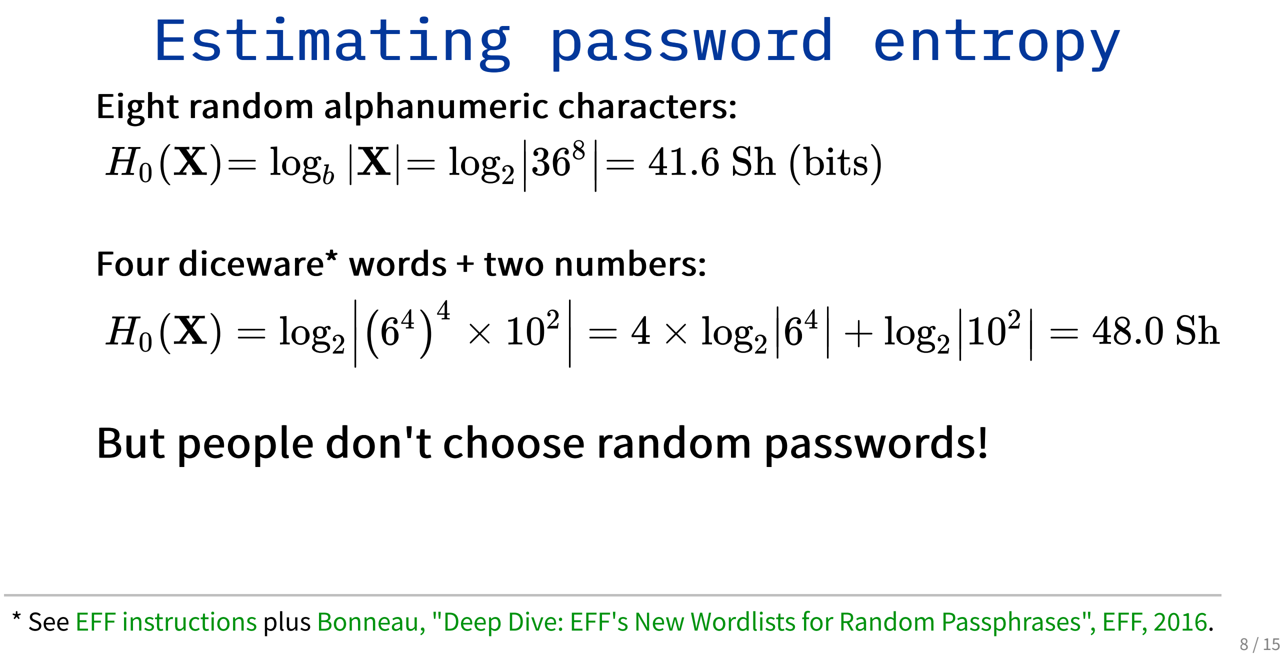
$$H_0(\mathbf{X}) = \log_b |\mathbf{X}|$$

Guessing entropy, which is more directly relevant to password guessing, is a bit harder to calculate, but it is bounded by Shannon entropy. See: Massey, "Guessing and Entropy", *Proc. IEEE Int. Symp. on Info. Th.*, 1994.

Claude Shannon defined one way of measuring entropy, based on how ________________ specific values in a distribution are. In this definition, a distribution that includes only equally-likely values will have higher entropy than a distribution that has a different distribution.

Important note: ____________________________________

The Hartley function can be used to compute Shannon entropy ________________________ __________________.



Estimating password entropy

Eight random alphanumeric characters:

$$H_0(\mathbf{X}) = \log_b|\mathbf{X}| = \log_2\left|36^8\right| = 41.6 \text{ Sh (bits)}$$

Four diceware* words + two numbers:

$$H_0(\mathbf{X}) = \log_2\left|\left(6^4\right)^4 \times 10^2\right| = 4 \times \log_2\left|6^4\right| + \log_2\left|10^2\right| = 48.0 \text{ Sh}$$

But people don't choose random passwords!

