Cracking 400,000 Passwords

Matt Weir
Sudhir Aggarwal
Florida State University

Special Thanks:

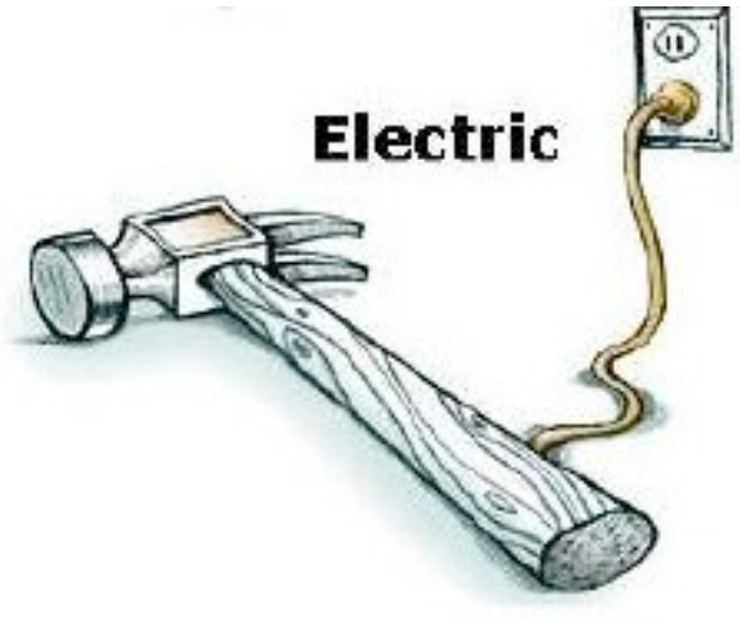
- Dr. Sudhir Aggarwal
- Professor Breno de Medeiros
- National Institute of Justice
- National White Collar Crime Center

My Research

- Assist Law Enforcement
- Develop better ways to model how people actually create passwords
- Investigate how we can make passwords more secure

What I’m going to try and avoid focusing on…

Tools



Trivia

For Tools and Trivia…

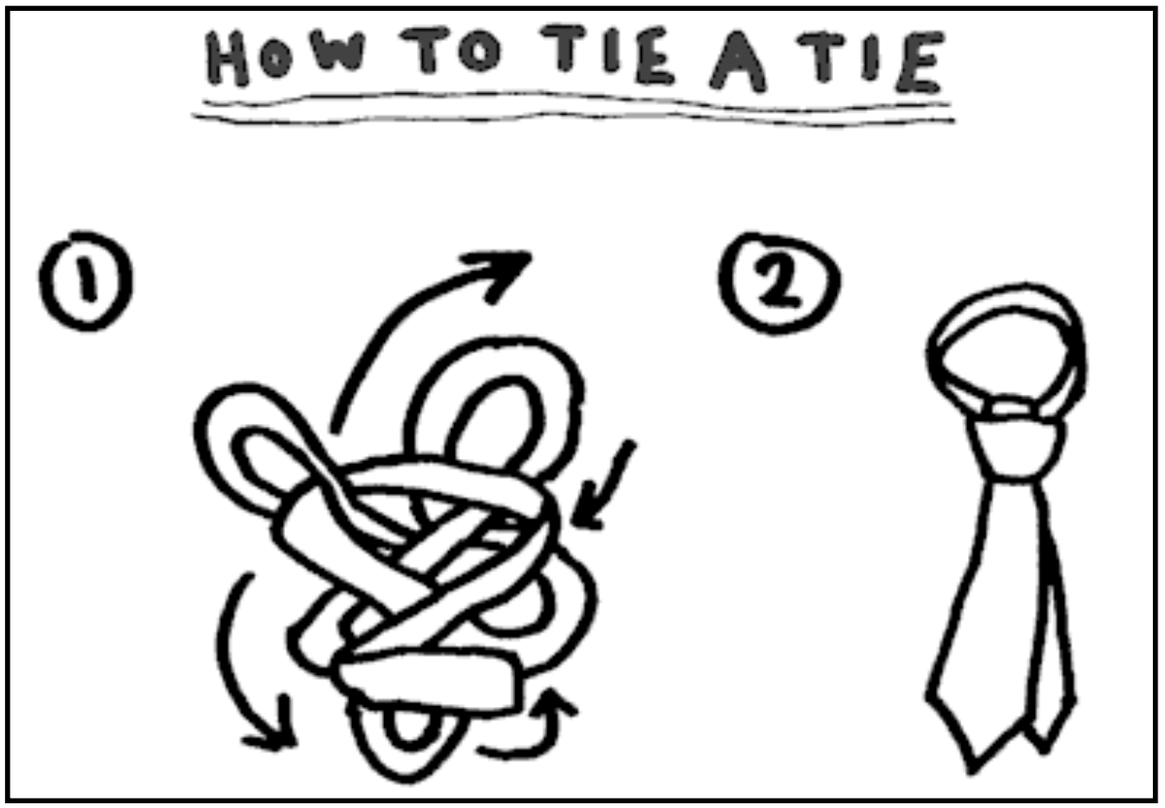
- My Research Blog
  - http://www.reusablesec.blogspot.com
- Tools Page
  - http://sites.google.com/site/reusablesec/

And Certainly Not…

- OMG Passwords Suck!
- Users are stupid!
- We're all doomed!

The Main Goal

- What does a password cracking session look like?
- What steps go into cracking a password list?



The Plan

1. Password Cracking Basics, (for the CISSPs out there)
2. Cracking the phpbb.com list
3. Cracking the webhostingtalk.com list
4. Breakout Room: Questions + Dealing with TrueCrypt, pass-phrases and non-standard passwords

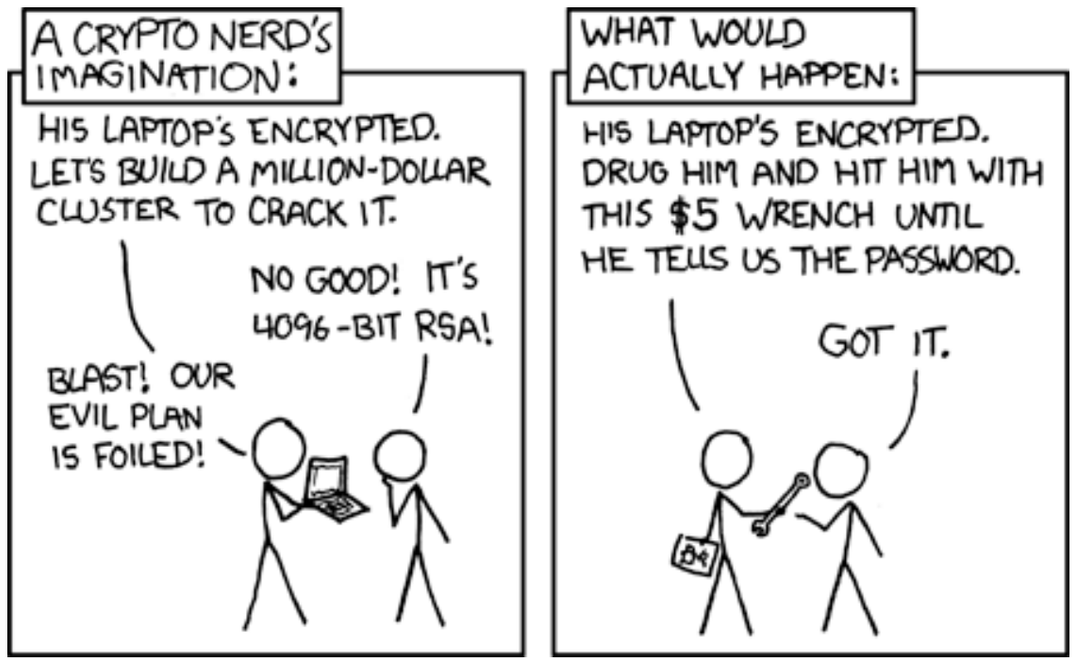
Password Cracking Basics

Two Types of Password Cracking

- Online
  - The system is still operational
  - You may only be allowed a few guesses
- Offline
  - You grabbed the password hash
  - Computer forensics setting

