Testing of Password Policy

Anton Dedov

ZeroNights 2013

Who Am I

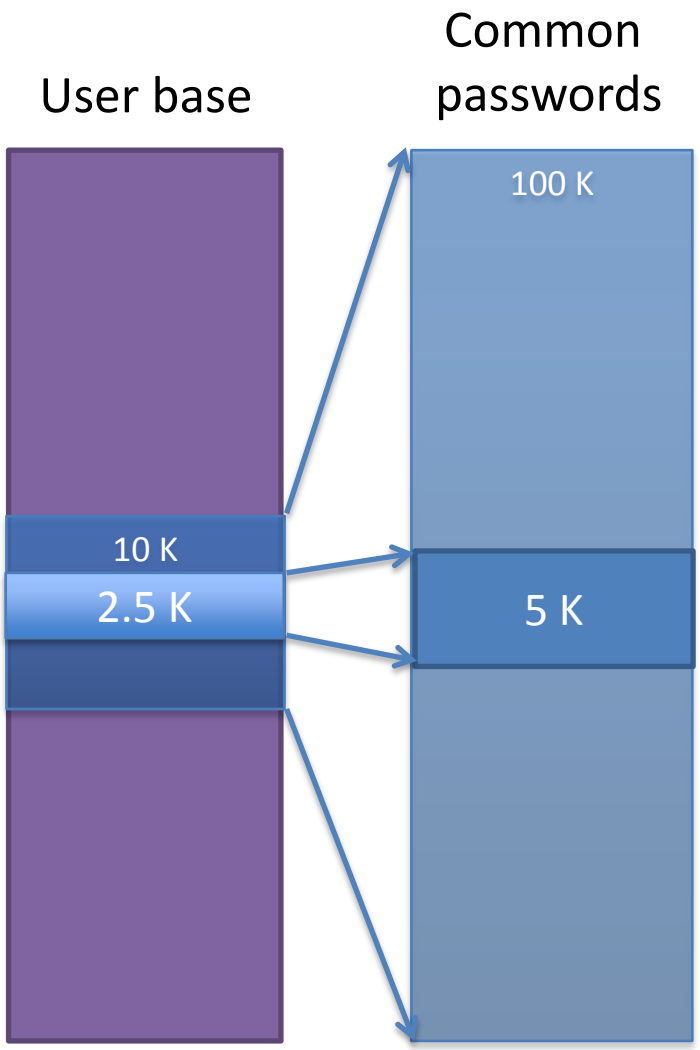
- Software Developer and Security Engineer @ Parallels Automation
- Open source developer
- Mail: adedov@gmail.com
- Twitter: @brutemorse

Motivation

- It is hard for application developers to choose between existing password meters reasonably.
- Worse, some implement their own [or customize existing] without understanding of security and psychological implications.
- Need some framework/criteria that would help reasonable choice.

NAÏVE SECURITY MODEL

Untargeted Online Attacks



- 1 guess per user / day
- 2 days to find first password
- 100 days to find 50 passwords

- 1 guess per user / day
- 10 days to find first password
- 1.5yr to find 50 passwords

Targeted Online Attacks

- 10 failed attempts → 1 hour block
- 240 attempts per user / day
- 7200 attempts per user / month
- **86400** attempts per user / year
- More IP-s scale linearly

Offline Attacks

- Huge dictionaries
- Specialized hardware and clusters
- No time/complexity limitations except
  - Enforced password quality
  - Hash speed
  - Salt uniqueness

TESTING PASSWORD METERS

Candidates

- Plesk
- jquery.complexify
- zxcvbn
- libpwquality
- passwdqc

Method

- Apply meters to password bases
- Dictionary attacks with JtR
- Rule-based attacks with JtR
- Collect essential parameters

Apply Meters

- **Requirement**: meter should provide unambiguous signal about if password is accepted or not.
- Passwdqc tells straight “OK” or “Bad”.
- Others return score. Minimal accepted score documented.

Password Bases

- Real customers
- RockYou all
- CMIYC-2010 not cracked
- Random passphrases
- Random 10-char passwords

Red for attacks; blue for psychological acceptance.

