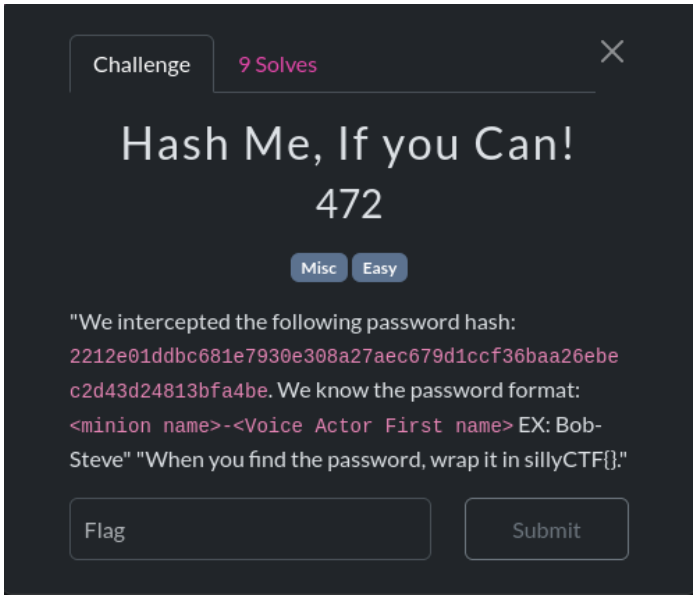

Hash given: 2212e01ddbc681e7930e308a27aec679d1ccf36baa26ebec2d43d24813bfa4be

Based on my research this is the **minion name** and **voice actor first name**.

Minion Name	**Voice Actor First Name**
Bob	Steve
Kevin	Pierre
Stuart	Jon
Dave	Jemaine
Otto	Julie
Carl	Sandra
Jerry	Trey
Tim	Danny
Mark	Michael
Phil	Miranda
Tom	Russell
Lance	Will
Ken	Benjamin
Mike	John
Josh	Lucy
Chris	Scarlett
Paul	Jean
Donny	Dana
Frank	Allison
Norbert	

Ask my best friend to create script brute force sha256 match with the hash:

```
import hashlib
import itertools

# Target hash
```

```
target_hash =
"2212e01ddbc681e7930e308a27aec679d1ccf36baa26ebec2d43d24813bfa4be"

# Expanded list of minion names (including obscure ones)
minions = [
  "Bob", "Kevin", "Stuart", "Dave", "Otto", "Carl", "Jerry", "Tim", "Mark", "Phil", "Tom", "Lance",
  "Ken", "Mike", "Josh", "Chris", "Paul", "Donny", "Frank", "Norbert"
]

# Expanded list of voice actors or related names
voice_actors = [
  "Steve", "Pierre", "Jon", "Jemaine", "Julie", "Sandra", "Trey", "Danny", "Michael", "Miranda",
  "Russell", "Will", "Benjamin", "John", "Lucy", "Scarlett", "Jean", "Dana", "Allison"
]

# Possible formatting options
formats = [
  lambda m, a: f"{m}-{a}",
  lambda m, a: f"{m.lower()}-{a.lower()}",
  lambda m, a: f"{m.upper()}-{a.upper()}",
  lambda m, a: f"{a}-{m}",
  lambda m, a: f"{a.lower()}-{m.lower()}",
  lambda m, a: f"{m}_{a}",
  lambda m, a: f"{m}{a}",
]

# Try all combinations with different formats
found = None
for minion, actor in itertools.product(minions, voice_actors):
  for fmt in formats:
    combo = fmt(minion, actor)
    hashed = hashlib.sha256(combo.encode()).hexdigest()
    if hashed == target_hash:
      found = combo
      break
  if found:
    break

# Return the flag
flag = f"sillyCTF{{{found}}}" if found else "No match found."
print(flag)
```

```
┌──(nyamuk㉿nyamuk)-[~/Downloads]
└─$ python3 minionHash.py
sillyCTF{Lance-Benjamin}
```

Flag: sillyCTF{Lance-Benjamin}