Cracking Passwords

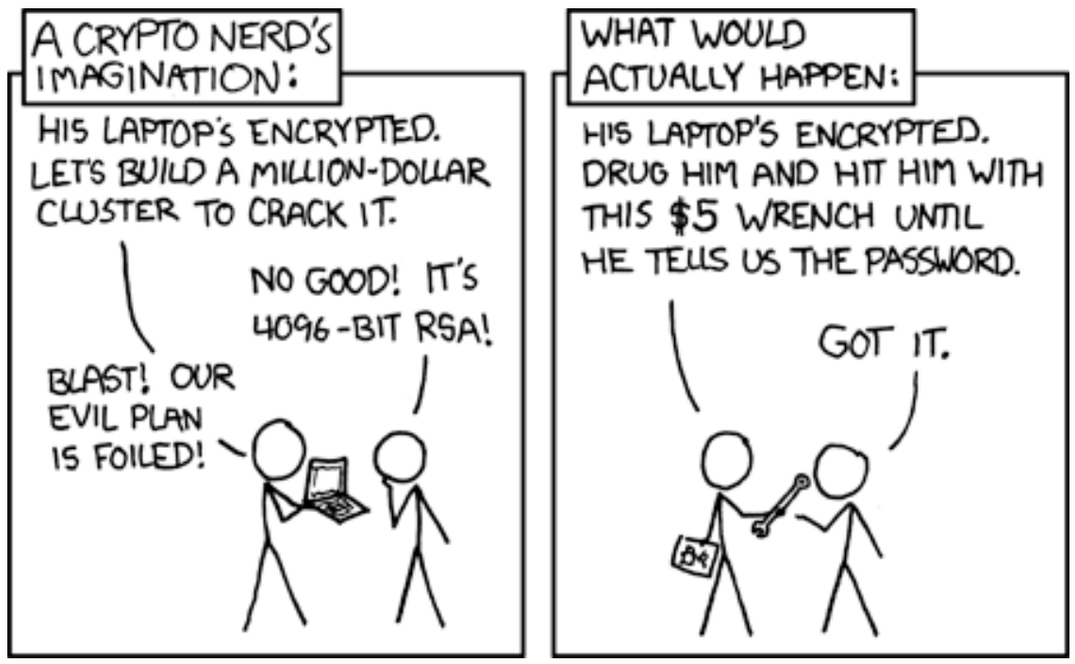
Step 1) Create a password guess



Cracking Passwords

Step 1) Create a password guess

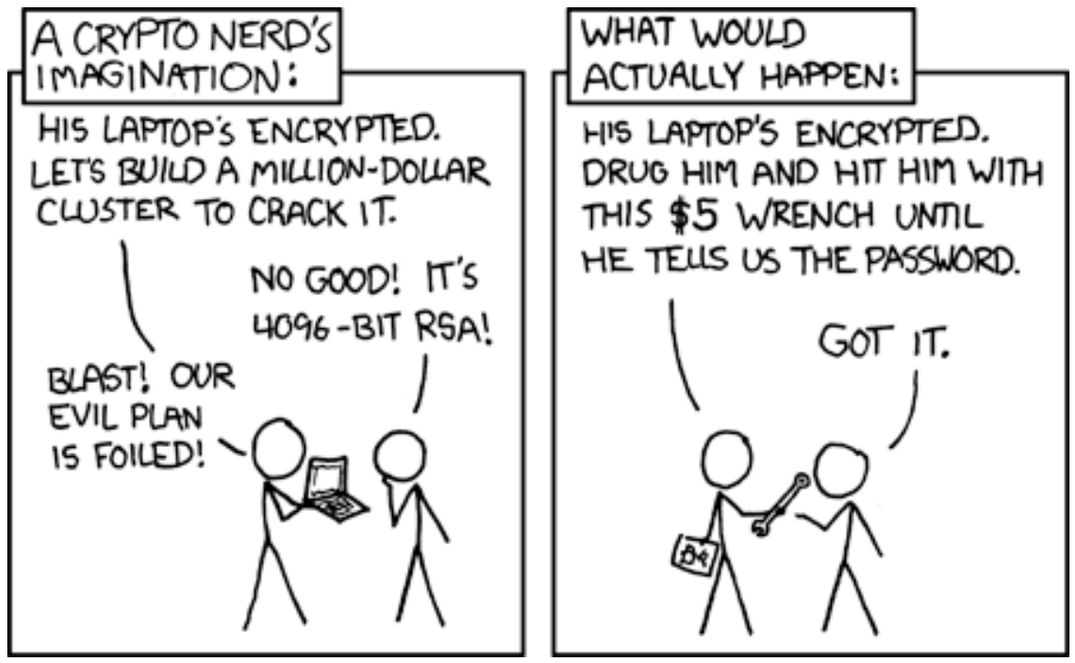
password123



Cracking Passwords

Step 2) Hash the Guess

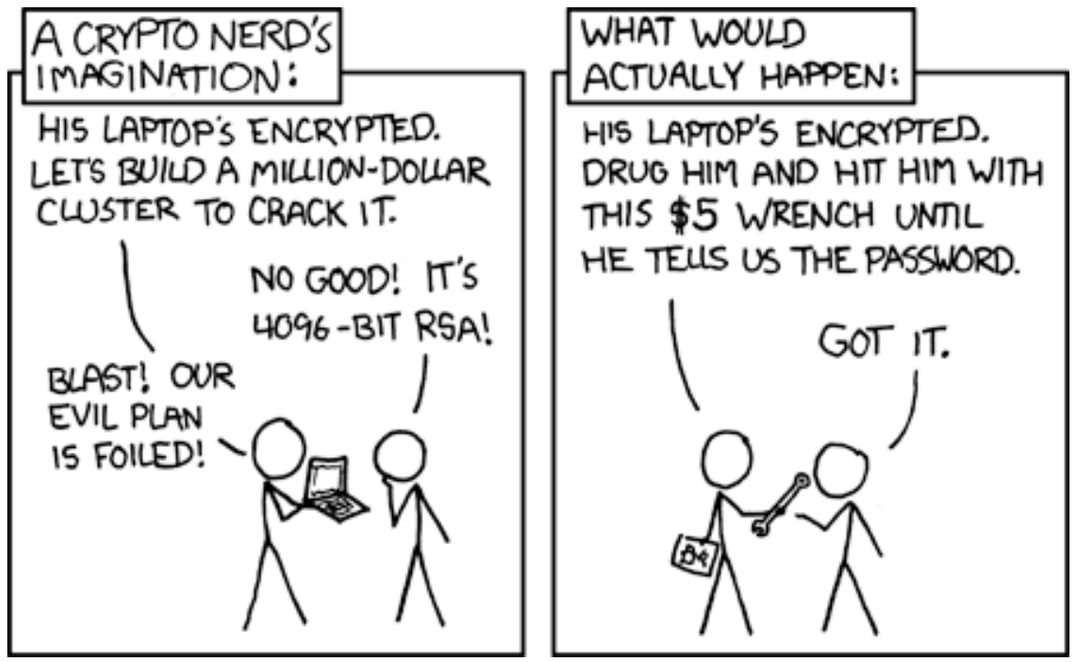
password123



Cracking Passwords

Step 2) Hash the Guess

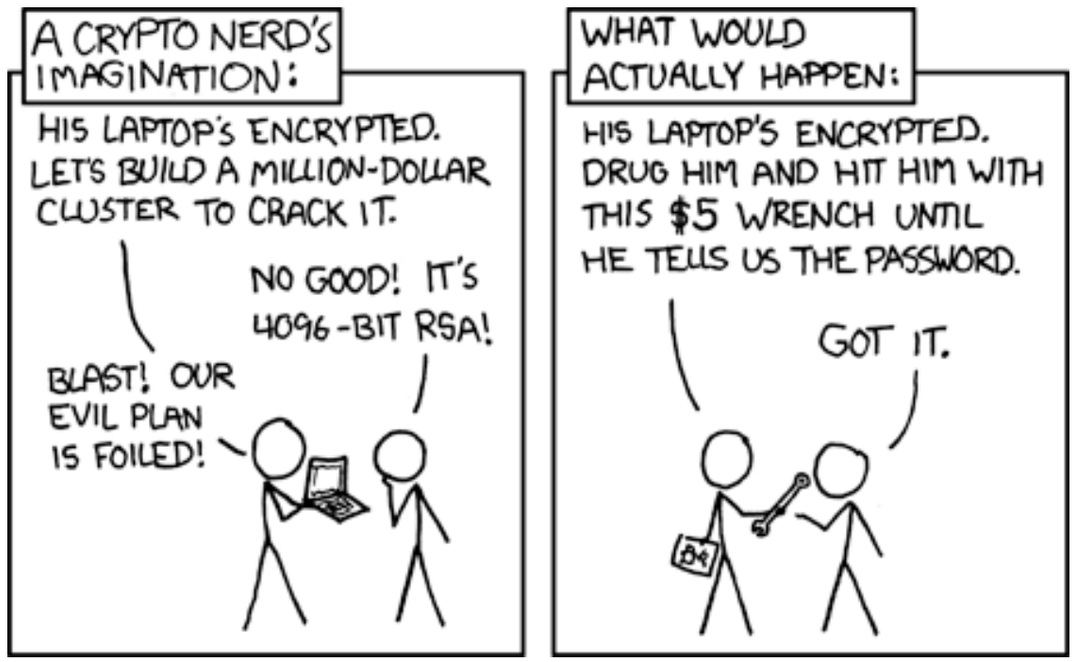
52F8A73082B1290



Cracking Passwords

Step 3) Compare it against the target hash

52F8A73082B1290



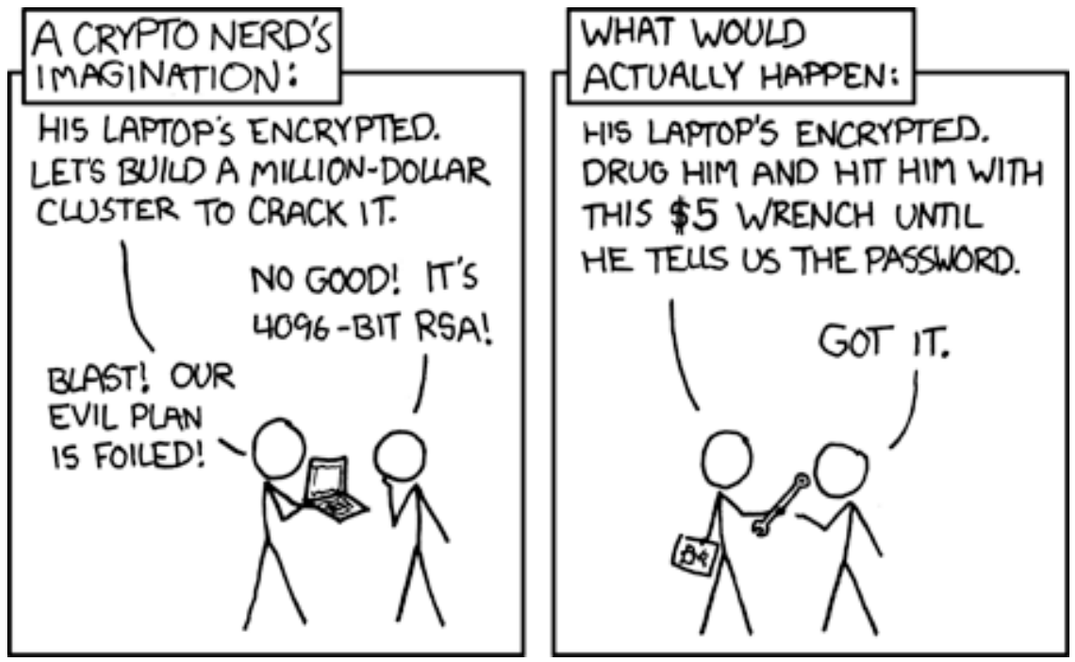
Cracking Passwords

Step 3) Compare it against the target hash

52F8A73082B1290

=

82503CA693453D1



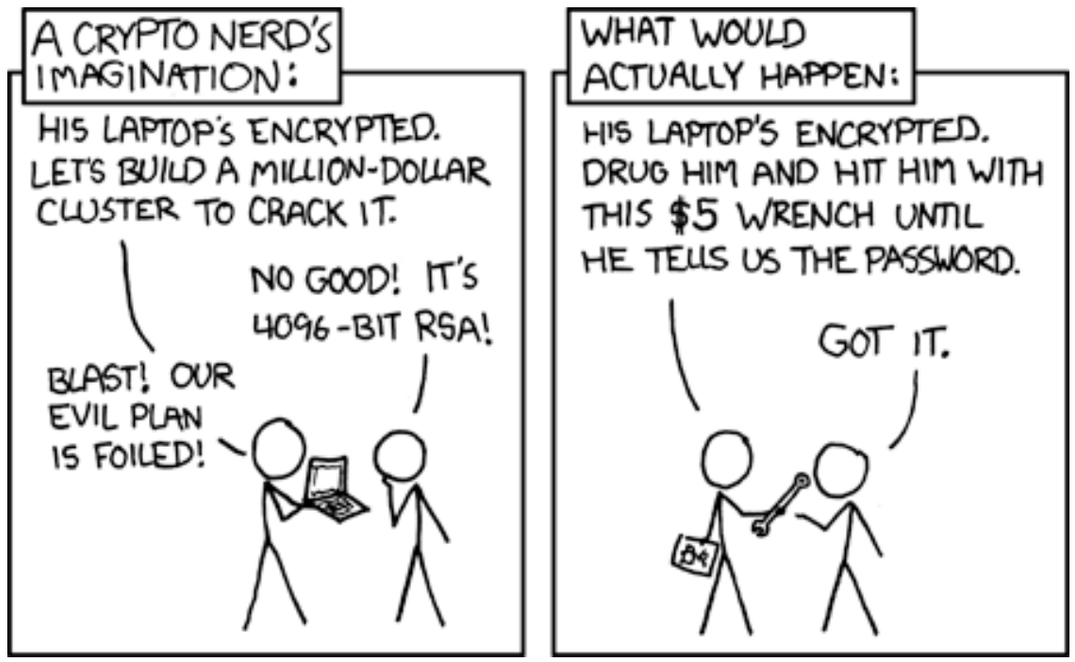
Cracking Passwords

Step 3) Compare it against the target hash

52F8A73082B1290

≠

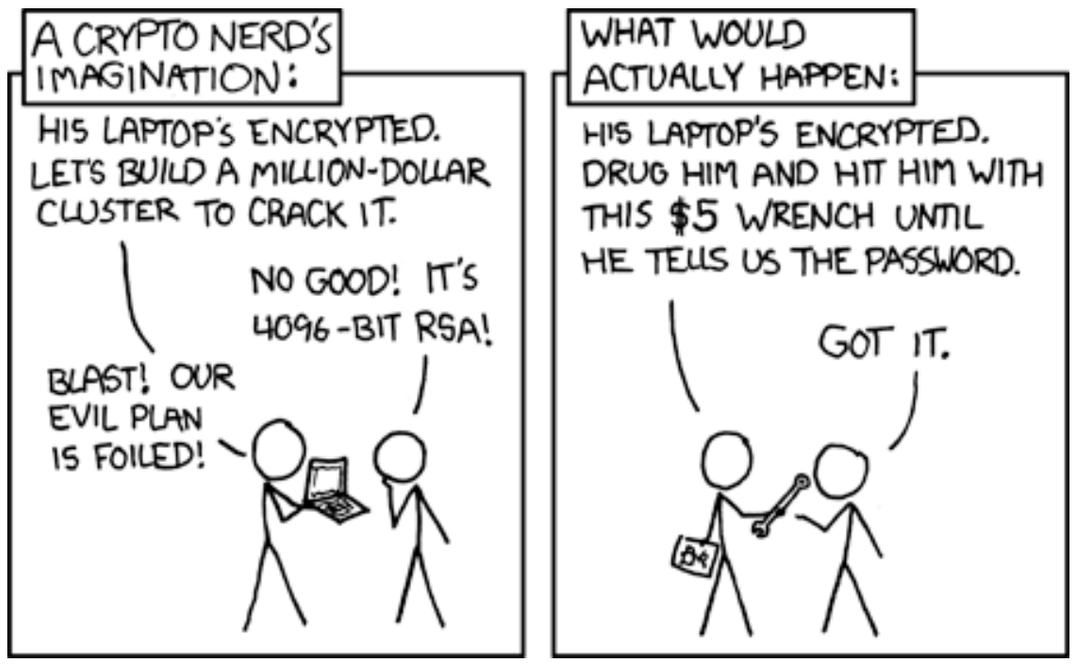
82503CA693453D1



Cracking Passwords

Step 1 ... again) Make another guess

monkey123



Password Salts

- Salts are a value added to a password to make it harder to crack
- For example, you could add the username
  - MD5(“bob”+”defcon”)
  - **09f20200fe8131d1114581e916381d04**
  - MD5(“tom”+”defcon”)
  - **b19263f7cadf7a03ee644ad60591a91c**
- In real life, use a RANDOM value

