SSH Keys

- To login to a remote server like Perun or Graham, you would typically open your terminal and type

  ```
  $ ssh -XY <user>@<HostName>
  ```

  And then proceed to type in your password
  Altogether A LOT of keystrokes

- It is however much much more convenient and safer to switch from a password authentication system to an **SSH Key based authentication system**

ACCOUNT CREATION

PASSWORD AUTHENTICATION

PASSWORD AUTHENTICATION

LET'S SEE ..

ARMISURPLUS = SIGNUM
VITRIOLIX = DIGNUSEST
GIVUSABONUS = SESTERTIUS
ASTERISCUS = AVECAESAR
GLUTEUSMAXIMUS = OLYMPUS

PASSWORD
AUTHENTICATION
PASSED

A HACKER LISTENING IN

AHA!

ACC: ASTERISCUS
PWD: AVECAESAR

I AM ASTERISCUS

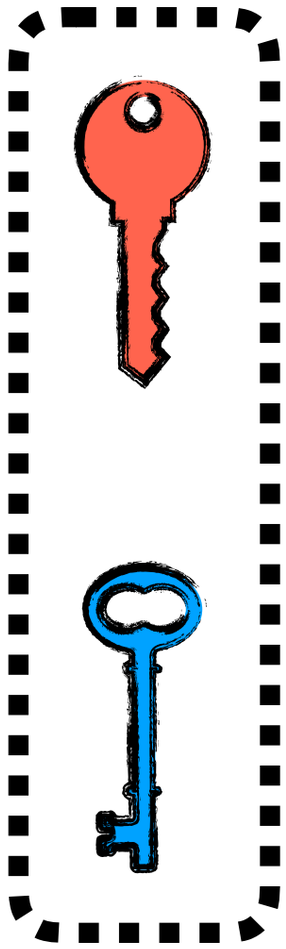
PASSWORD IS 'AVE CAESAR'

WHO ARE YOU?

WHAT'S THE PASSWORD?

SSH KEY AUTHENTICATION

A public private key pair



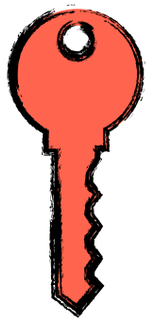
A **public key *encrypts*** data
A public key can be freely distributed to anyone

A **private key *decrypts*** data
It can only decrypt data that was encrypted with the associated public key
A private key must NEVER be shared with anyone!

SSH KEY
KEY PAIR CREATION AND PUBLIC KEY TRANSFER

Create
key pair

Copy **public key**
and transfer



Authenticate
with your
password

SSH KEY AUTHENTICATION

ASTERISCUS' PRIVATE KEY

ASTERISCUS' PUBLIC KEY

I AM ASTERISCUS

7G1O3L...