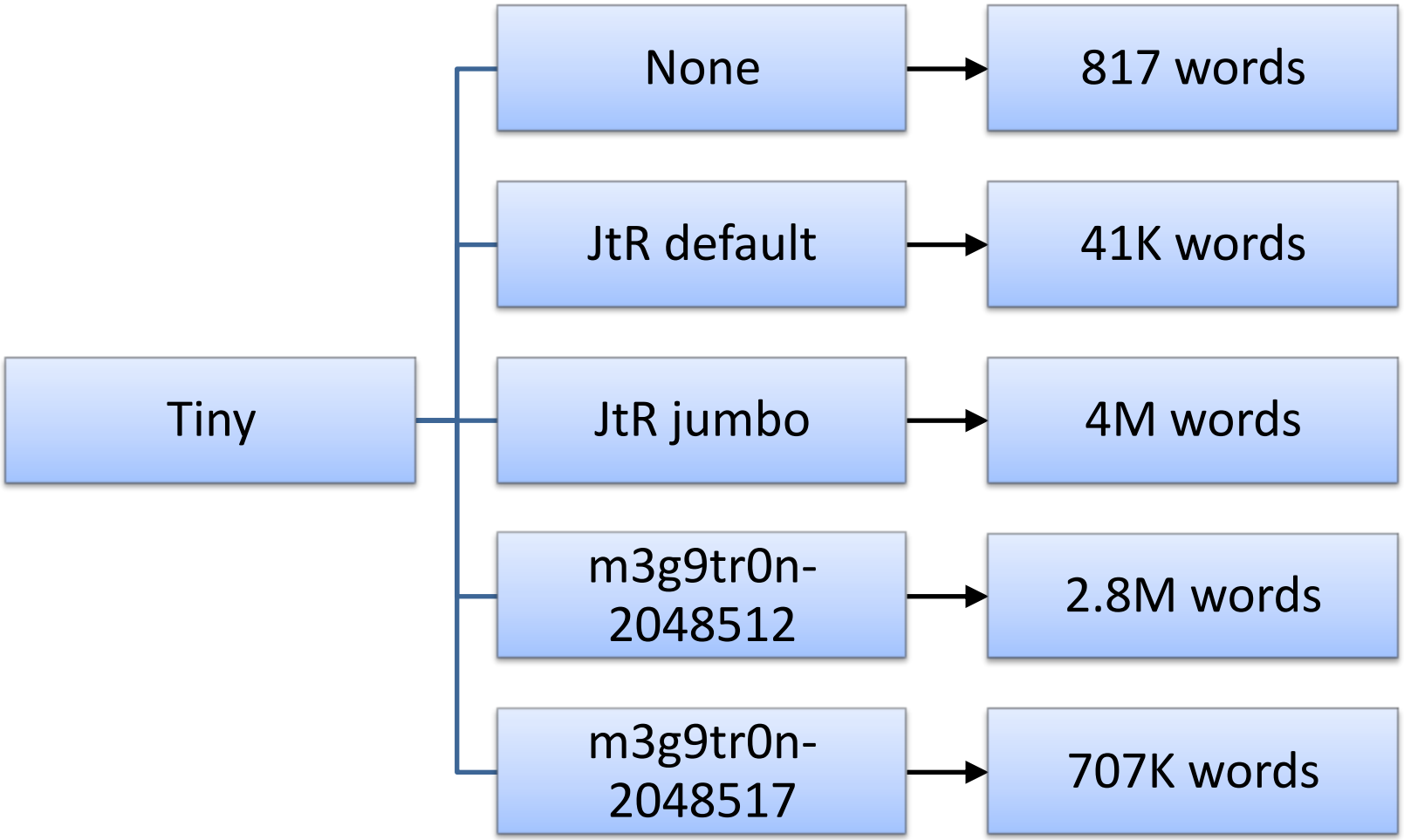
Dictionaries

Dictionary	Size, words
Tiny English	817
RockYou top	1438
Common-passwords	3546
English	54316
Tiny English crossed / 8 chars	72100

Rules

Rule	Factor
JtR defaults	~ 40
JtR jumbo	~ 5500
m3g9tr0n-2048512	= 3510
m3g9tr0n-2048517	~ 860

Cracking Sessions



Cracking Sessions

- **25 attacks** per **password base** per **meter**
- Min dictionary size 817
- Max dictionary size 396M

RockYou dictionary was not used against RockYou password base.