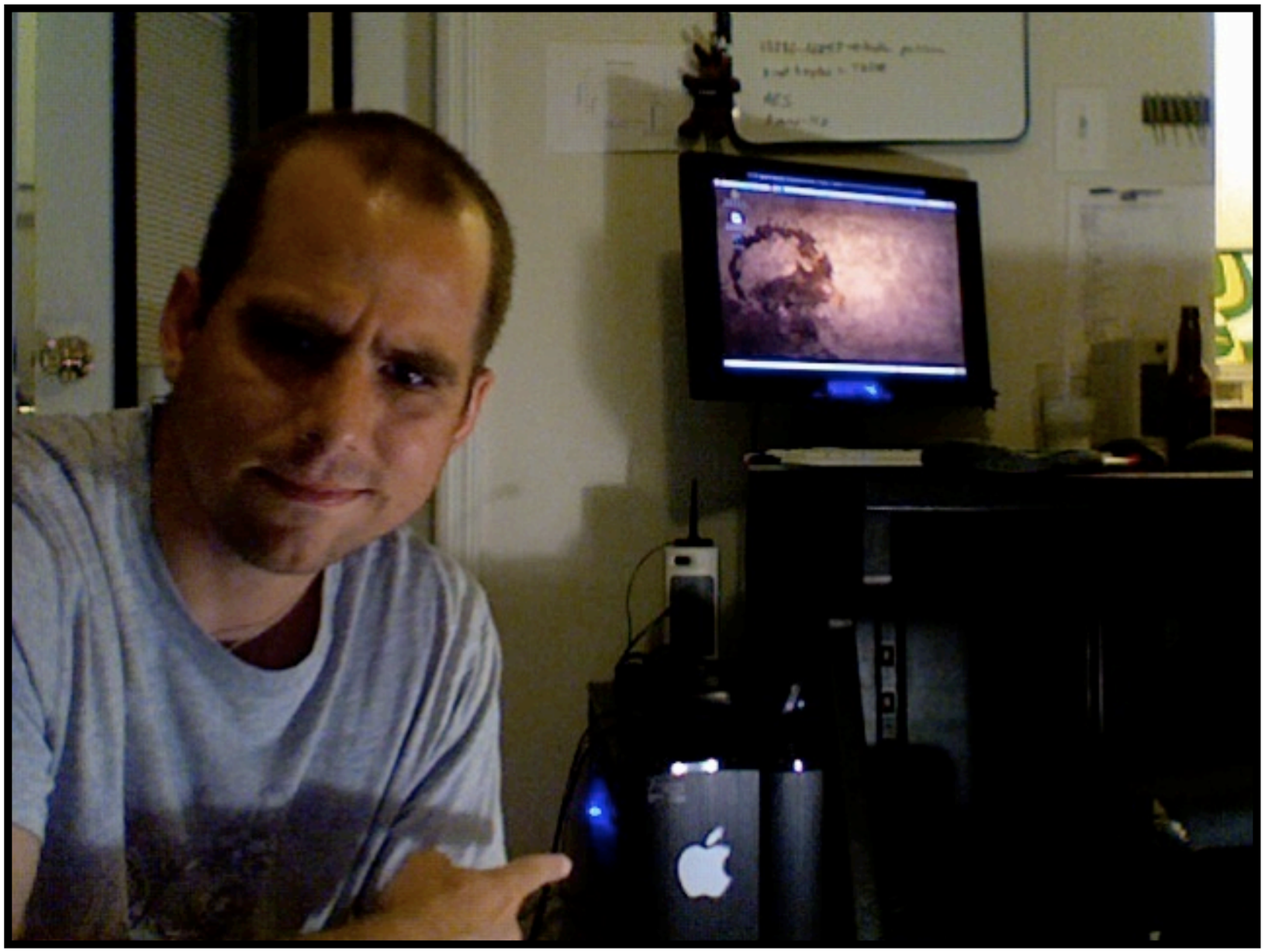
Password Salts (cont.)

- Important Points
  - Not secret
  - User does not need to know it.
  - Should be unique per user
  - If the attacker is only targeting one user, it only prevents against hash lookup attacks

Now on to the Cracking!

Original Hardware Setup

- 2.4 GHz Core Duo
- 3 Gigs of Ram
- NVIDEA GeForce 8800 GTS

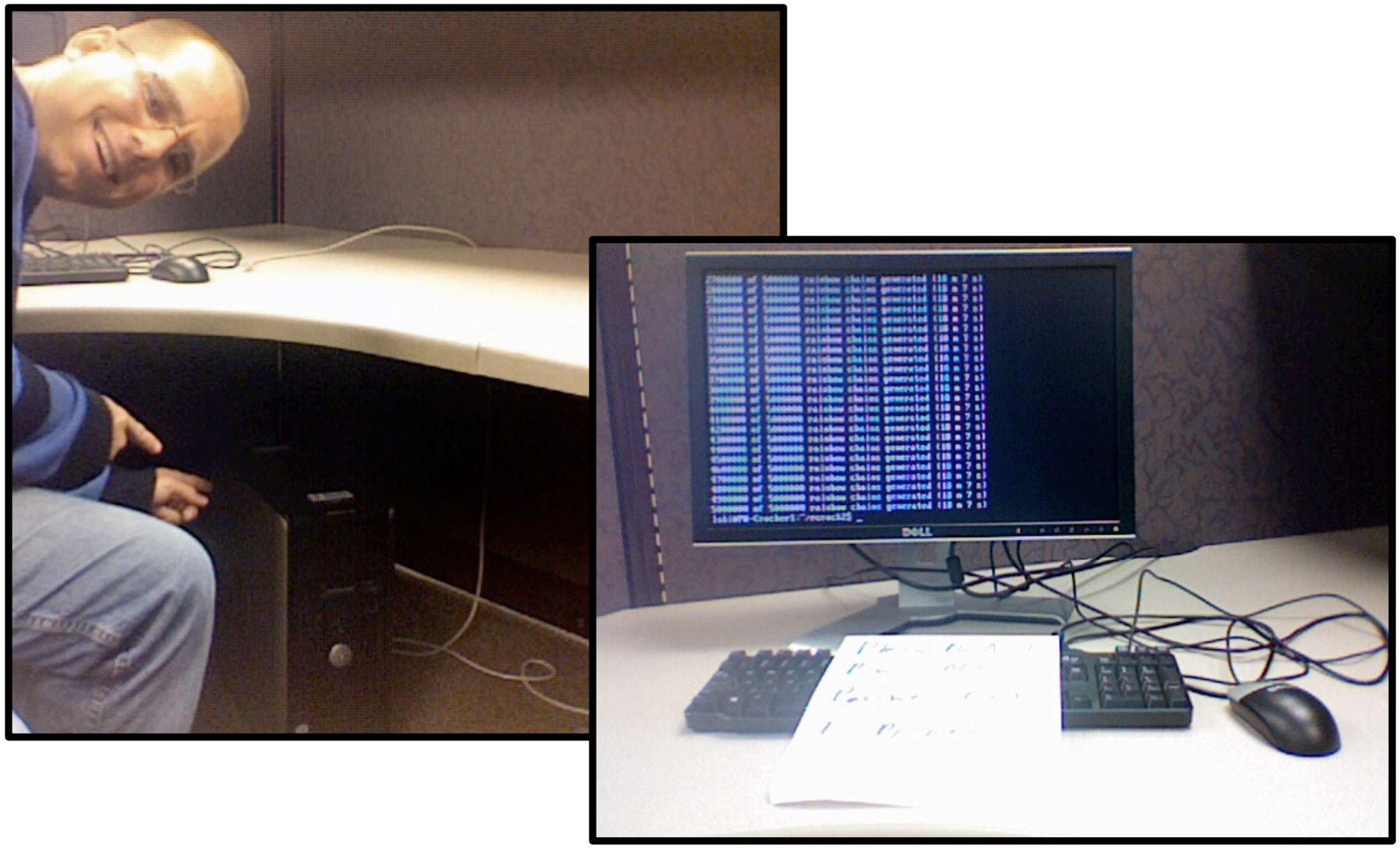


Then the Power-bill Arrived…

- Our power bill had gone up by about 75%
- There were other causes as well but that's a hard conversation to have...

Picture of Power Bill Removed to Avoid Further Annoying My Roommate, (the power bill is in his name)

Current Hardware Setup



The Phpbb.com List

- Development site for the phpbb forum software
- Originally Hacked Jan 14th 2009
- List was posted online early February

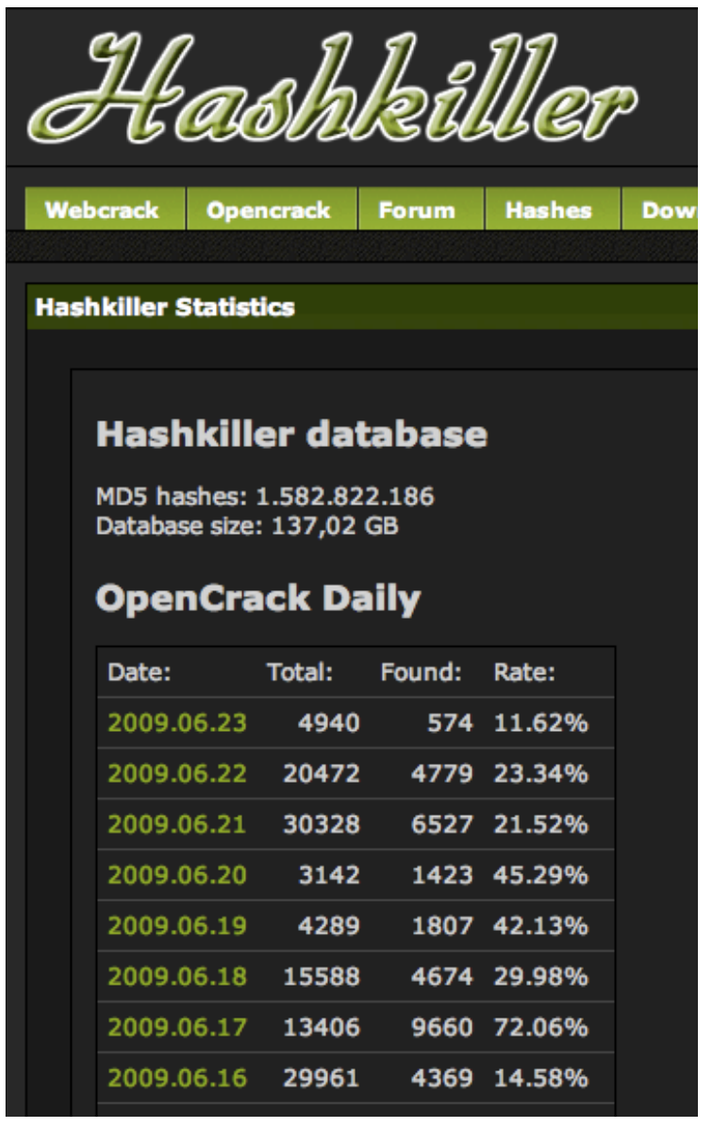
Details About the List

- Contained 259k unsalted MD5 password hashes
- Also had 83k salted hashes using the phpbb3 hashing algorithm
- We only attacked the MD5 hashes

The Hacker's Attack

- The hacker had attempted to crack 117k of the password list
- Used an online web-cracker over a one to two week period
- Cracked 28,635 passwords, aka 24% of them

Comparing Online Password Crackers

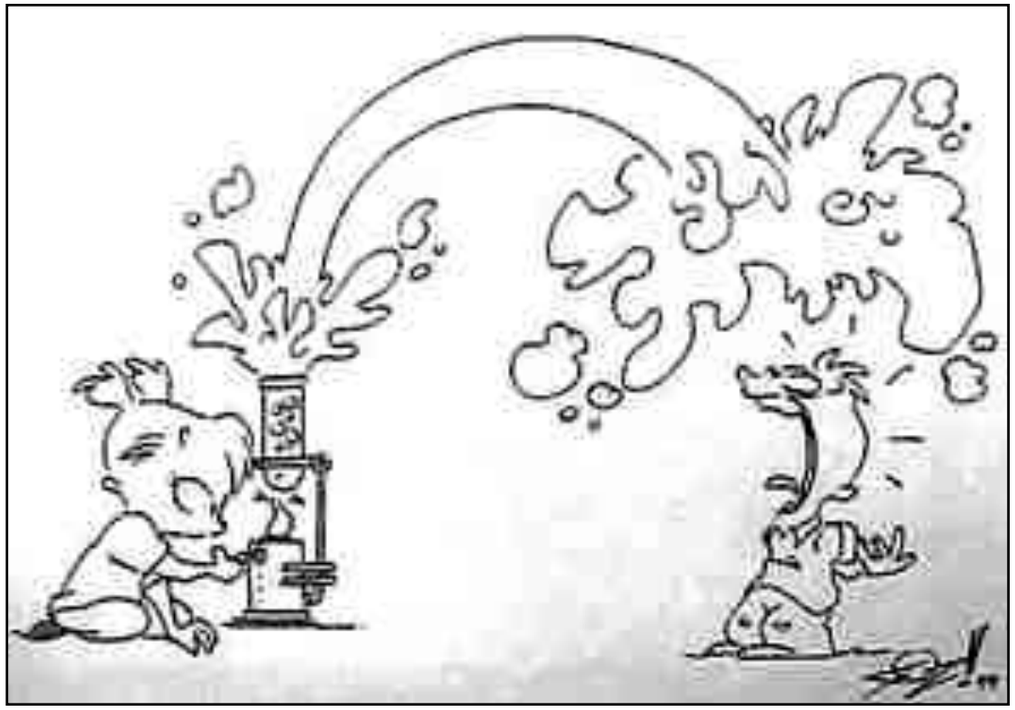


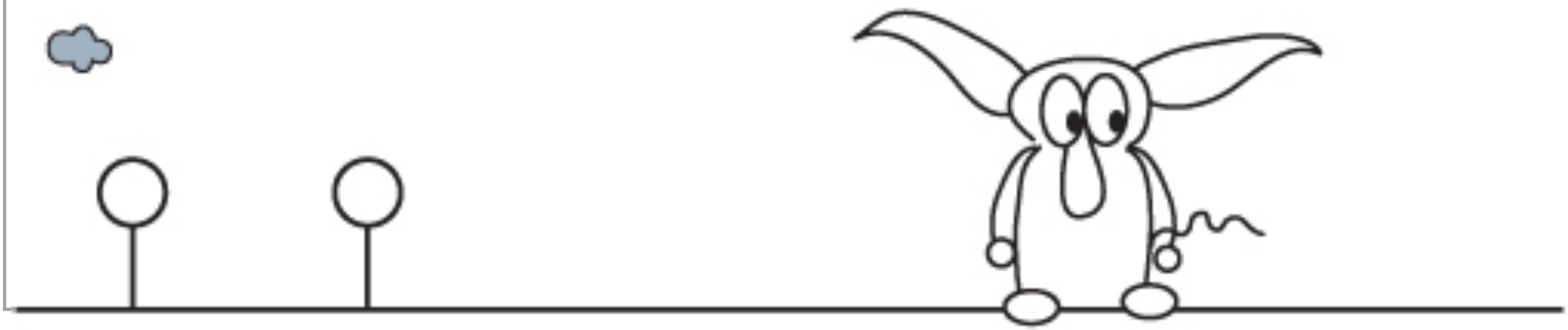
- www.hashkiller.com
- Most online password crackers crack around 20-40% of passwords submitted to them
- MD5-utils will submit password hashes to many of the online sites

