Password Cracking on a Budget

Matt Weir
Dr. Sudhir Aggarwal
Florida State University

Special Thanks To

- Bill Glodek
- Professor Breno de Medeiros
- National Institute of Justice

About Me

- Name - Matt Weir
- Occupation - PhD Student, Florida State University
- Previously worked for Northrop Grumman TASC
  - Network Security Engineer
  - Last project I supported forensic investigators working with the JTF-GNO
- Disclosed Password - xcom
  - Real strong, right?

Disclaimer

- I’m a student. I don’t crack passwords for a living
- I’ve been wrong about many things before
- I’ve probably made just about every mistake possible while learning how to crack passwords
- I’ve been known to write passwords down

How this will be Different from the Shmoocon Talk

- What! I can’t just use the same slides?
- The Shmoocon talk focused on three main areas
  - The ethics of password cracking
  - Where we get disclosed password lists to do research on
  - Our analysis of those password lists and an overview of how people actually create passwords
- You can see a video of that talk + slides at www.shmoocon.org

This is Defcon

- All that information is neat but...
  - How do you go about applying this in real life?
  - Without having to spend a lot of money
  - Note: That being said, having money to throw around makes things a lot easier

Because this is a 50 minute talk...

- I'll be available to answer questions, go into more detail, rant, and listen to better ideas afterwards
- You can also e-mail me
  - weir@cs.fsu.edu
- It's important to me that this research actually helps somebody

Getting the Tools

- We've developed a lot of custom tools and scripts to make cracking passwords easier
- While they are included on the Defcon CD, you can get the most up to date versions at the following website
- www.ecit.fsu.edu
  - Select "Password Recovery Tools"

Password Basics

PASSWORD
the classic word association game
PASSWORD Junior
the classic word association game
PASSWORD Junior
PASSWORD Junior

- I want to avoid giving everyone a CISSP prep course on password cracking

- That being said, if you have questions, please ask them

Two types of password cracking

- Online
  - Trying different passwords to log in
  - Can be slow and noisy
  - You may only be allowed a few guesses
- Offline
  - You grabbed the password file
  - You now are only limited by how fast your computer is

Password Hashes

- Hopefully your computer, website, online bank, does not keep your passwords in plain text
- If it does, then there isn’t much need to crack any passwords once someone grabs the password list

Password Hashes (continued)

- Step 1) User creates password : “password”
- Step 2) Computer Hashes the password
- MD5(“password”) = 5F4DCC3B5AA765D61D8327DEB882CF99
- Step 3) To log in the user types “password”
- Step 4) The computer hashes “password” and compares it against the hash it stored

Salts

- Salts are a value added to a password to make it harder to crack
- For example, you could add the username

MD5(“bob”+”password”)