Parameters

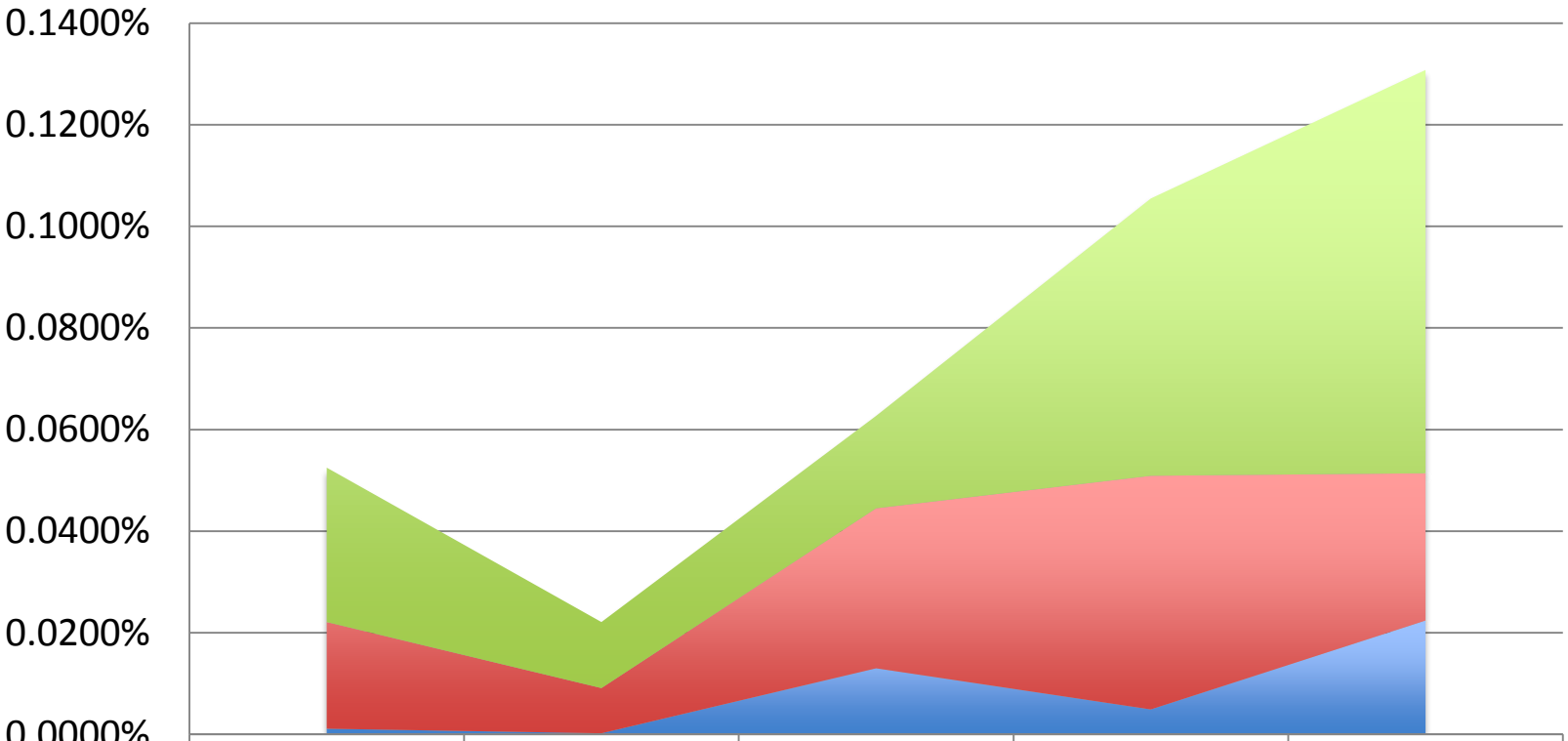
- M – passwords approved by meter
- D – attack dictionary size
- C – # of guessed passwords during attack
- Attack effectiveness $\frac{C}{M}$
- Attack economy $\frac{C}{D}$