  http://sourceforge.net/projects/md5-utils/

Existing Password Crackers

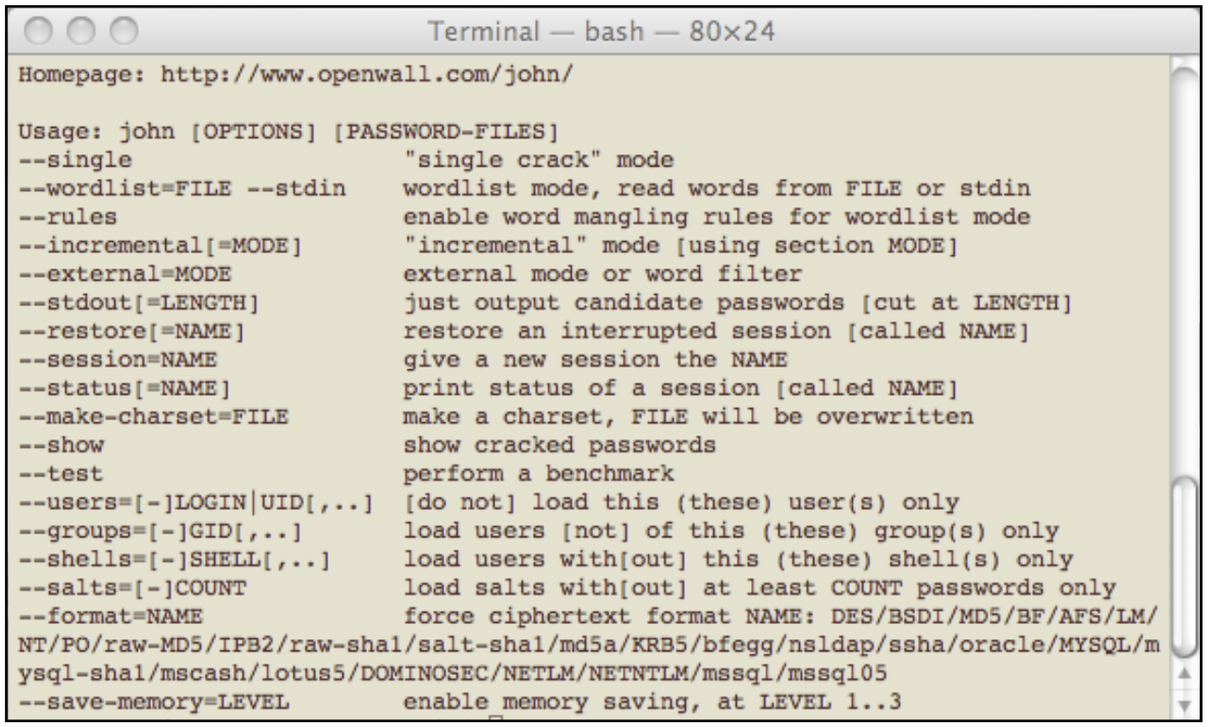
- John the Ripper
- Cain & Able
- L0phtcrack
- Access Data's PRTK





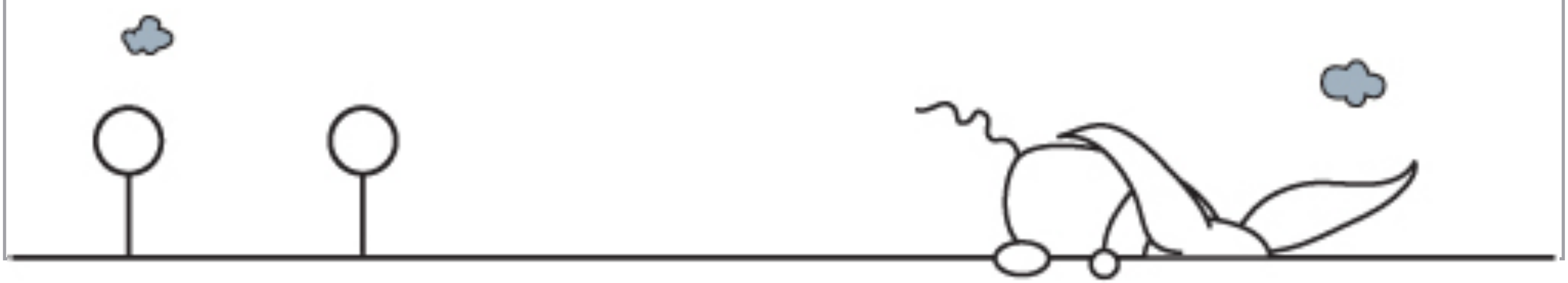
John the Ripper

- Source-code is available
- If you can think of it, it's probably been done in JtR



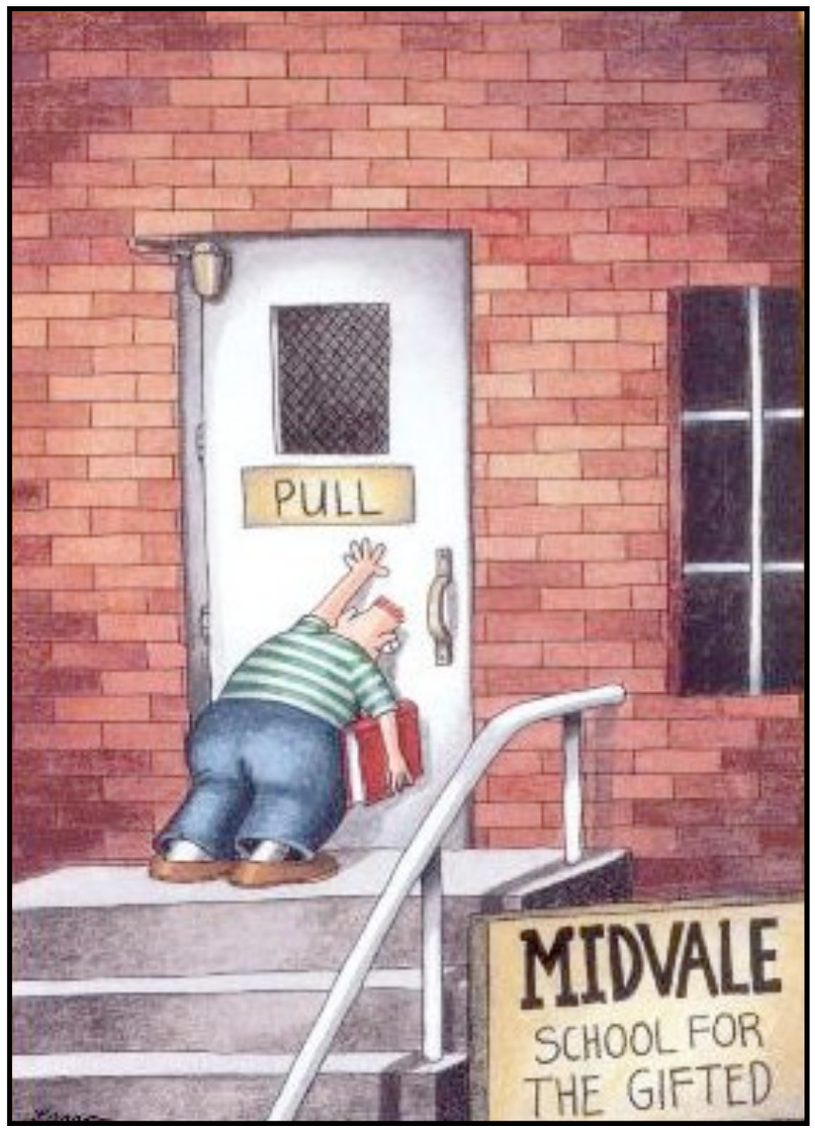
My Favorite Option in JtR

-STDIN



Make Sure You Check For Updates…

- Older versions of JtR choke when passed a large password list
- There was a patch, but I didn't realize it until later...



Time it took to crack, (even with JtR problems)

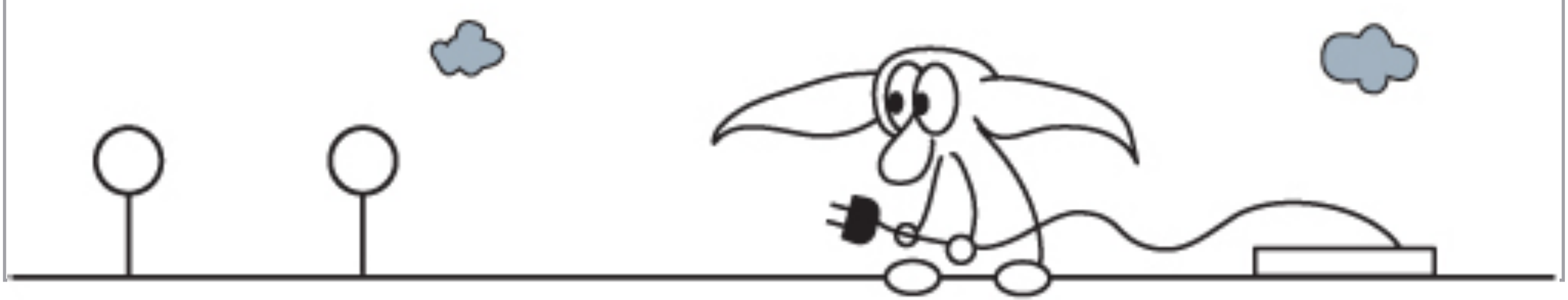
- “4 hours” - 38% of the passwords cracked
- 1 week - 62% of the passwords cracked
- 1 month + 1 week - 89% of the passwords cracked
- Currently - 95% of the total passwords cracked
  - 93% of the unique MD5 hashes