* See EFF instructions plus Bonneau, "Deep Dive: EFF's New Wordlists for Random Passphrases", EFF, 2016.

In this example, we're using 26 possible letters plus 10 digits, so a total of 36 symbols that can be used. Eight of these symbols means that there are 36^8 possible passwords that can be formed with this scheme; if they are all __________________, we can use the Hartley function to calculate the entropy of this __________________.

These calculations show how we can work with distributions that contain multiple components, e.g., multiple words or multiple classes of characters. It's worth noting, however, that the entropy of the distribution of passwords derived from just four diceware words would be:

$$H_0(\mathbf{X}) = \log_2\left|\left(6^4\right)^4\right| = 4 \times \log_2\left|6^4\right| = 41.4 \text{ Sh}$$

So, four diceware words will stand up to a brute-force attack about as well as a random eight-character alphanumeric password. But which will be easier for a human to remember?

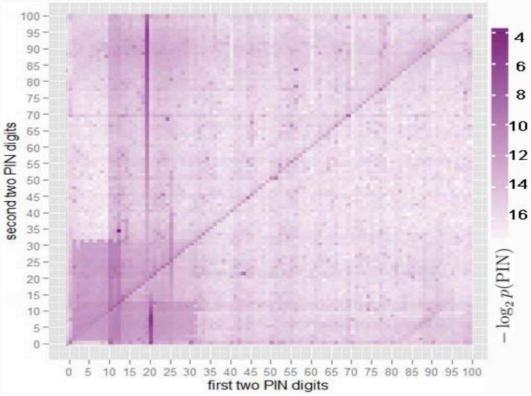
Calculating the entropy of random distributions isn't so hard. Unfortunately, however, people typically don't use randomly-generated passwords (although they should!).

Actual password entropy

People don't choose random passwords!

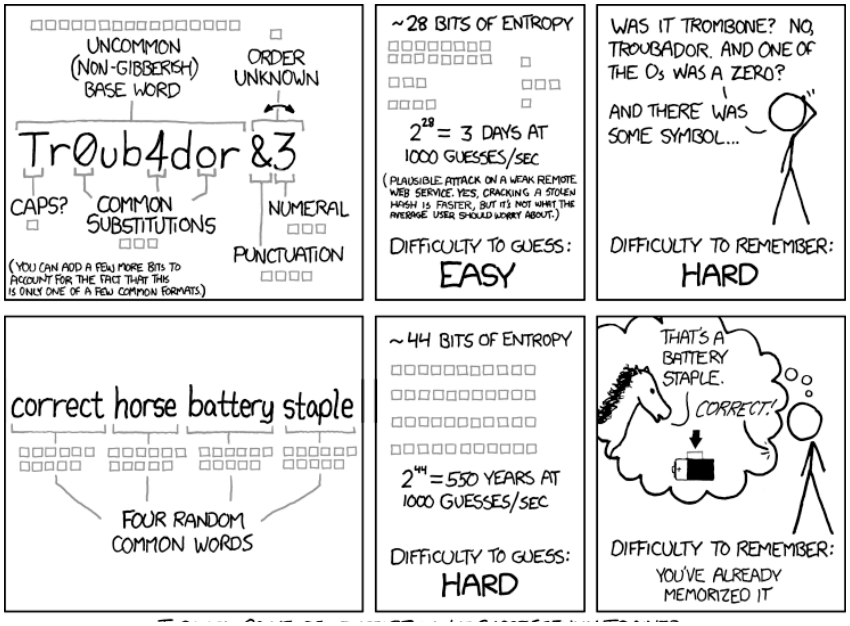
Or even random PINs!