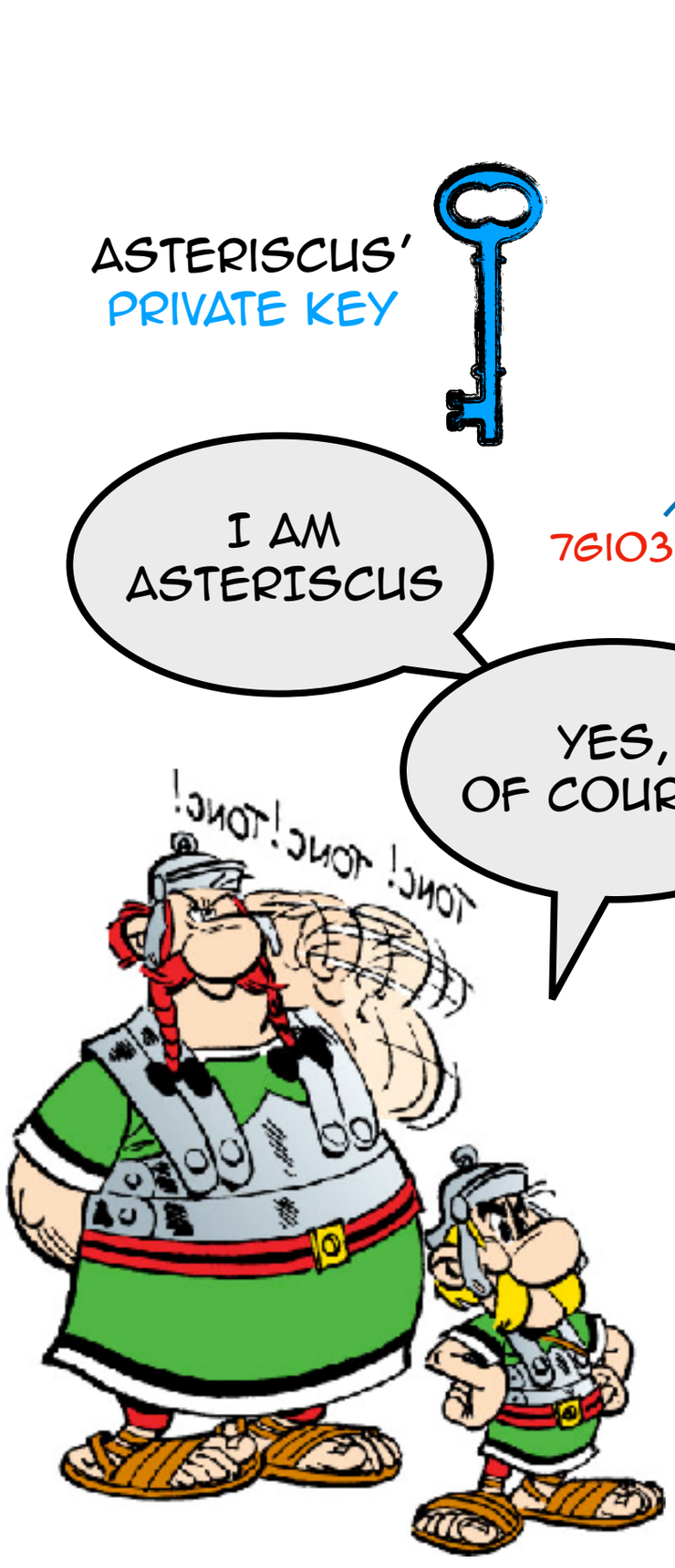
39485

YES, OF COURSE!

WHO ARE YOU?

7G1O3L...

39485



CAN YOU DECRYPT THIS RANDOM ENCRYPTED NUMBER?

SSH KEY AUTHENTICATION

AHA!

KEY ID:
ASTERISCUS

ENCR #:
7GIO3L7X987CJ
2457CBJNDVMC
DFG87KJV82

I AM ASTERISCUS

YES, OF COURSE!

WHO ARE YOU?

CAN YOU DECRYPT THIS RANDOM ENCRYPTED NUMBER?

SSH KEY AUTHENTICATION PASSED

- **No more password** required

- **Safer** because an external listener will
  (a) have an EXTREMELY hard time to decrypt the number without the private key
  (b) the number used is different each time you log in

- **RISK:** if an attacker gains access to your system in any way, they can enter any remote that you have an SSH Key set up for without having to enter a password

  You can offset that risk by **setting a passphrase**, but this means you have to type the passphrase each time you login. Setting a passphrase encrypts your private key file

- The SSH Key + passphrase combination is the safest option, by far

  When connecting to computer clusters, you should probably set up an SSH Key + passphrase authentication !!!

CREATING SSH KEY PAIRS

1. `$ ssh-keygen -t rsa`

2. Save private key as `~/.ssh/<id>_rsa`
   Public key auto saved as
   `~/.ssh/<id>_rsa.pub`

3. Optionally choose a passphrase

4. `$ ssh-copy-id -i ~/.ssh/<id>_rsa.pub <user>@<HostName>`

5. Enter regular password

6. The public key is now stored in `~/.ssh/authorized_keys` server side

You can now login without typing the password with

```
$ ssh -i ~/.ssh/<id>_rsa <user>@<HostName>
```

ADDING A PASSPHRASE TO A PRE-EXISTING SSH KEY PAIR

```
$ ssh -p -f ~/.ssh/<id>_rsa
```