Other Results

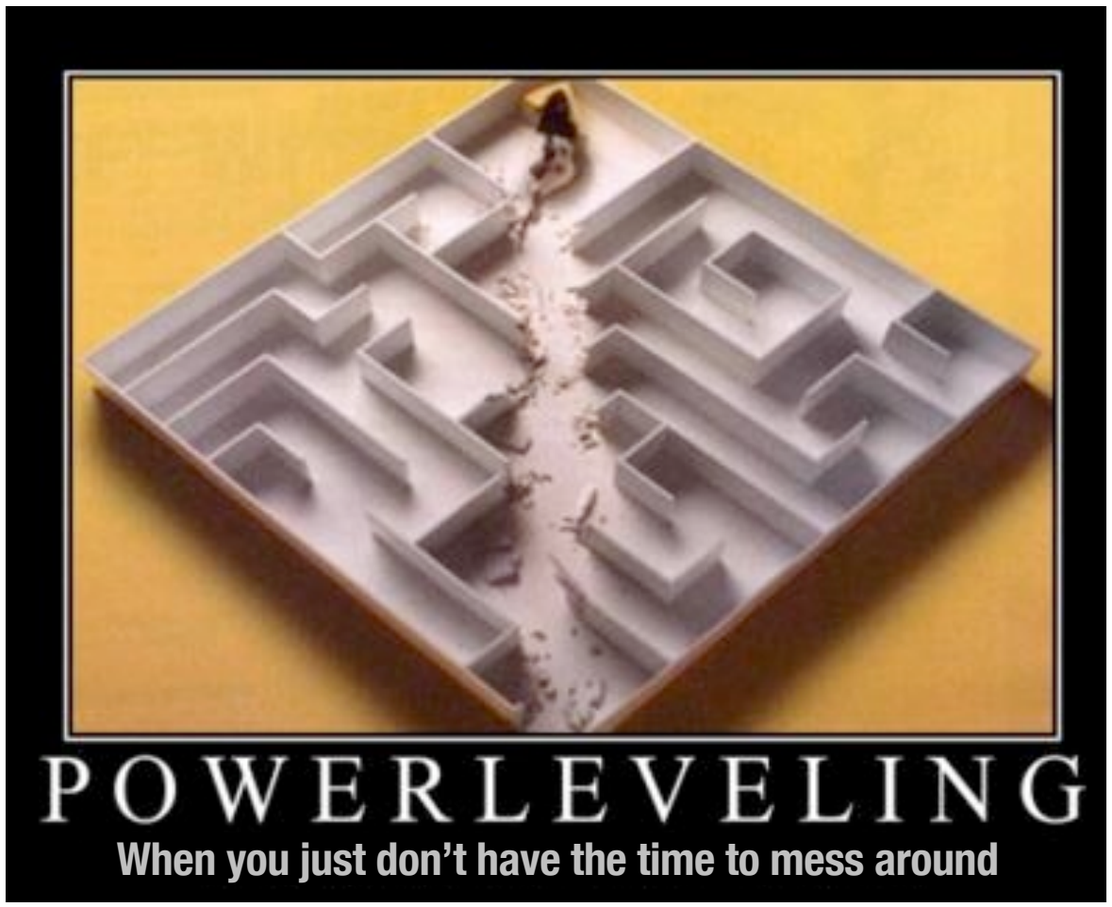
- Brandon Enright - 95% of the MD5 hashes cracked
  - He cracked 2,525 unique hashes that I missed
  - I've cracked 2,677 unique hashes that he missed

Some Quick Statistics

- Average Length: 7.2 characters long
- Only 6% of them contained an UPPERCASE letter
- Only 1% of them contained a special character!
- 51% contained only lowercase letters
  - ✳ Note: Does not include the 5% of the passwords we have not cracked - duh

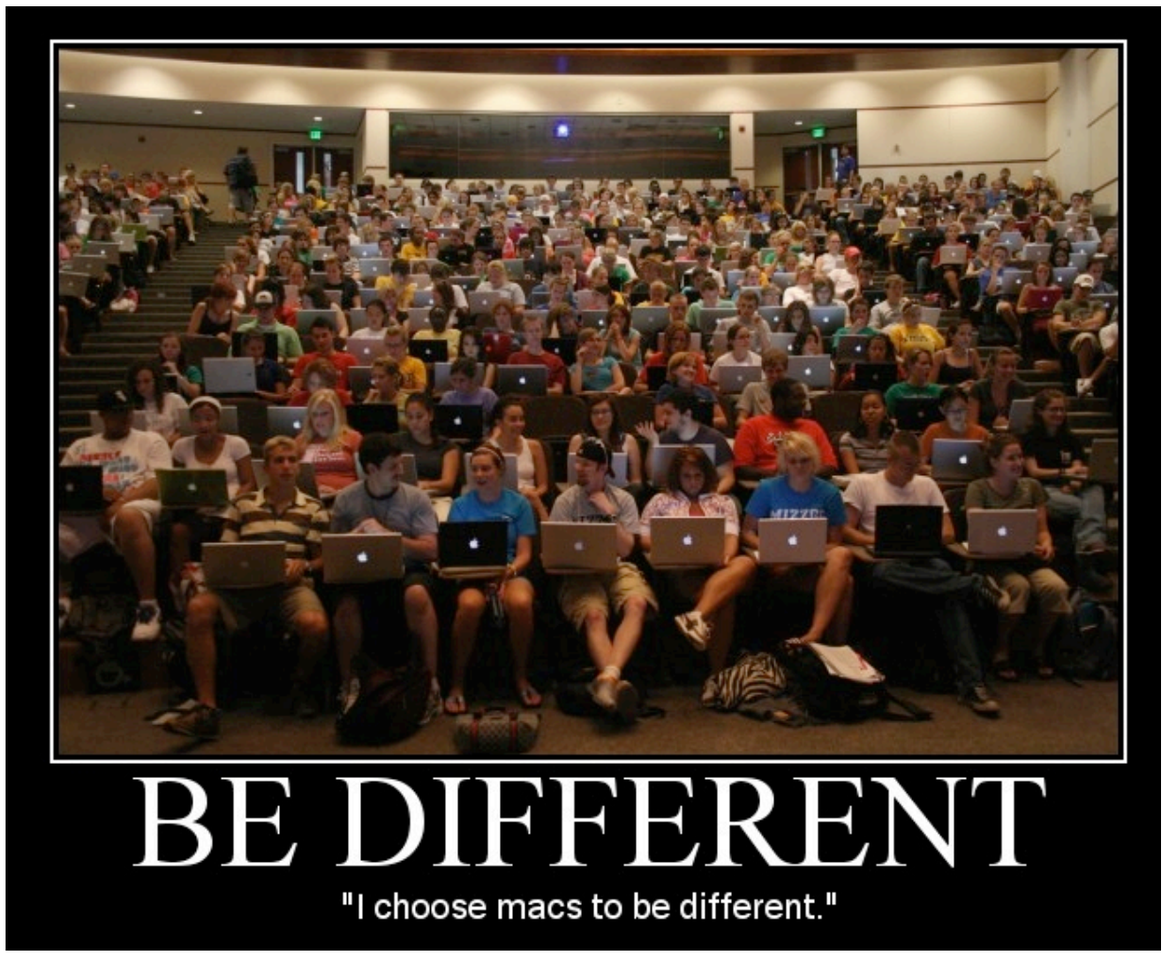


Limited Resources



- Unless we're attacking LANMAN, we're limited in the time we can spend
- Therefore, we have to choose between different attack strategies
- We can't just try everything

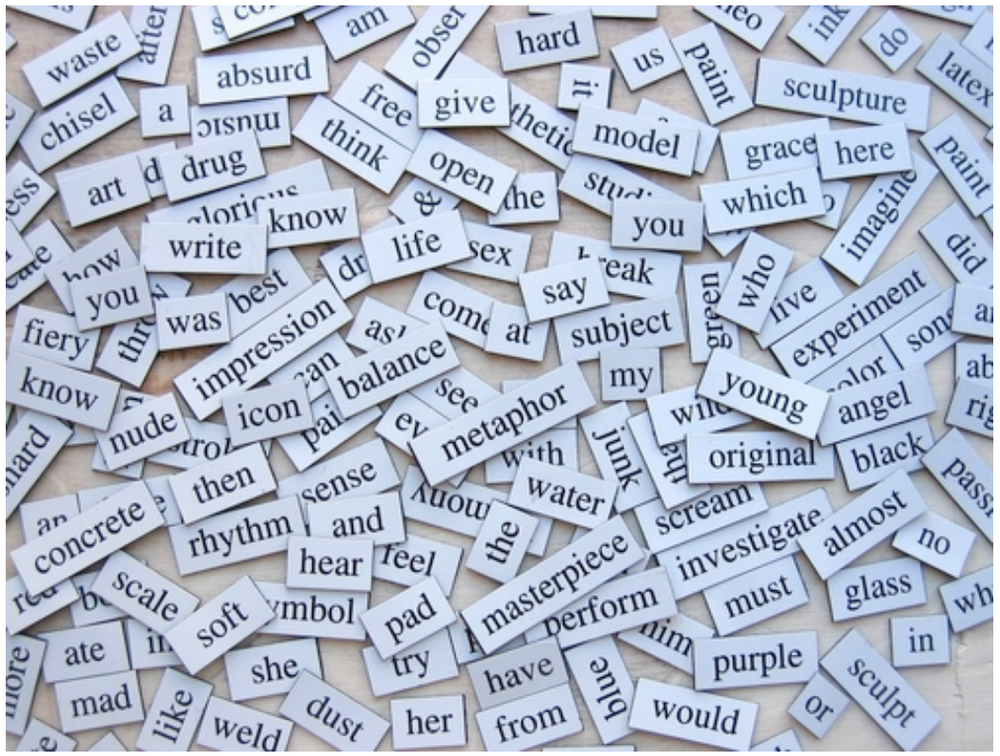
Creating Strong Passwords

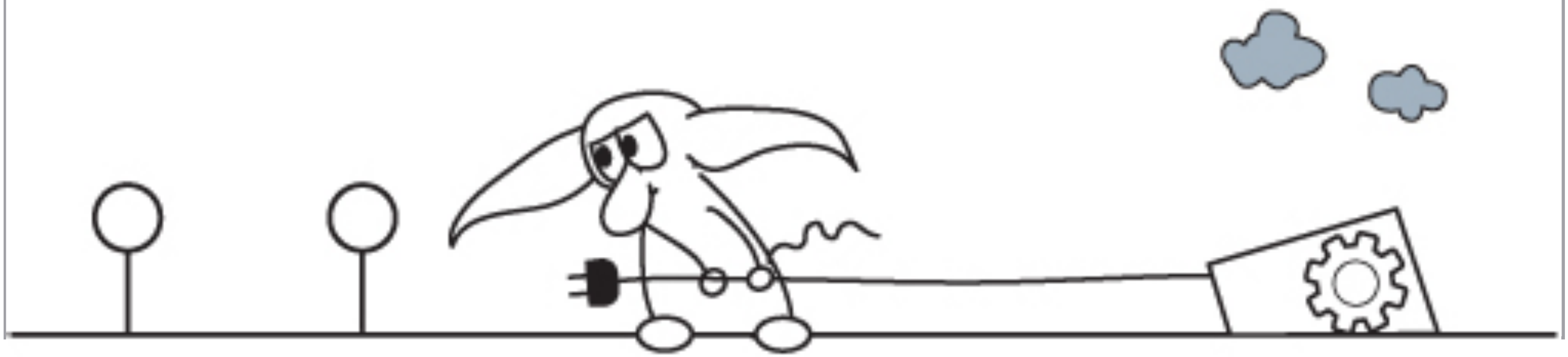


- It's "easy" for an individual to create a strong password
  - Just do something unique
- It's much harder to get everyone to be unique

Dictionary Attacks

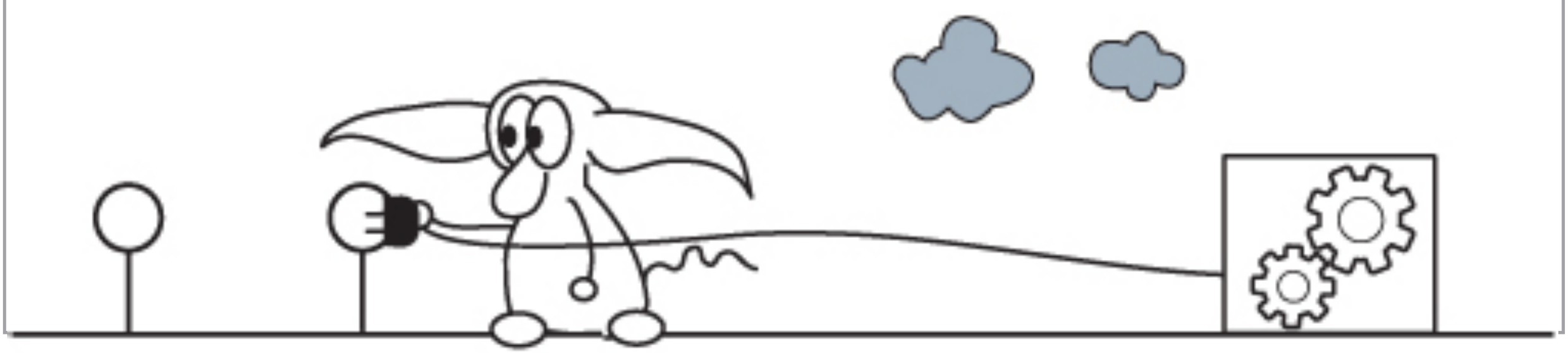
- Take a dictionary word
- Mangle it to your heart's content