- diagonal line: repetition
- lots of `19xx` and even `20xx`

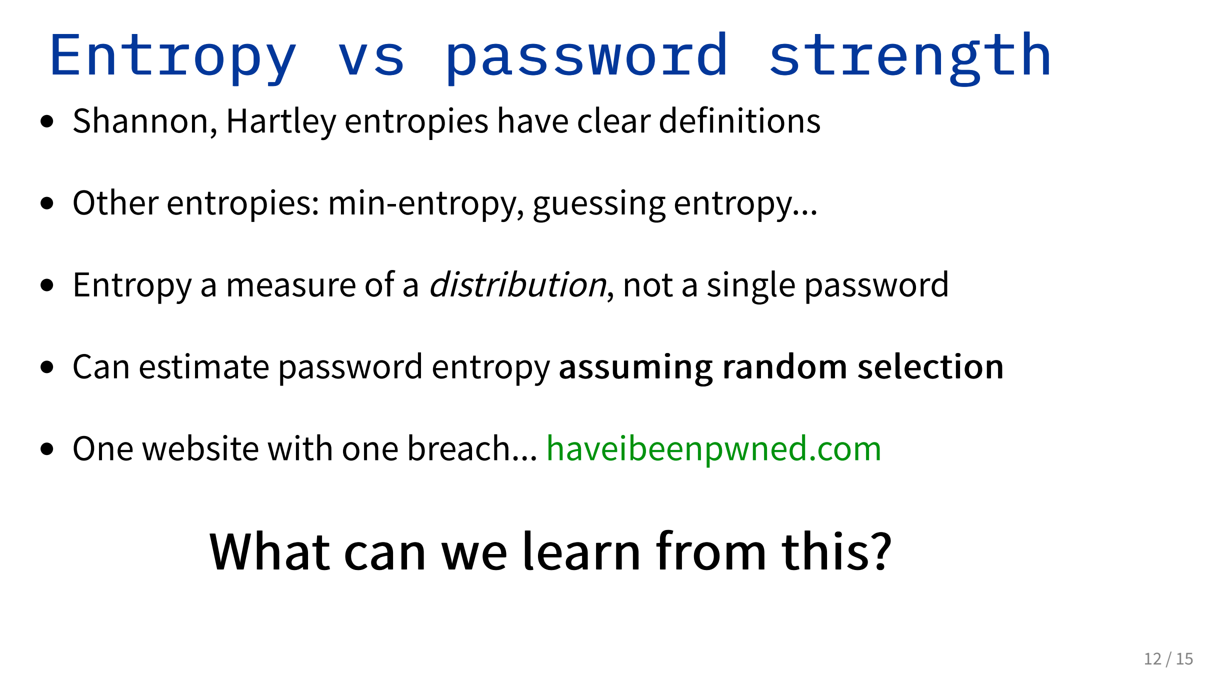


Wang et al., "Understanding Human-Chosen PINs: Characteristics, Distribution and Security", in *ASIA CCS '17*, 2017. DOI: 10.1145/3052973.3053031.

This graphic was taken from PINs that were found as a subset of the password in the original RockYou data set. For those curious about "RockYou 2021": here's a nice writeup.


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

XKCD, although it's a web comic, often has spot-on analysis. Comedy and satire can make a point in a punchier way than a prosey explanation, and this is no exception!