Online Attacks Effectiveness

For dictionaries < 100K

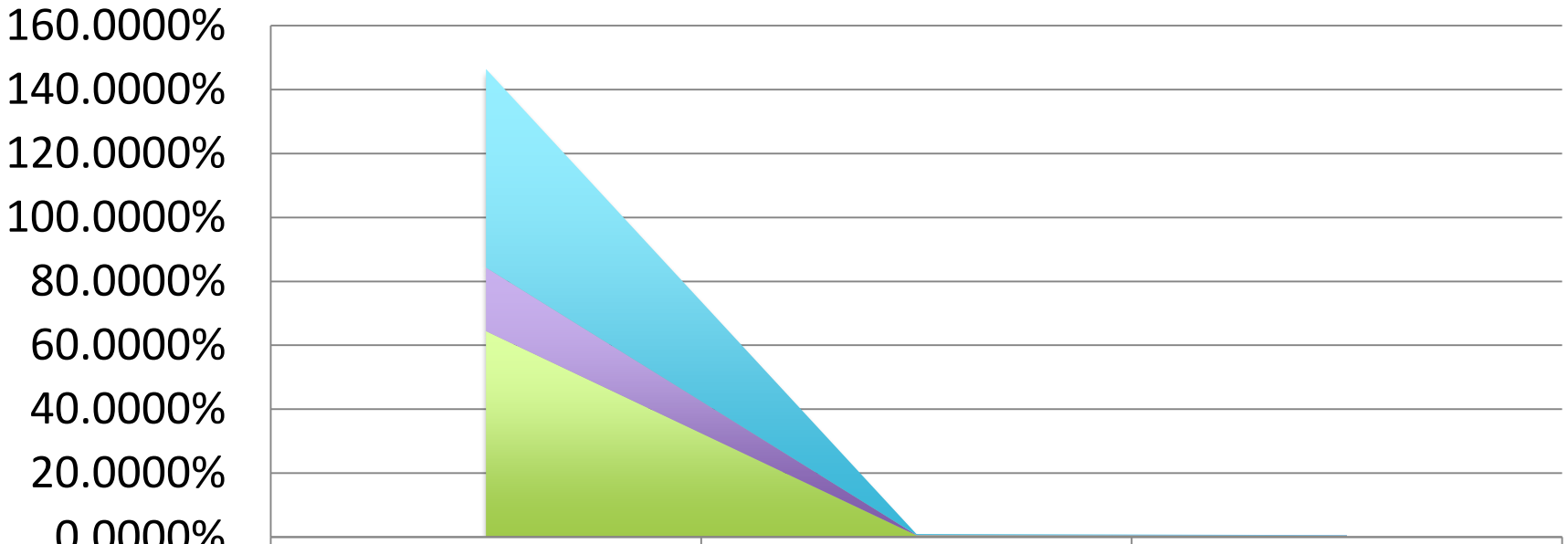
Max guess rate 0.007%

Max Attack Effectiveness



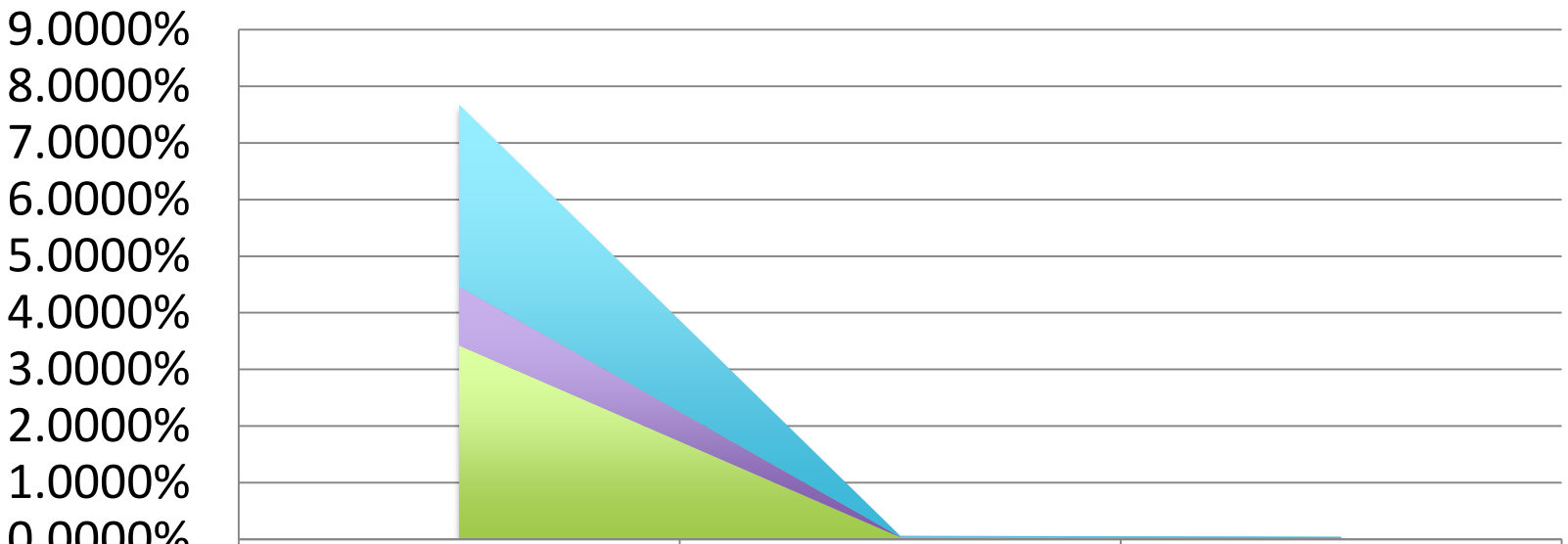
	passwdqc	plesk	zxcvbn	complexify	pwquality
customer2	0.0304%	0.0130%	0.0182%	0.0546%	0.0794%
customer1	0.0210%	0.0089%	0.0315%	0.0460%	0.0290%
rockyou	0.0011%	0.0002%	0.0130%	0.0049%	0.0224%

Max Attack Economy



	rockyou	customer1	customer2
pwquality	62.1545%	0.2782%	0.1224%
complexify	19.8816%	0.1224%	0.1224%
zxcvbn	64.1850%	0.2782%	0.1224%
plesk	0.1224%	0.1224%	0.1224%
passwdqc	0.1224%	0.1224%	0.1224%