3690eb69b329e009ecd053e27e7454b5

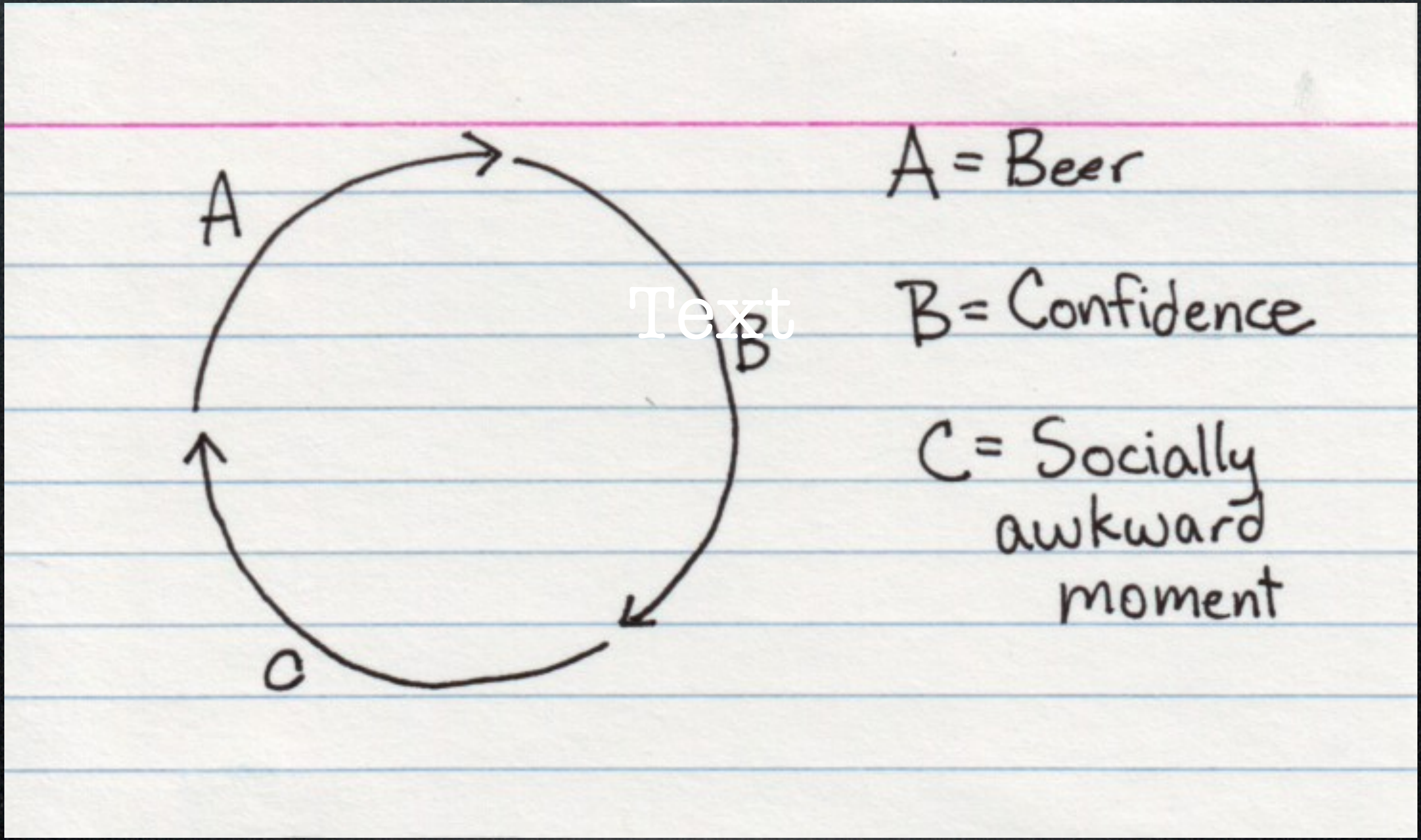
MD5(“tom”+”password”)

4125d856a8860ebf67e1fbad03167452

The Brick Wall

- There are usually two factors that can stop you from cracking a password
- You don't try the right dictionary word
- You don't try the right word mangling rule