Entropy vs password strength

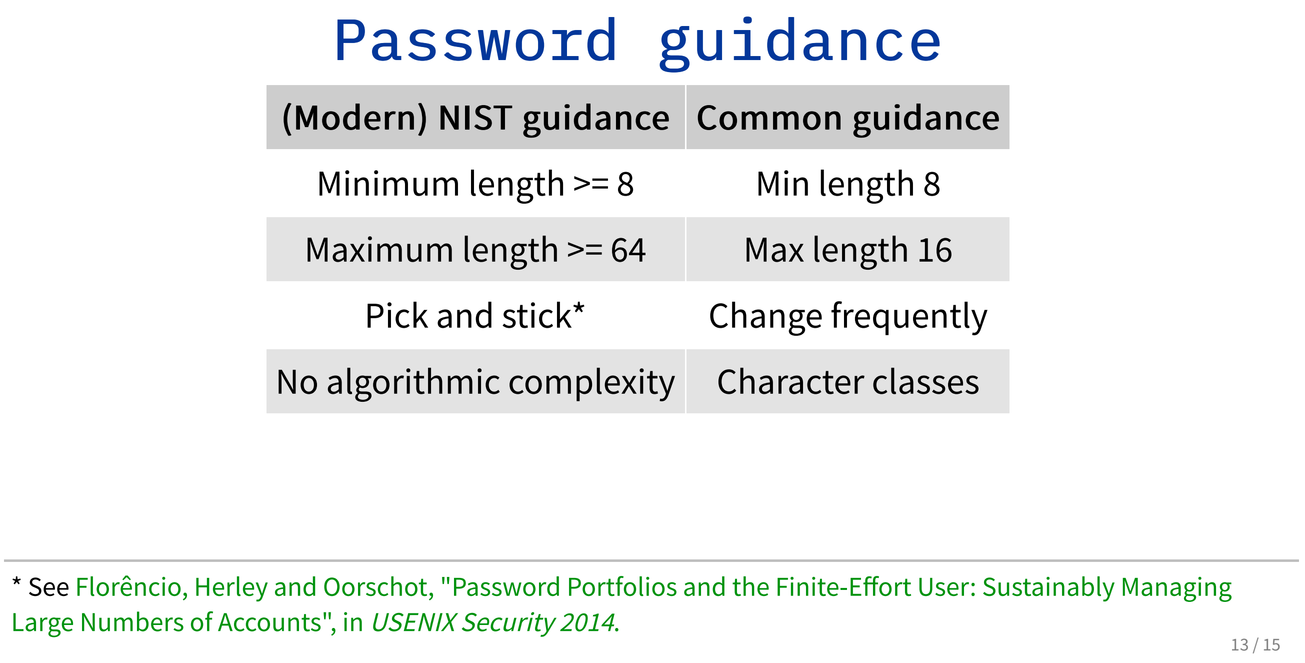
- Shannon, Hartley entropies have clear definitions
- Other entropies: min-entropy, guessing entropy...
- Entropy a measure of a *distribution*, not a single password
- Can estimate password entropy **assuming random selection**
- One website with one breach... haveibeenpwned.com

What can we learn from this?

If we assume that ____, we can compute the entropy of ____. You should note that's exactly how I've phrased one of the questions in Assignment 2!

We can learn things about actual user password choices, etc., from password databases that have been leaked. We can also learn whether or not particular passwords have been compromised!

If a password has been leaked in the clear, ____ __________! Perhaps they weren't hashing, perhaps they weren't salting, perhaps they were allowing terrible password hints, but whatever it was, somebody should lose their Internet License!



Password guidance

(Modern) NIST guidance	Common guidance
Minimum length >= 8	Min length 8
Maximum length >= 64	Max length 16
Pick and stick*	Change frequently
No algorithmic complexity	Character classes

* See Florêncio, Herley and Oorschot, "Password Portfolios and the Finite-Effort User: Sustainably Managing Large Numbers of Accounts", in *USENIX Security 2014*.

Some of these pieces of folk wisdom came from the previous NIST password guidance publication, others are accretions of myth on top of legend. The previous version of NIST's guidance was based on a very specific set of assumptions about the environment they'd be used: classified data that needs to be protected for a specific period of time (e.g., 15 years) against a dedicated cracker _______________ ___ _______________________________________ _____________. The new guidance reflects reality for more general computer security uses.

More password guidance

(Modern) NIST guidance	Common guidance
No hints	Allow "rhymes with assword"
No KBA	Non-password passwords
Screen for compromises	???
Careful about 2FA	SMS FTW

More password guidance

(Modern) NIST guidance	Common guidance
No hints	Allow "rhymes with assword"
No KBA	Non-password passwords
Screen for compromises	???
Careful about 2FA	SMS FTW

... which will be the topic of our next lecture (2FA)