Average Attack Economy

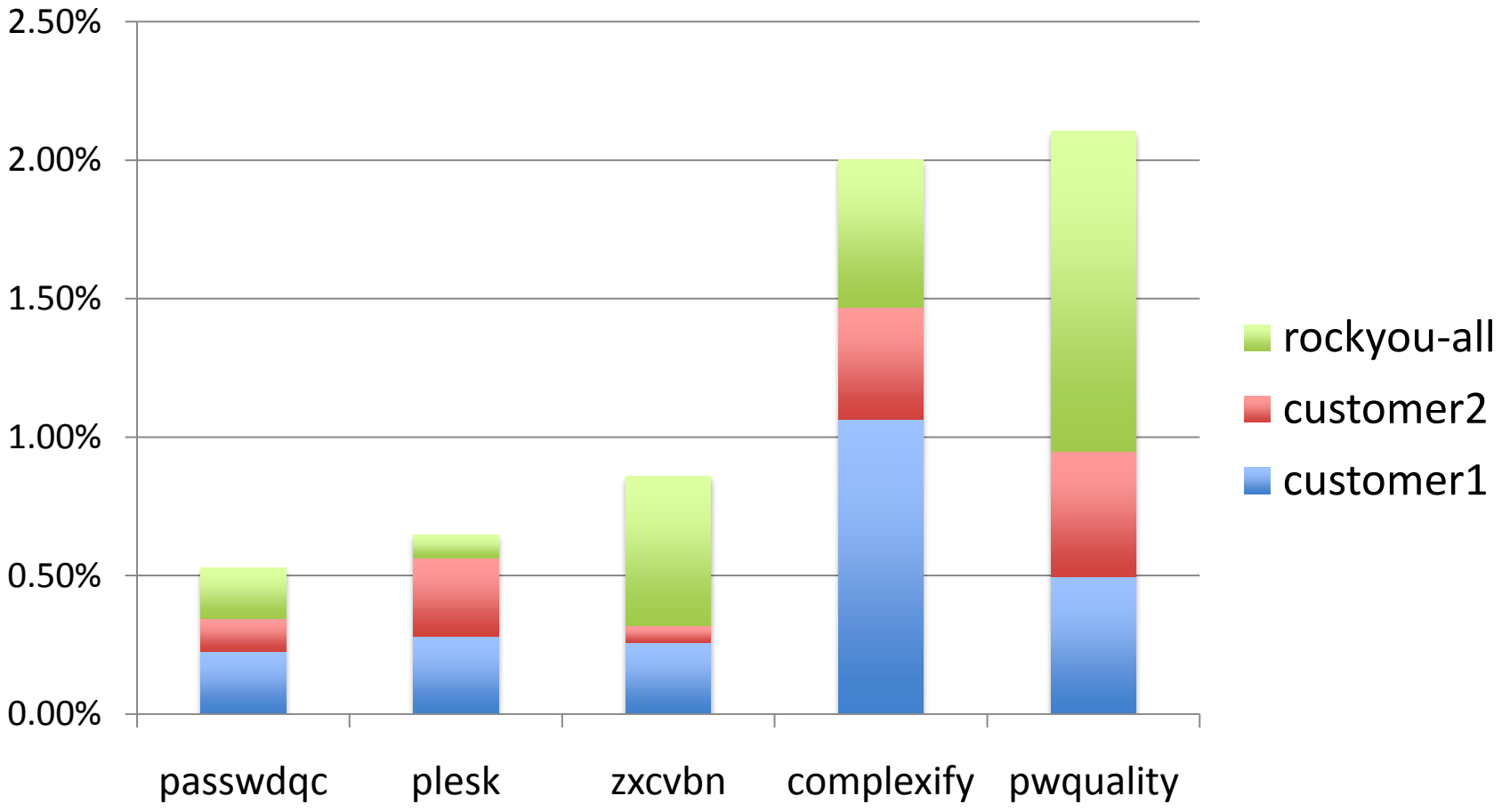


	rockyou	customer1	customer2
pwquality	3.2154%	0.0177%	0.0093%
complexify	1.0375%	0.0095%	0.0101%
zxcvbn	3.4033%	0.0180%	0.0096%
plesk	0.0079%	0.0092%	0.0092%
passwdqc	0.0137%	0.0092%	0.0094%

Guesses Totals

Meter	RockYou	Customer 1	Customer 2
plesk	0.08%	0.28%	0.28%
passwdqc	0.18%	0.23%	0.12%
zxcvbn	0.54%	0.26%	0.06%
complexify	0.54%	1.06%	0.40%
libpwquality	1.16%	0.50%	0.45%