A Quick Break to Kick off a few Demos

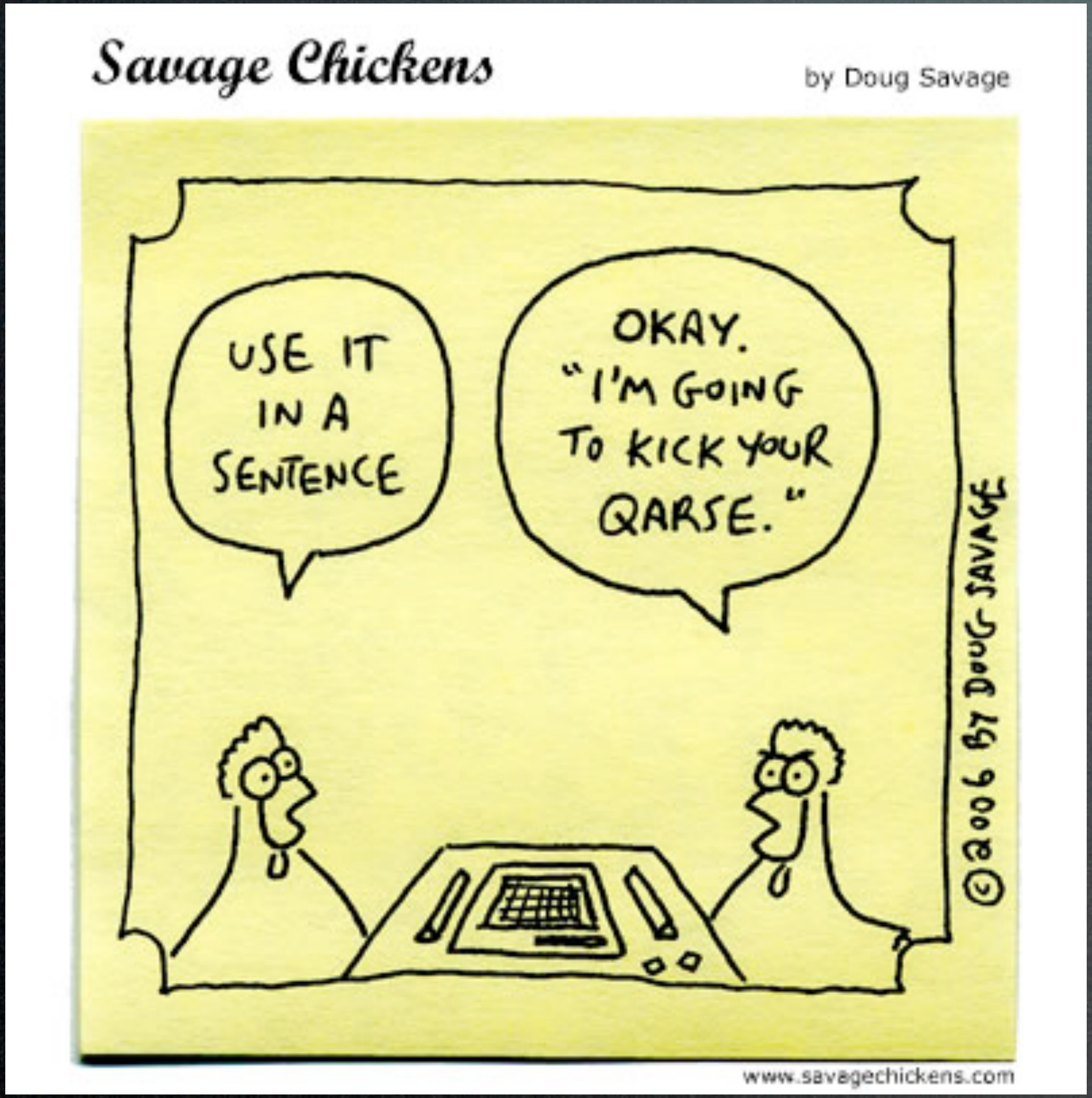


Graph stolen from indexed.blogspot.com

So you hit the wall...

- Do you try more wordlists?
  - Unless you are very careful, this can result in a lot of wasted work as wordlists often have significant overlap
- Do you try more word mangling rules
  - Advanced word mangling rules often start to resemble brute force.

Let's Talk about Wordlists



- Very important when cracking passwords
- Boring as Hell

Common Places to Find Wordlists

- http://www.word-list.com/
- http://www.outpost9.com/files/WordLists.html
- www.theargon.com/achilles/wordlists/theargonlists
- Xploits Master Password Collection on Bittorrent

Creating Better Wordlists

- The wordlists you find online leave a lot to be desired
- David Smith at Georgetown University is doing some really good work at creating wordlists off of hard drive images
- Creating wordlists by hand based on online info is a pain, but effective

The Care and Feeding of Wordlists

- Try and avoid duplicate words
- How are the words terminated?
- Standardize capitalization
- How many artifacts does the wordlist have?
- Is the word length important?