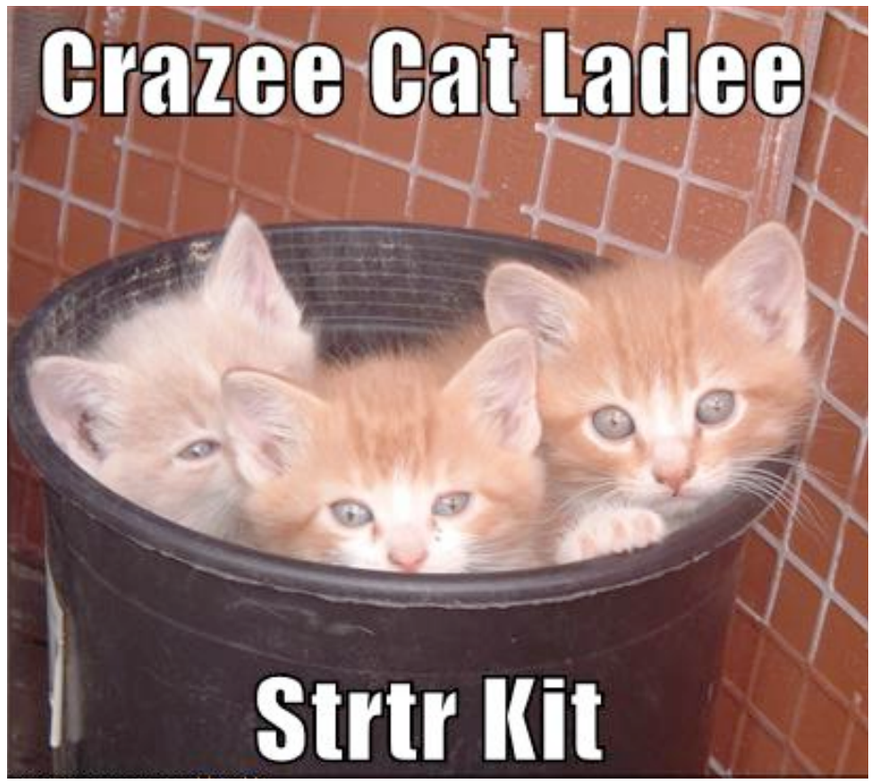
Reasons A Dictionary Attack can Fail

- You didn't try the right dictionary word
- You didn't try the right word mangling rule



Choosing an Input Dictionary … or 40

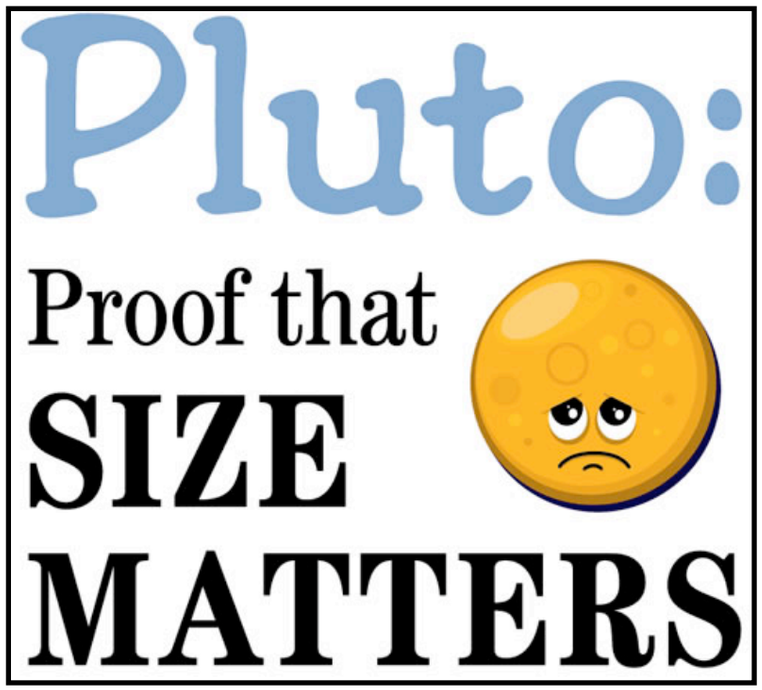
- People tend to go a bit overboard collecting input dictionaries
- After a while it starts to resemble brute force



If No Password Creation Rules Were Enforced

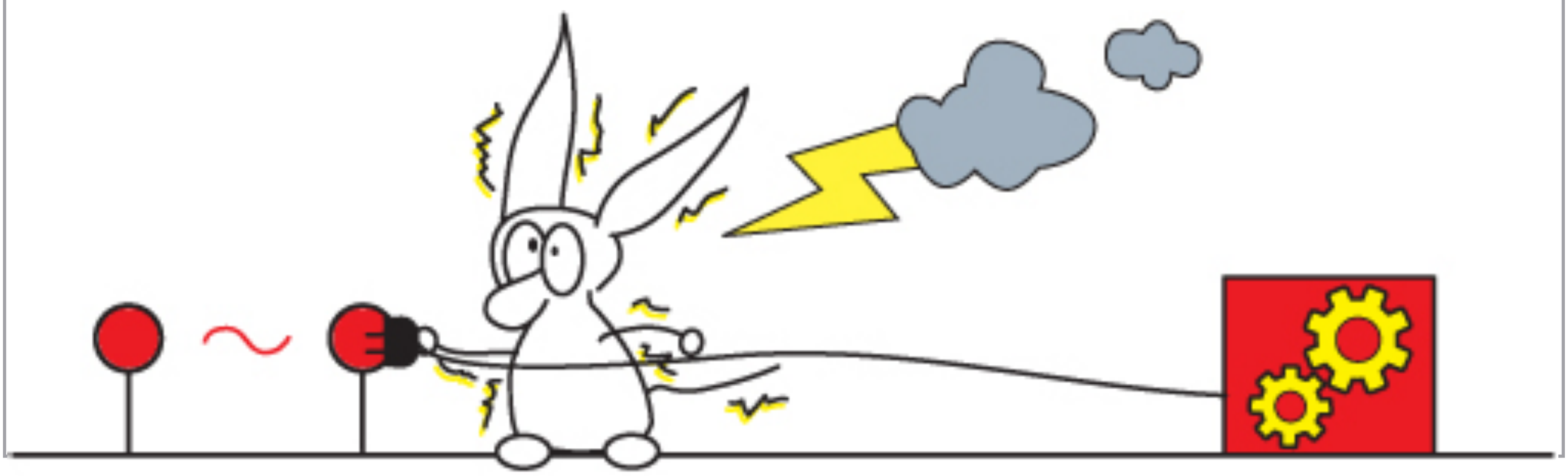
- Larger input dictionaries are better
- Check out a wordlist made from every wiki article, at Sebastien Raveau's blog

- http://blog.sebastien.raveau.name/



When there was a Password Creation Policy

- Smaller more targeted wordlists are better
- The best are based on previously cracked passwords



Word Mangling Rules

- Learn new ones from previously cracked passwords
- I’ve made some of my JtR rules available for download
- Minga also posted some online

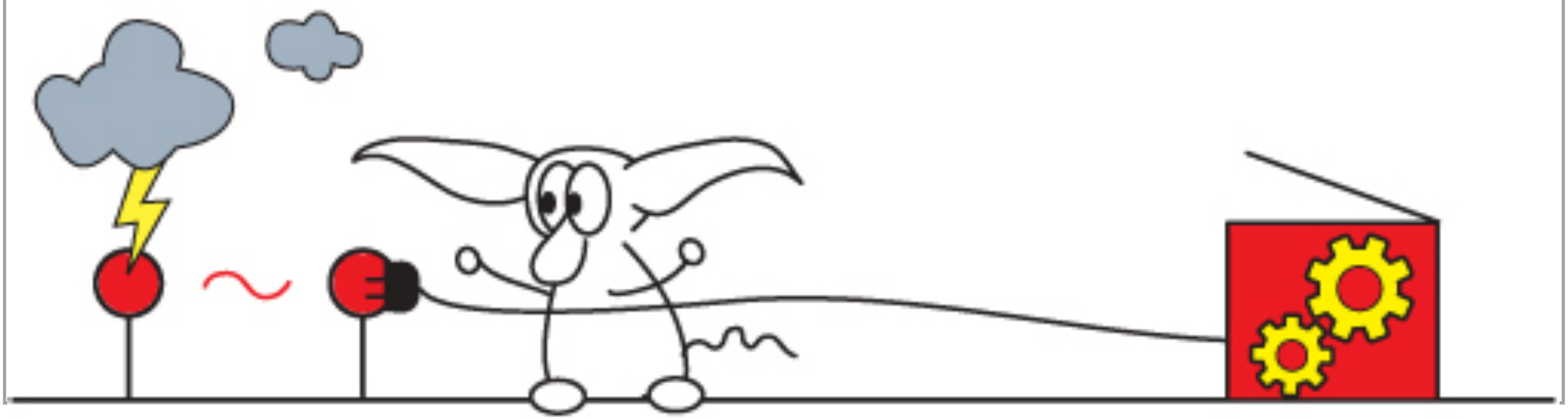
- http://marc.info/?l=john-users&m=123820850908275&w=2
- http://marc.info/?l=john-users&m=124053430313891&w=2

Probabilistic Cracking

- Some words are more likely than others
  - password, monkey, football
- Some mangling rules are more likely than others
  - 123, 007, $$$, Capitalize the first letter

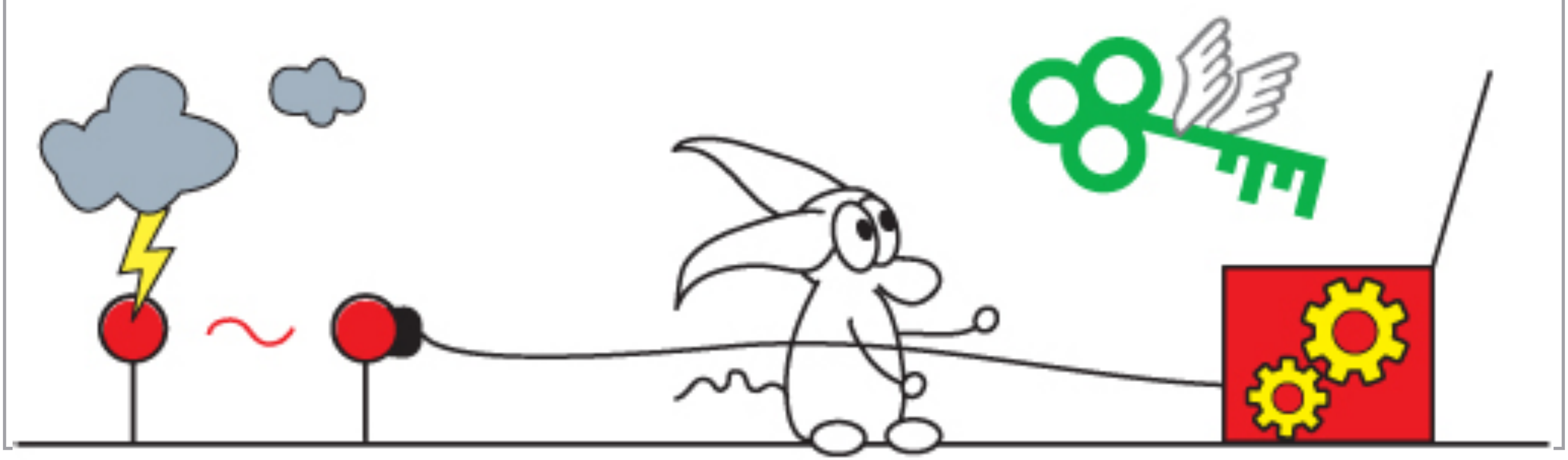
Which Should We Try First?

- A common word with an uncommon mangling rule?
  - 13!password13!
- An uncommon word with a common mangling rule?
  - zebra123

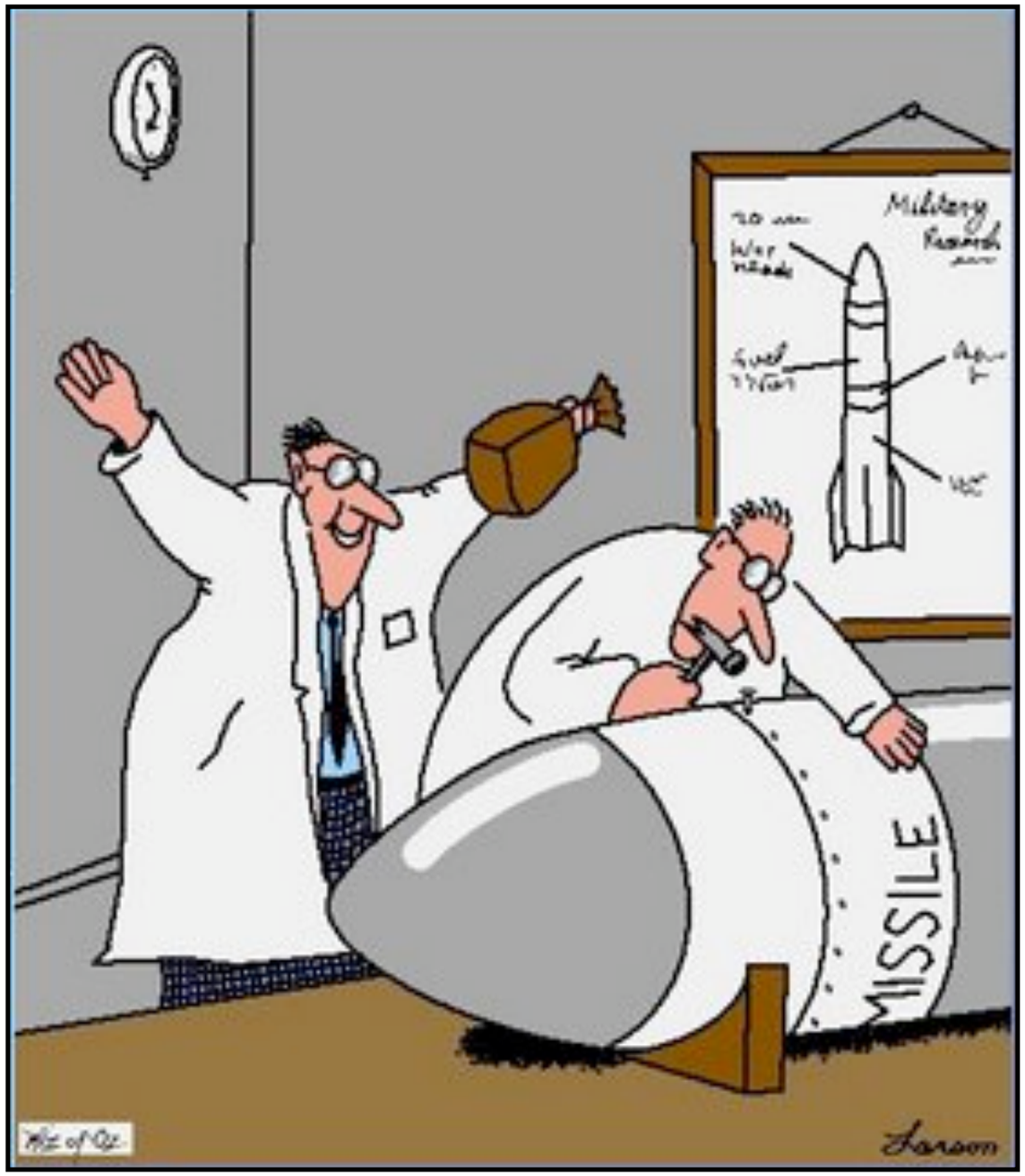


Our Probabilistic Cracker

- Assigns a probability to just about everything
  - dictionary words
  - word mangling rules
  - specific replacements, aka two digits go to “12”

