Hashing

Topics covered today

- Homework and review
- What is hashing
- Class exercise

Homework review

Did anyone do any further digging?

Are there any questions over this?

What questions would you NOT want me to ask on a quiz?

What is Hashing?

You guys tell me:

Wikipedia

> *A cryptographic hash function is a one way mathematical function that maps data of an arbitrary size to a bit array of fixed size. The output bit array is commonly called a hash or digest.*

Key tenets of a hash algorithm:

1. ***Fast***, must not take a lot of resources or time to compute
2. ***Irreversible***, must not be able to retrieve the original data from the resulting hash (one way transformation)
3. ***Deterministic***, for the same input value, a hash function must always provide the same output
4. A small change to the input should result in large change to the hash value (***avalanche effect***)

What is Hashing used for:

- verify the integrity of data (block chain, downloads, git commits, secure software)
- digital signatures
- as a part of verifying a given users *Authenticity* (a part of *Authentication*)
- Proof-of-work (mining, defense against Denial of Service)

Class Exercise

Bogus quiz results are in!

But first, a pet peeve of mine. . .

Link to quiz result data `sha256sum` :
`ec39ca84b1ddc4608a161ba8338846b51e5de14cc7f7b8057c61d7b3db8`

do you trust the above

Follow along (in `bash`) for some data science fun:

What is the difference between the following:

- `echo "Hi!"`
- `printf "Hi!"'` <—— Hint: Use this one for the exercise

What do the following do?

- `|` The pipe character
- `awk -F ',' '{ print $1 }'` Hint: pipe the `cat` output of quiz-data.csv into this
- `grep <string>`

Now about the quiz data, can we reverse our hash?

What are some common (semi) unique identifiers for people?

- campus W-number
- campus UID
- Social Security Number
- First initial Last name combo ie. `MKijowski`

There is a quiz covering today's in class exercise and the homework!

Quiz is available till 11:59pm tonight! One chance only this time! Take the quiz! Note: your grade might be low on this one, I may have broken the auto-grader...

Ugh

Day 2 Hashing

Anatomy of a git commit

Nonce versus salt

Apparently we did not cover nonces in homework
Both have the same property, make it harder to brute force a hash by requiring much more computational power to create larger rainbow tables.
This does not increase any guarantee of integrity!!
Salts are intended to be public values that are stored alongside the hash. They are used each time the hash is authenticated against.
Nonces (numbers used only once) **can** be public but are not necessary to keep. They are used only once to hash some data then not used again, even when hashing the same data.
Lets do a real world test!
Linux password hashes are stored in `/etc/shadow`. Lets make a new user and see if we can verify their hash!