Some of our Work with Wordlists

- Wiktionary grabber
  - Creates language specific word lists
- Wikipedia grabber
  - Attempts to create custom wordlists based upon search criteria
  - Still needs a lot of work

Judging Dictionaries Based on Edit Distance

- We originally created customized dictionaries based on grabbing the alpha characters from disclosed password lists, (and making some assumptions)

  - P@ssword12 = password

  - *stuff* = stuff

  - firewa11 = firewa (Problem)

Edit Distance (Continued)

- Look at the edit distance between a password and an input dictionary
- Cons:
  - Can produce false positives and negatives
  - Only as good as the input dictionary
- Pros:
  - Produces useful custom wordlists
  - Quickly evaluates how good current wordlists are

Evaluation of Dictionaries vs. Myspace

- dic-0294
  - Description: Really BIG Dictionary
  - Percentage Found: 49.9%
  - Size: 869,228 Words
- words.english.txt
  - Percentage Found: 10.6%
  - Size: 213,557 Words

- common-password.txt
  - Percentage Found: 5.3%
  - Size: 816 Words
- Wiktionary English Words
  - Percentage Found: 32%
  - Size: 68,611 Words

Time to Check in on our Demos

Word Mangling Rules

- Generally what people focus on in password cracking
- Most password crackers are fairly limited in their rule sets
- LANMAN hashes spoiled us