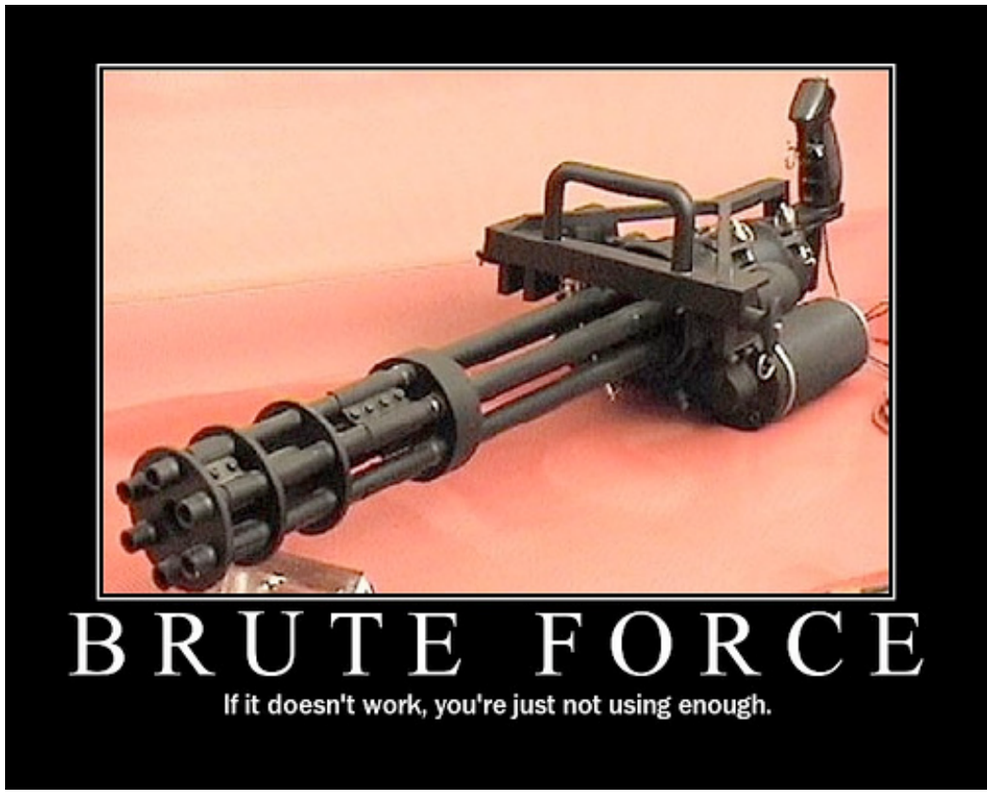


Time for a Quick Demo



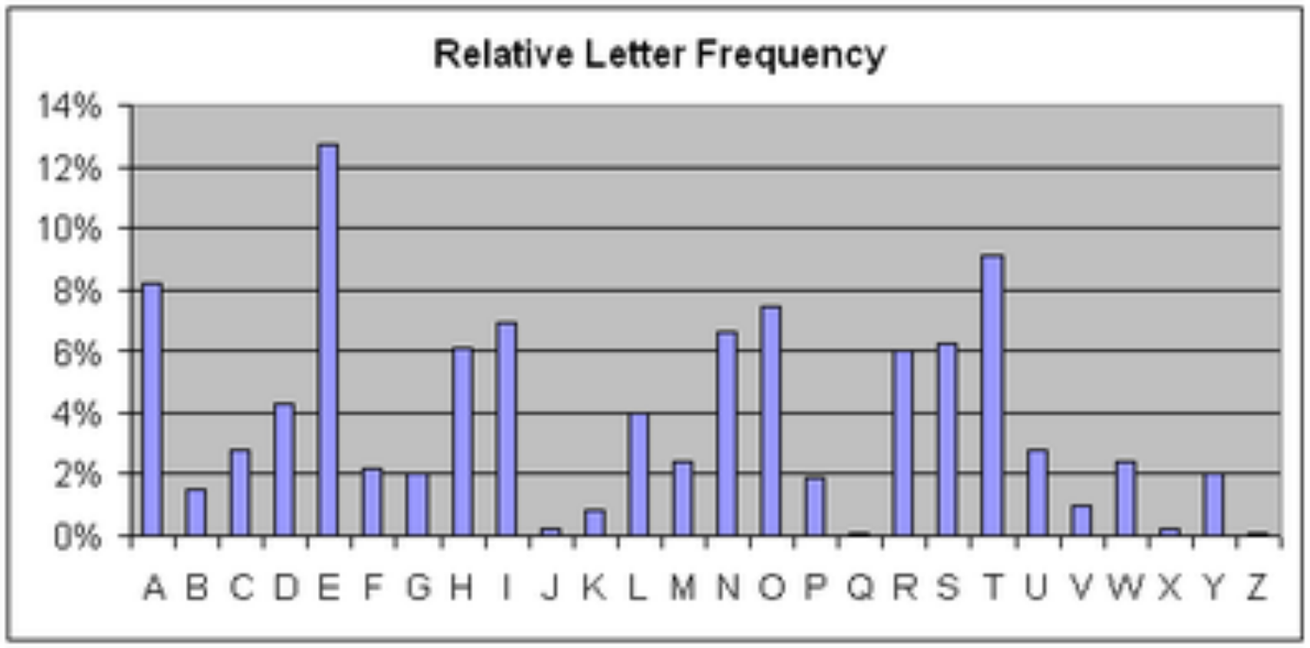
Brute Force

- Like most things, really powerful if you're not stupid about it



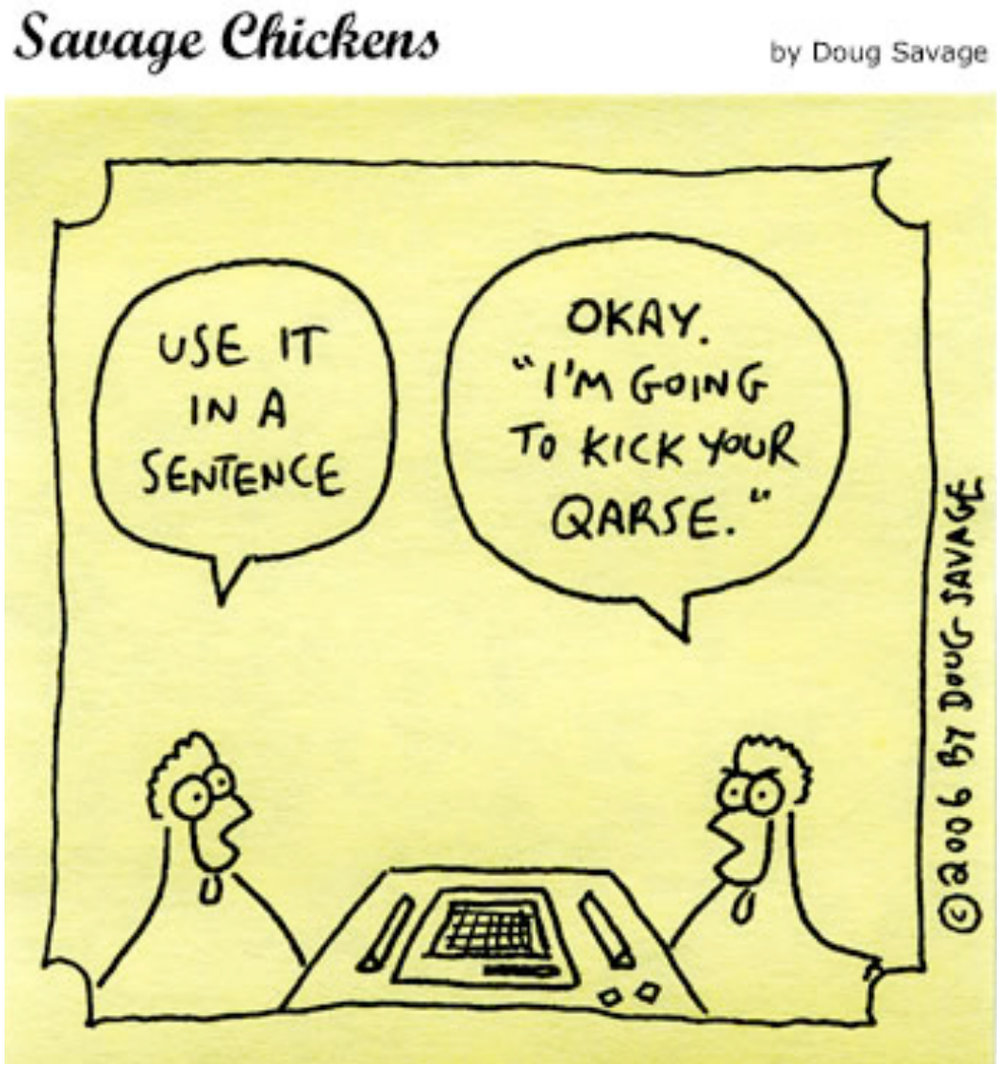
Letter Frequency Analysis

- The most basic brute-force optimization
- Very useful for figuring out what letters/ symbols not to try



Markov Models

- Conditional probability of letters
- Brute forces “human like” words
- Used in JtR’s Incremental mode



Targeted Brute Force

- People tend to capitalize the first letter
- They generally put numbers at the end of passwords
- For this case, people liked using the words “php” or “phpbb” in their passwords
- Check out the new version of the tool ‘Crunch’
  - Check the programming forum on the remote-exploit.com

An Example:

./john -incremental=Alpha -stdout -session=t1 |

An Example:

./john -incremental=Alpha -stdout -session=t1 |

Create guesses using JtR's Markov models.

bara	bony	stace	marine
sandy	bool	steve	maring
shanda	boon	stevy	marian
sandall	stark	stech	mariah
starless	start	steck	marley
dog	stack	sanda	marler

An Example:

| ./middleChild -cap first -append s1d1 |

An Example:

| ./middleChild -cap first -append s1d1 |

Capitalize the first letter
Add a special character and digit to the end

Bara!1	Sandy!1	Shanda!1	Sandall!1
Bara!2	Sandy!2	Shanda!2	Sandall!2
Bara!0	Sandy!0	Shanda!0	Sandall!0
Bara!3	Sandy!3	Shanda!3	Sandall!3
Bara!4	Sandy!4	Shanda!4	Sandall!4
.....	.....	.....	.....