CHOOSING A PASSWORD OR PASSPHRASE

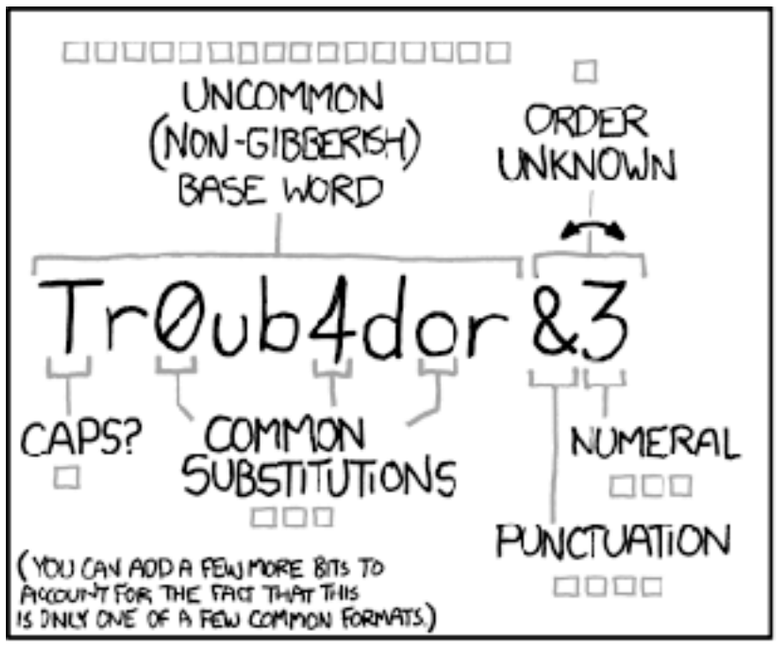
- **RULE #1**
  Do NOT use 'password', '1234', 'qwerty', 'letmein' or something like that

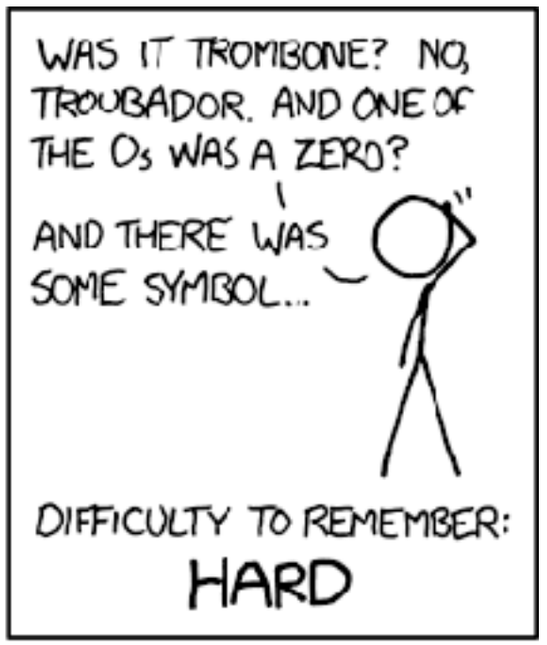
- **RULE #2**
  Do NOT use a password < 8 characters, a hacker can *bruteforce* it!!