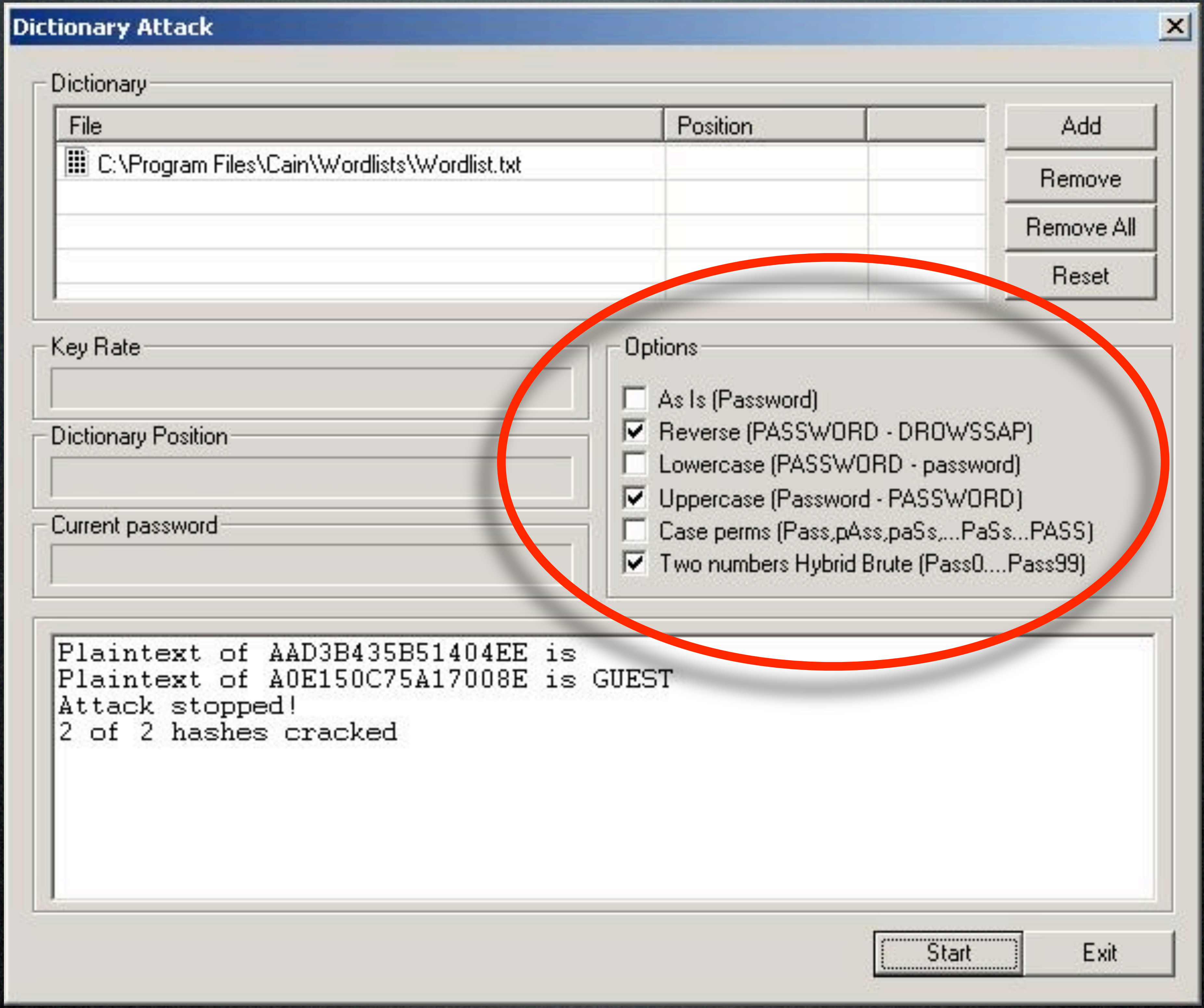
Word Mangling Rules + Teamwork = Hard

- It’s easy to crack passwords created with only one mangling rule
- The trick is dealing with passwords that use more than one mangling rule
  - P@ssWord12
- Or they use a nonstandard rule
  - p7assword

Cain and Able vs John the Ripper

- They are the two major free password crackers out there
- Which one should you use?
- Answer:
  - John the Ripper

Why not Cain and Able?



C:\Program Files\Cain\Wordlists\Wordlist.txt

- ☐ As Is (Password)
- ☑ Reverse (PASSWORD - DROWSSAP)
- ☐ Lowercase (PASSWORD - password)
- ☑ Uppercase (Password - PASSWORD)
- ☐ Case perms (Pass,pAss,paSs,...PaSs...PASS)
- ☑ Two numbers Hybrid Brute (Pass0....Pass99)

```
Plaintext of AAD3B435B51404EE is
Plaintext of A0E150C75A17008E is GUEST
Attack stopped!
2 of 2 hashes cracked
```

Getting the Most out of John the Ripper

- Install the unofficial patches if you need support for other types of hashes
- Do NOT use the default john.config file
    - It's a pain, but learning the rule syntax is very useful
    - The RULES readme file is your friend

Brute Force with John

- By default, JtR uses Markov models to generate brute force guesses
  - You can actually train the Markov model based on passwords you already have
  - Warning: it does require a lot of passwords to train it

Targeted Brute Force

- Often you will want to brute force certain types of passwords
- AKA six letters followed by two numbers
- You can do this in John, but it’s a bit of a hack

Targetd Brute Force (continued)

- Create a input wordlist of a-z
  - aka a b c d e f g ..... z
- Now create a rule to add all the other values
  - $[a-z]$[a-z]$[a-z]$[a-z]$[0-9]$[0-9]
- You can even get fancy and apply some Markov models of your own

Probabilistic Context Free Grammar

- Guess which project we are writing a paper on...
- In a nutshell, it allows you to define very detailed rules easily
- It assigns a probability to every word mangling rule, number, word, capitalization, special character, etc