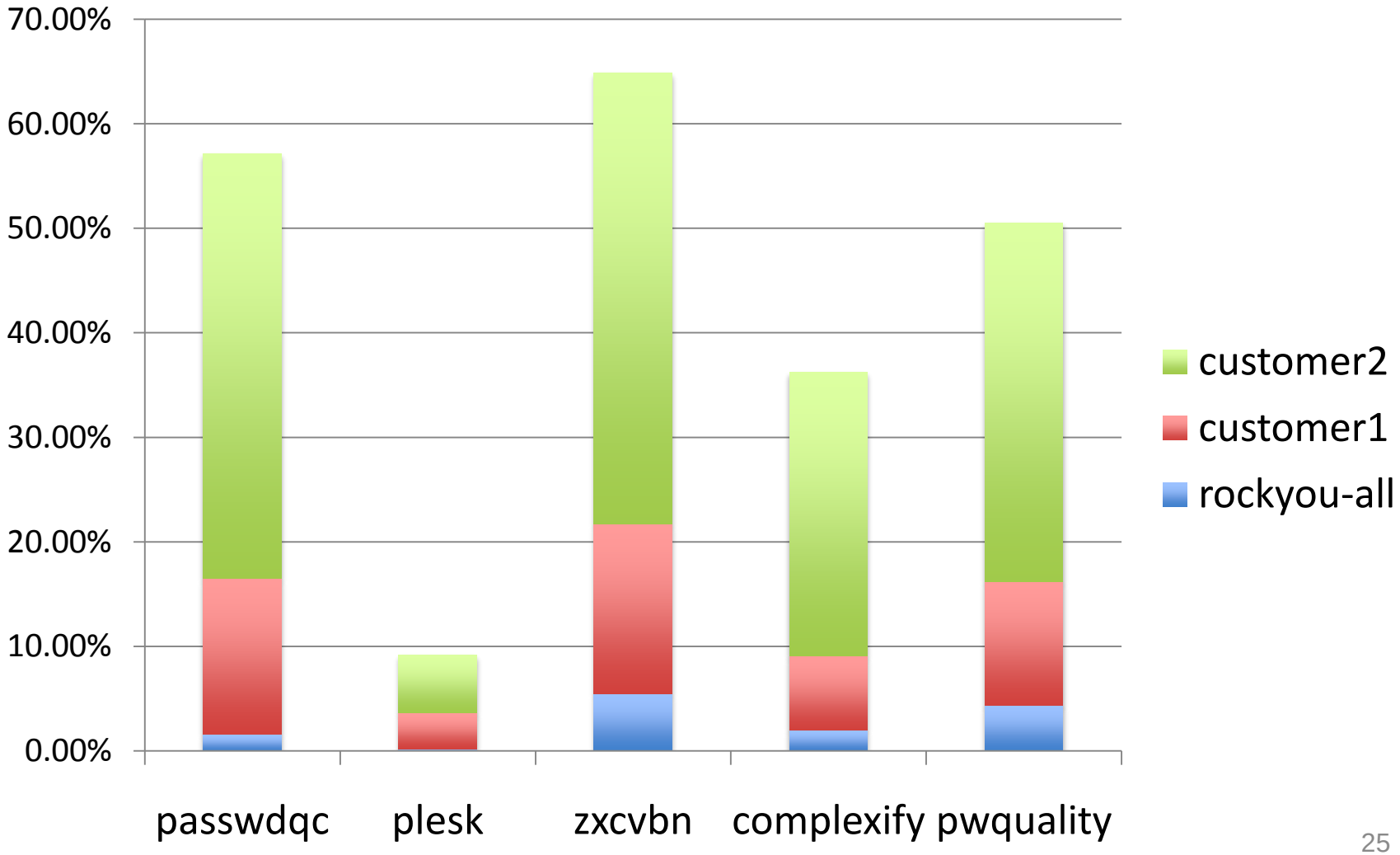
Guesses Totals



Psy. Acceptance: User Passwords

Meter	RockYou	Customer 1	Customer 2
plesk	0.21%	3.45%	5.53%
passwdqc	1.60%	14.90%	40.62%
zxcvbn	5.43%	16.29%	43.16%
complexify	2.03%	7.05%	27.18%
libpwquality	4.32%	11.88%	34.27%