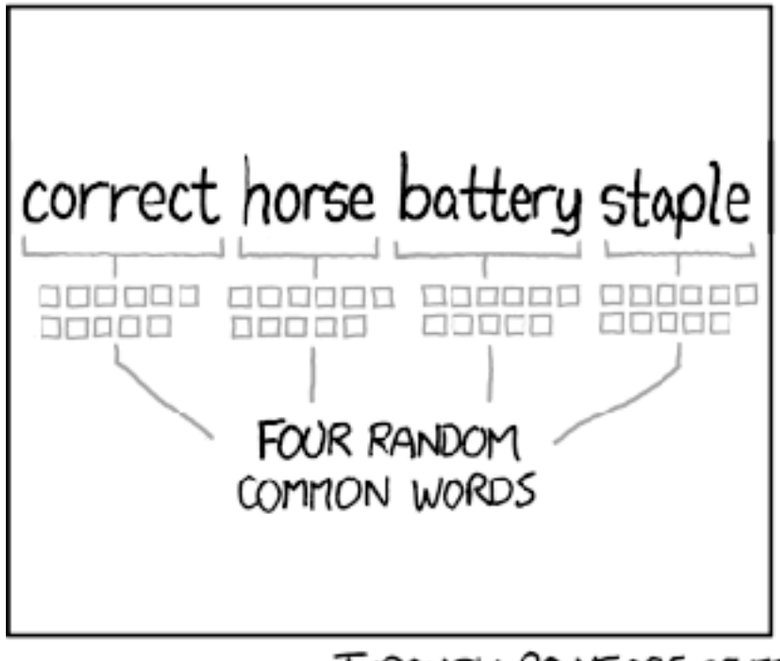
- **RULE #3**
  Do NOT use common words, a hacker can *dictionary-attack* it!!
  That includes replacing A with 4, E with 3, 0 with O, etc etc

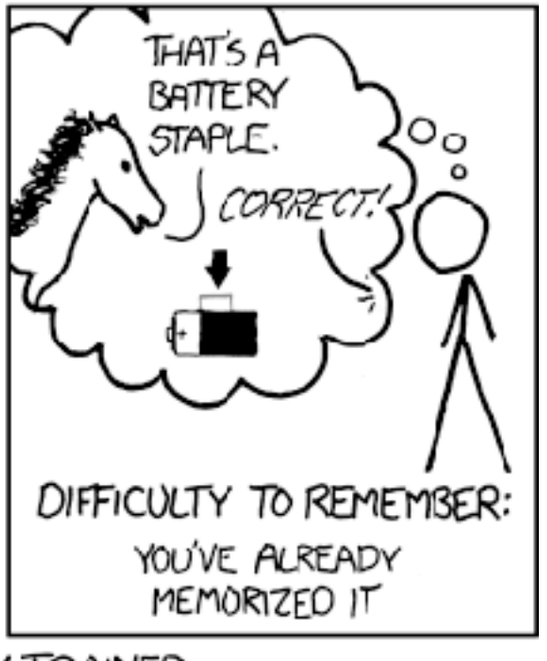
  See
  https://www.youtube.com/watch?v=3NjQ9b3pgIg
  https://www.youtube.com/watch?v=7U-RbOKanYs&t=1s

- **RULE #4**
  NEVER use the same password twice









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.

- **TIP #1**
  Use long nonsense but pronounceable words or words from some strange language

  nonsense word generator
  https://www.soybomb.com/tricks/words/

  Fairly easy to remember and easy to type

- **TIP #2**
  Place a symbol in a place that doesn't make sense

  correct_horse_battery_staple ❌
  corre_cthor_sebatte_rysta_ple ✅

- **TIP #3**
  Store your passwords in a password manager or a little physical notebook

SSH Config

- `~/.ssh/config`
- If it doesn't exist yet, you can create it

  ```
  $ touch ~/.ssh/config
  ```

- A configuration file that remembers for you which ssh parameters you want to run when you want to connect to a particular server
- `$ ssh -i ~/.ssh/<id>_rsa -XY <user>@<HostName>`

  equals

  ```
  $ ssh <hostAlias>
  ```

  when in your `~/.ssh/config` you have :

```
Host <hostAlias>
   HostName <HostName>
   User <user>
   IdentityFile ~/.ssh/id_rsa
   ForwardX11 yes
   ForwardX11Trusted yes
```

- Saves you a TON of keystrokes!!
- Also add to your `~/.bashrc` :

```
# add tab completion for remote hosts in ssh config file
complete -o default -o nospace -W "$(grep '^Host' ~/.ssh/config | cut -d' ' -f2)" ssh
```

BONUS

MAN PAGES

TLDR PAGES

```
$ apt install tldr    # Ubuntu
$ brew install tldr   # MacOS
```

BONUS


```
$ apt install bat    # Ubuntu
$ brew install bat   # MacOS
```