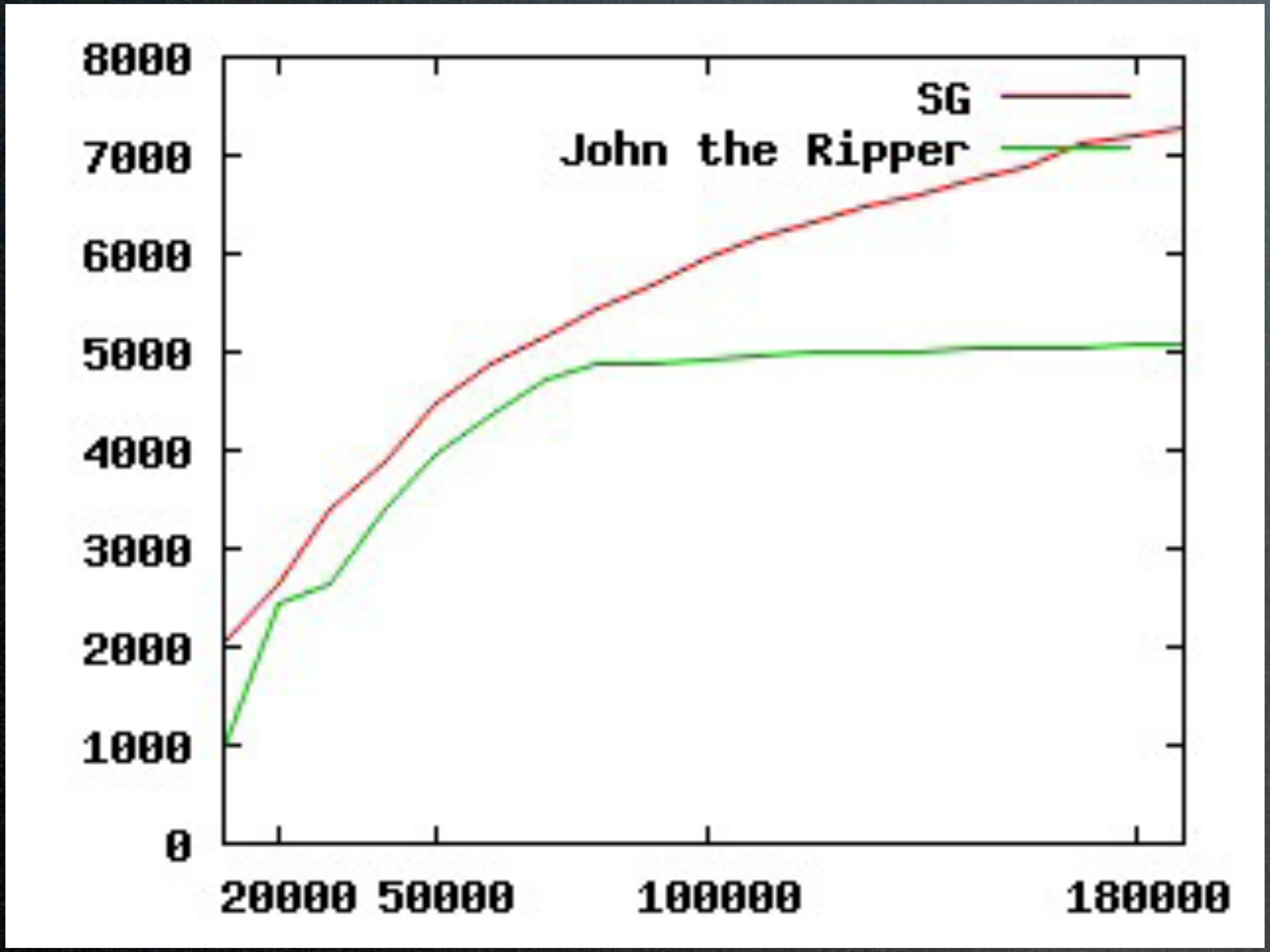
PCFG Password Crackers (continued)

- It is trained off of existing password lists
- This way, depending on the probabilities, it might try the following guesses in this order
  - password12
  - password!
  - password13
- You can simulate it to a certain extent by creating 100s, (or 1000s) of rules in John the Ripper

