Lost in Disclosure: On The Inference of Password Composition Policies

Saul A. Johnson[1], Joao F. Ferreira[*], Alexandra Mendes[†], Julien Cordry[1]

[1] School of Engineering, Computing and Digital Technologies, Teesside University

[*]Instituto Superior Técnico, University of Lisbon and INESC-ID, Lisbon, Portugal

[†]HASLab, INESC TEC and Computer Science Department, University of Beira Interior, Covilhã, Portugal

A (Very) Brief Introduction!

I'm Saul, a password security researcher at Teesside University in the UK, working mainly with formal methods for password security.

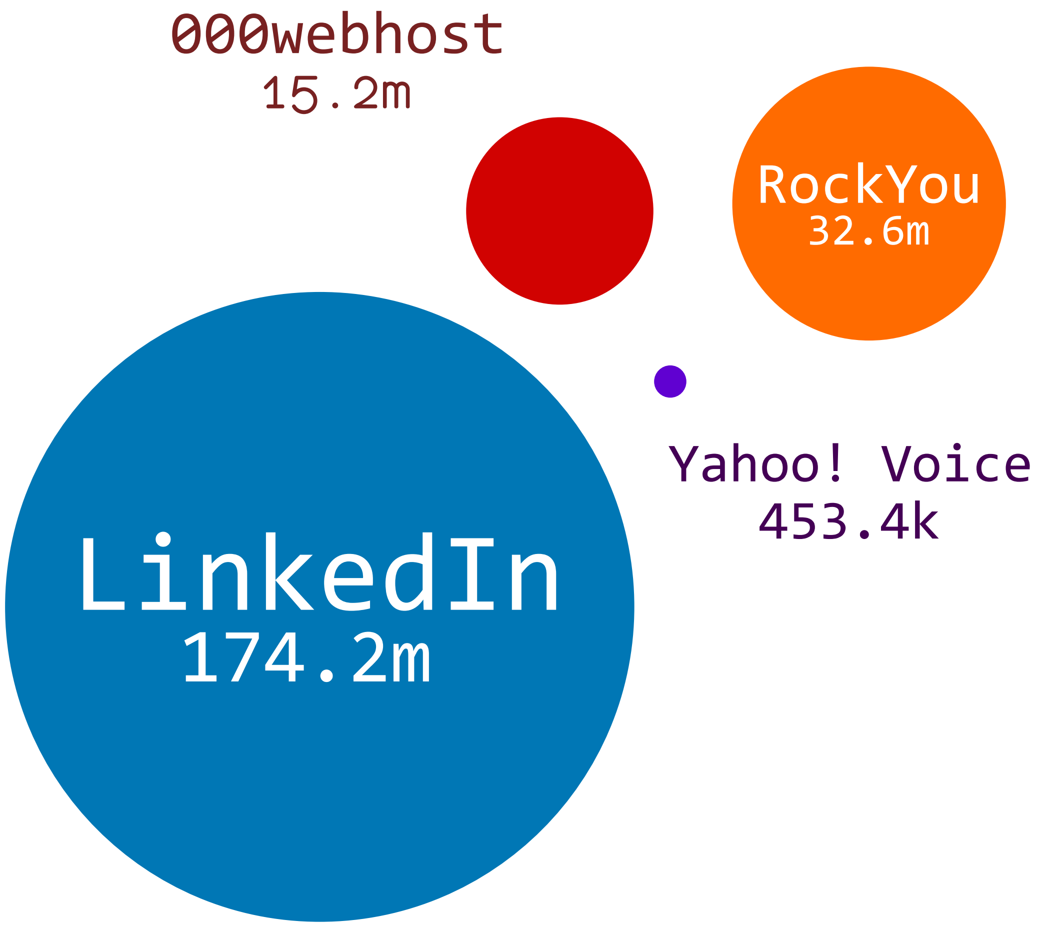
GitHub: @lambdacasserole

Twitter: @lambdacasserole

Web: https://sauljohnson.com

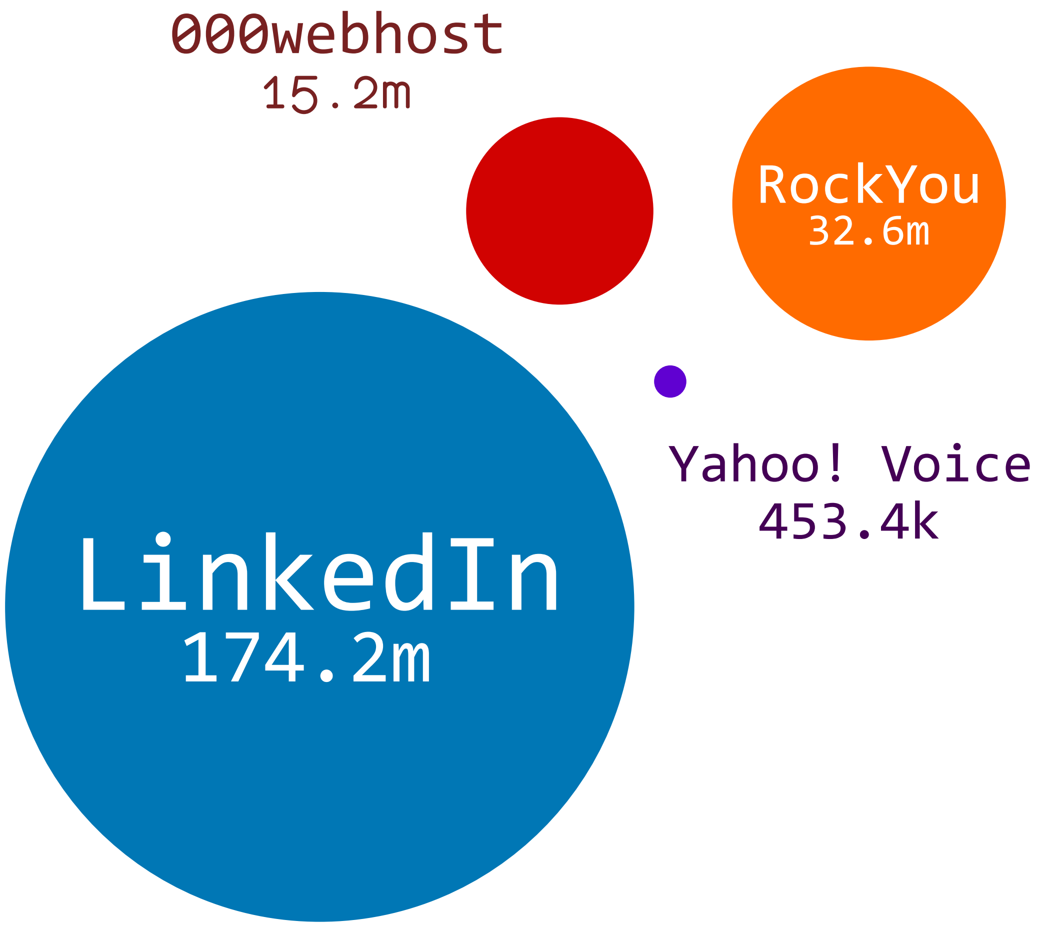
User Credential Data Breaches

- Hundreds of millions of usernames and passwords (credentials) are compromised from websites every year and leaked online.[1]