An Example:

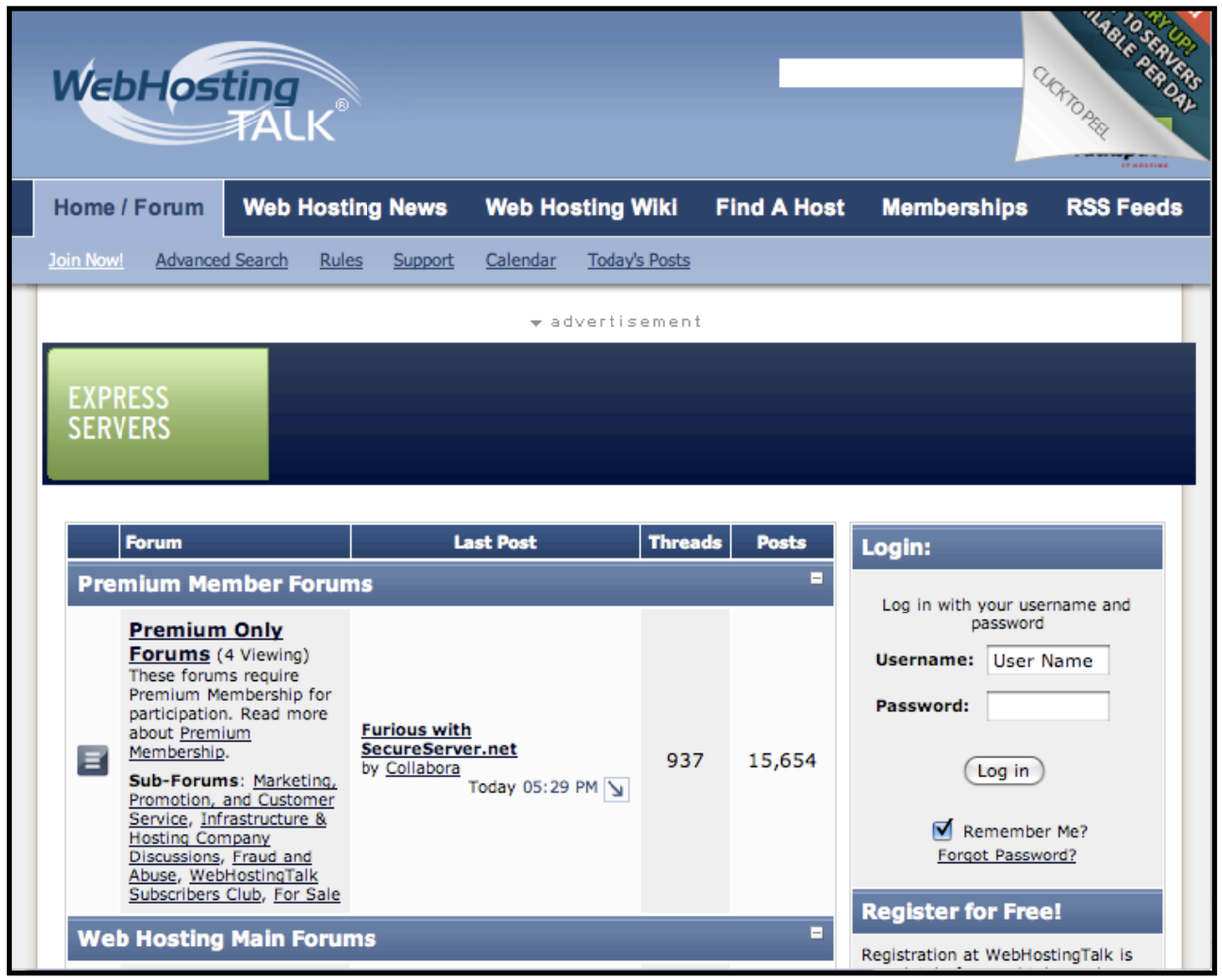
```
| ./john -stdin -hash=raw-MD5 ./hashes.txt
```

An Example:

| ./john -stdin -hash=raw-MD5 ./hashes.txt

Now pipe everything back into JtR so we can actually try to crack the hashes

Cracking the Web Hosting Talk List



- Originally hacked March 21st, 2009
- Over 200k salted hashes were stolen

Don’t Worry Though...

“Passwords are hashed with salt. It would be an unprecedented event to reverse engineer our passwords. I change my password periodically though, so maybe today is a good day for that.”

- SoftWareRevue
  iNET Community Coordinator

Oh, and....

"Absolutely no credit card or PayPal data was compromised."

- SoftWareRevue
  iNET Community Coordinator

Stuff Happens...

- Web Hosting Talk was compromised again by the same hacker on April 7th
- The hacker posted 202k password hashes + 2,218 credit card numbers

I want to make this clear

- People get hacked. I'm not blaming Web Hosting Talk for that.
- Getting someone out of your system once they compromised it is also a tough problem.
- What I do have a problem with is Web Hosting Talk downplaying the risks that their users faced

One Interesting Fact

- Number of users who changed their password after the first attack
  - 1348
  - That’s less than 1% of the total
  - 0.6% to be exact

So How Unbreakable is This Hash?

- First we need to figure out what the forum software is
- Google “Web Hosting Talk Forum Software”

“Yes. It's vBulletin.”

- SoftWareRevue
  iNET Community Coordinator

What Hashing Algorithm does vBulletin use?

- Google to the rescue again...

MD5(MD5(Password).salt)

So How Should We Test It?

1109 People used the password “password”

But what about the Salt

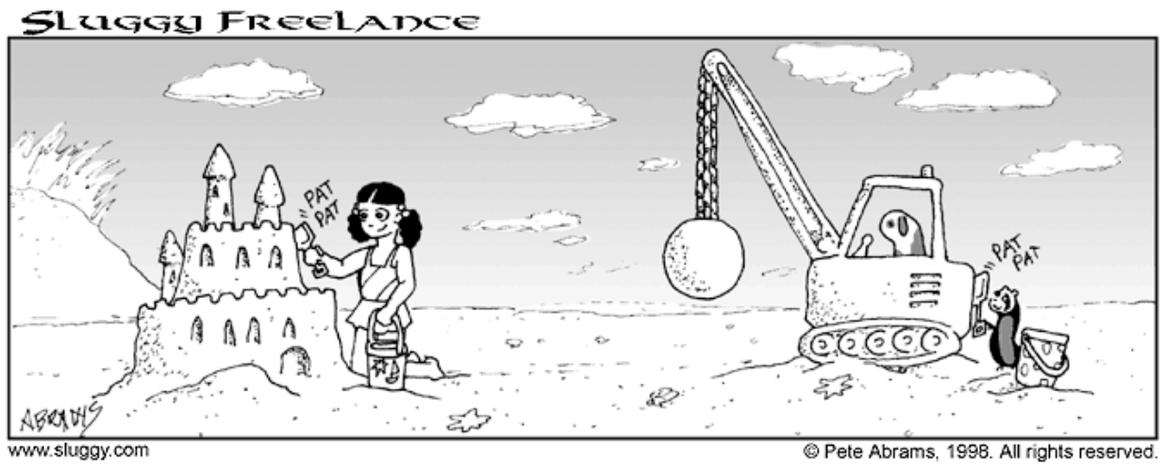
- It's a major problem
- Since each user's salt is different, we have hash each password guess for each user
- Compare it to the PhpBB attack
  - Assume you spent 1 hour attacking the PhpBB list
  - It would take you 200,000 hours to to run the same attack on the Web Hosting Talk list

That Being Said

- I’ve still managed to crack 34% of the passwords
- A majority of them were cracked using a list of previously cracked passwords from other sites, (no word mangling rules).
- Did you know people use the same password on more than one site?

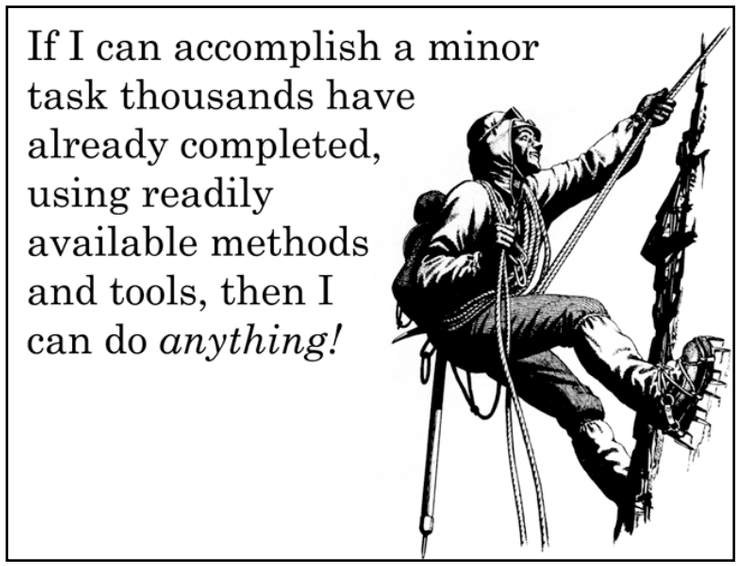
The Salt Doesn't Protect Individual Users

- Don't post your hash online claiming it is unbreakable
- It's possible to set up attacks to only target people with the words "admin" or "webmaster" in their e-mail address



Questions/Comments?

- My Research Blog
  - http://www.reusablesec.blogspot.com
- Tools Page
  - http://sites.google.com/site/reusablesec/
- E-Mail Address
  - weir@cs.fsu.edu



Dealing with Other Types of Passwords

- Note: The following slides were not covered in the actual talk due to time constraints

Cracking Pass-Phrases

- The main problem is we don't have many examples of pass-phrases
- One approach

1. Use an input dictionary of phrases
   - !!It's fun to try the impossible!

Cracking Pass-Phrases

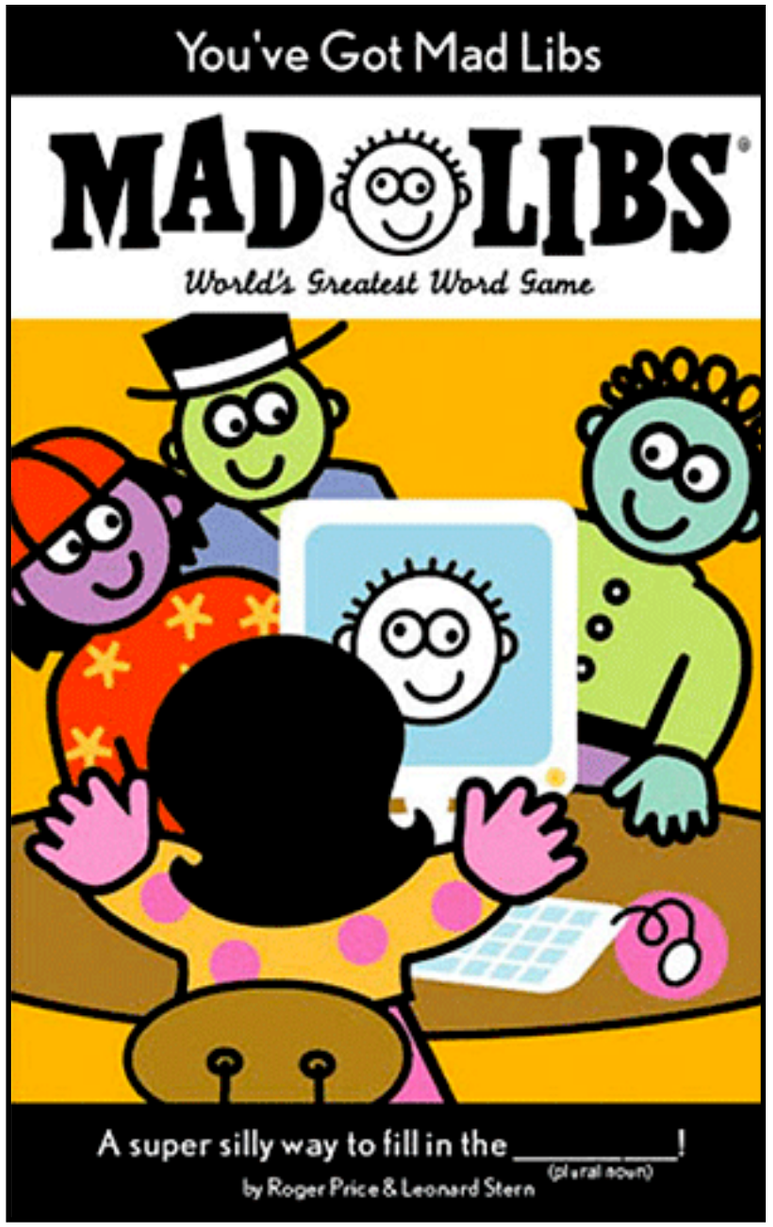
- The main problem is we don't have many examples of pass-phrases
- One approach

1. Use an input dictionary of phrases
   - !!ifttti!

Cracking Pass-Phrases

2. Use a Mad Libs Approach

   - Proper-Noun verbs a Noun
   - Proper-Noun loves Proper-Noun



Cracking Graphical Passwords

- People sometimes use ASCII art for their passwords
  - /><{{{{"> -- fish
  - //^\oo/^\\\ -- spider
  - d[o_0]b -- robot
  - (^_-) ~ ~ <===3 -- rocket ship?!

Solution....

- I've created some input dictionaries of ASCII art to use
- Probably the largest collection of NSFW ASCII art on the internet...

Targeted Attacks

- Assign higher probabilities to certain replacements
  - Kids names
  - Birth Years
  - Zip Codes

- Check out CUPP from the remote-exploit group

Perl Monks Statistics

- Disclosed in the ZF05 Data-set this Wednesday
  - Average Length: Also 7.2 characters long
  - 30% of them contained an UPPERCASE letter
  - Close to 8% of them contained a special character
  - 40% of them contained only lowercase letters