Using our PCFG Password Cracker

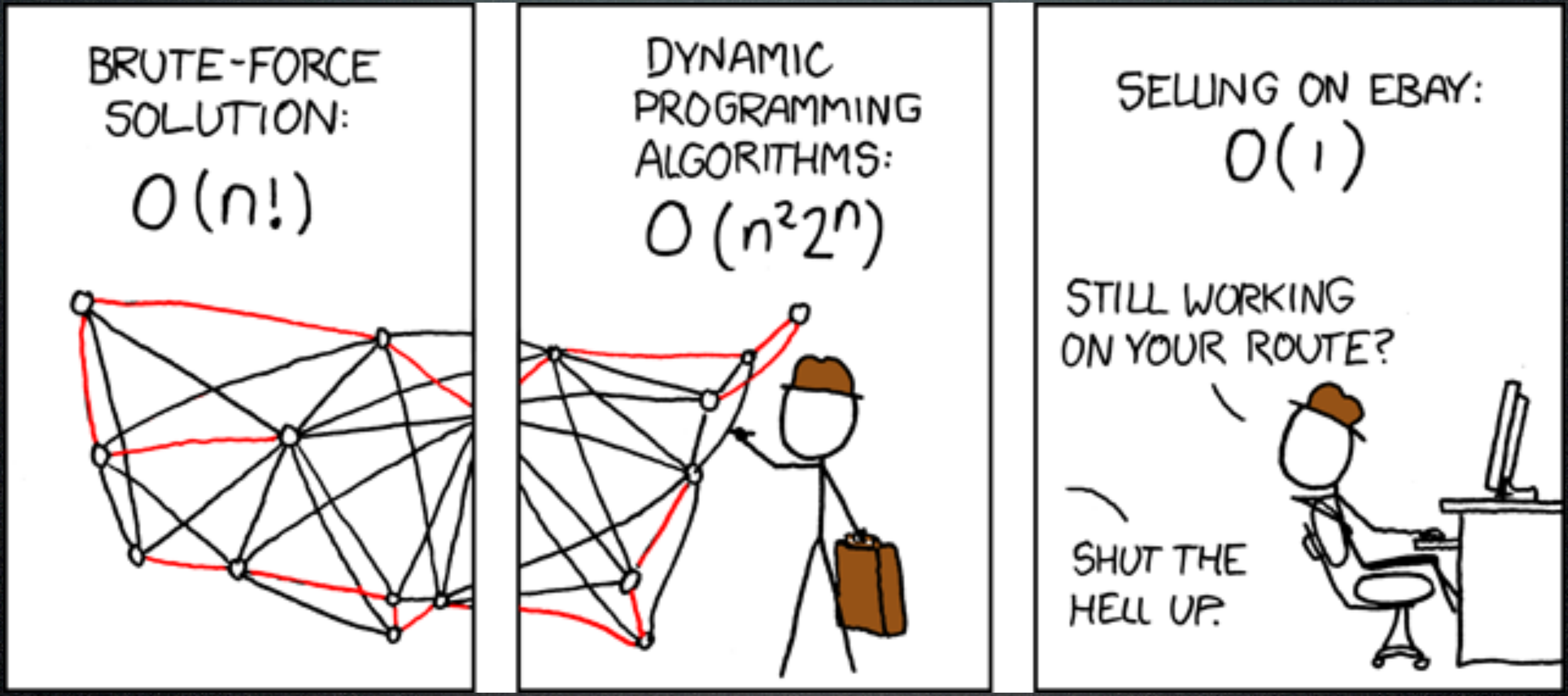
- It currently makes guesses and outputs them to stdout
- Pipe the guesses into JtR since we didn't want to write our own hashing / management software
- It does have some overhead, but going against strong hashes it's not significant