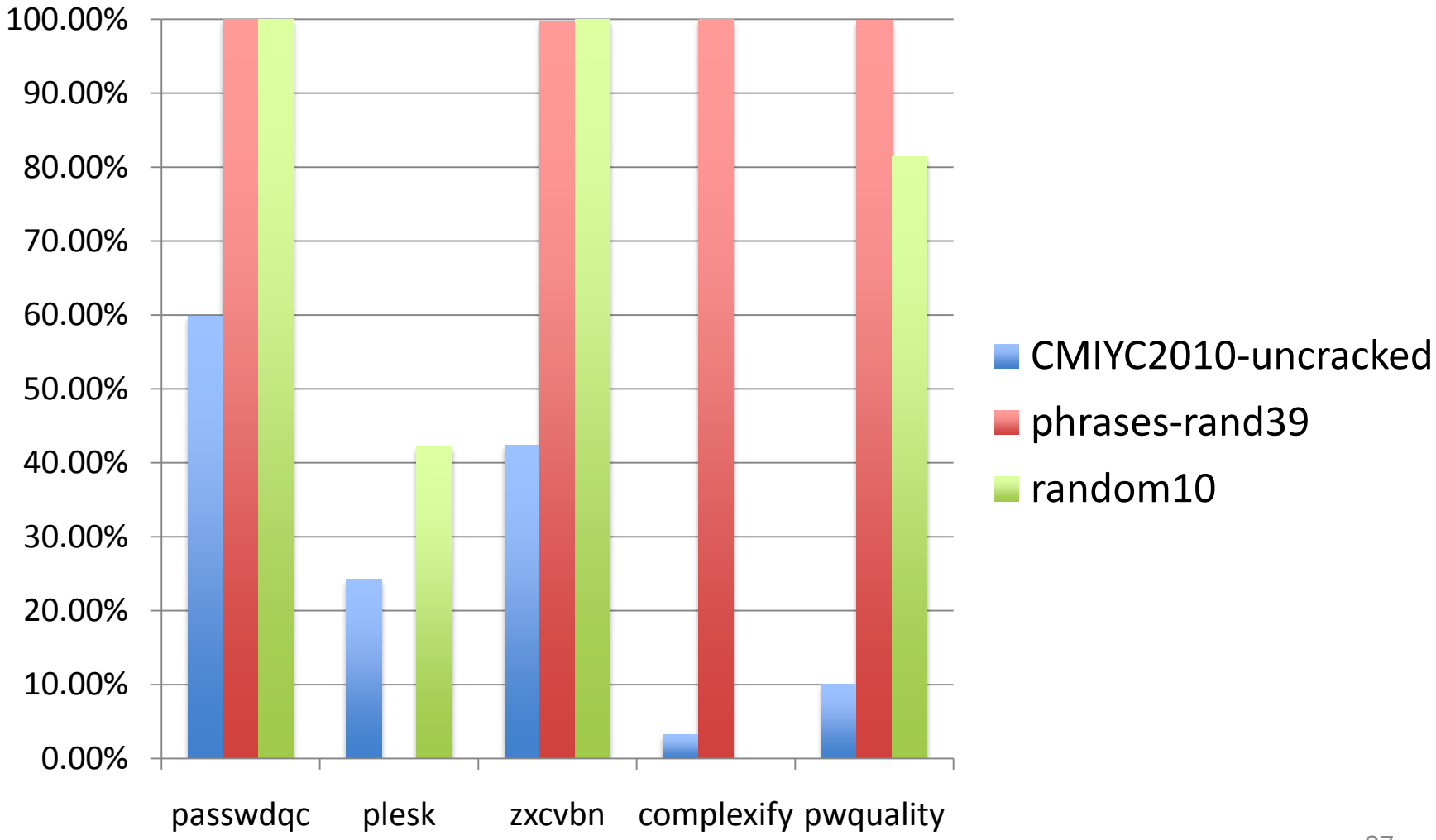
Psy. Acceptance: User Passwords



Psy. Acceptance: Hard Passwords

Meter	CMYIC-2010	Pass-Phrases	Random 10 chars
plesk	24%	0%	42%
passwdqc	59%	99.98%	100%
zxcvbn	42%	99.76%	99.99%
complexify	3%	99.94%	0%
libpwquality	10%	99.82%	81%

Psy. Acceptance: Hard Passwords



The “editors” choice

Security	Psychology
passwdqc	zxcvbn
plesk	passwdqc
zxcvbn	libpwquality
jquery.complexify	jquery.complexify
libpwquality	plesk

Conclusions

- Test your security tools for security
- Avoid write your own security tools
- All tested meters protect from online attacks
- Also *seem* protect from offline attacks
  (for slow hashes and unique salts)
- But *most* tend to deny more passwords than it is necessary, including known to be hard ones
- **Passwdqc** and **zxcvbn** look best

Where to go?

- Bigger dictionaries and brute force
- Testing on real people to
  - Learn evolution of “common passwords” lists
  - Test psychological acceptance empirically
- More meters?

Special thanks

Alexander Peslyak
Solar Designer

Bonus: time to process RockYou…
(MBP 2011)