Gotta Have at Least One Graph



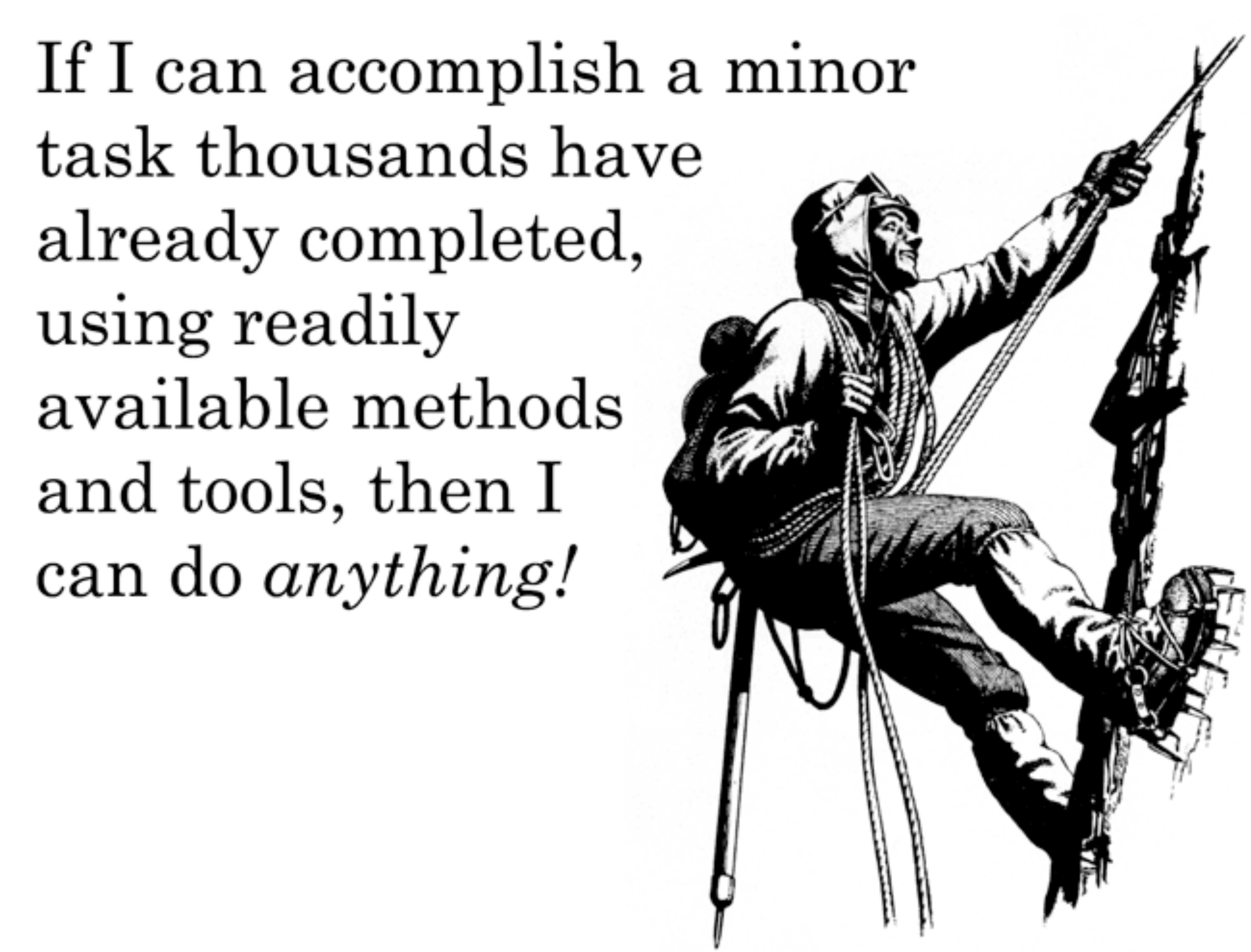
- Measures the performance of the default JtR rule set vs our PCFG
- X-axis=number of guesses
- Y-axis=number of found passwords

Check Final Results of Demo



Picture stolen from xkcd.com

Questions / Comments


If I can accomplish a minor task thousands have already completed, using readily available methods and tools, then I can do *anything!*

- Matt Weir
- weir@cs.fsu.edu
- www.ecit.fsu.edu

Picture stolen from marriedtothesea.com