- Very often these passwords are either not hashed at all (i.e. plaintext) or hashed using a weak algorithm (e.g. MD5).



User Credential Data Breaches (cont.)

- On the right here are just 4 of these, to scale:
  - Yahoo! Voice[2]
  - 000webhost[3]
  - RockYou[4]
  - LinkedIn[5]
- This data, though compromised by criminals, can be used to improve password security through research!



Improving Password Security

- We can nudge users towards creating more secure passwords using password composition policies.[6]
- These are sets of rules that constrain which passwords users are permitted to select.
- The datasets on the right are shown next to the password composition policies they were created under.

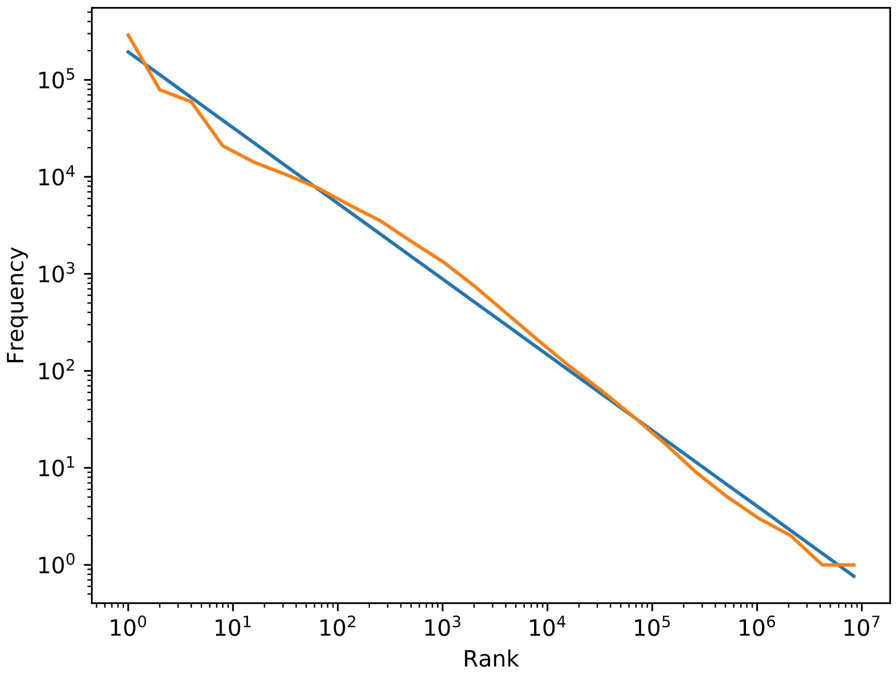
Dataset	Policy
RockYou	$length \geq 5$
Yahoo! Voice	$length \geq 6$
000webhost	$length \geq 6 \land digits \geq 1$
LinkedIn	$length \geq 6$

Password Policies and Security

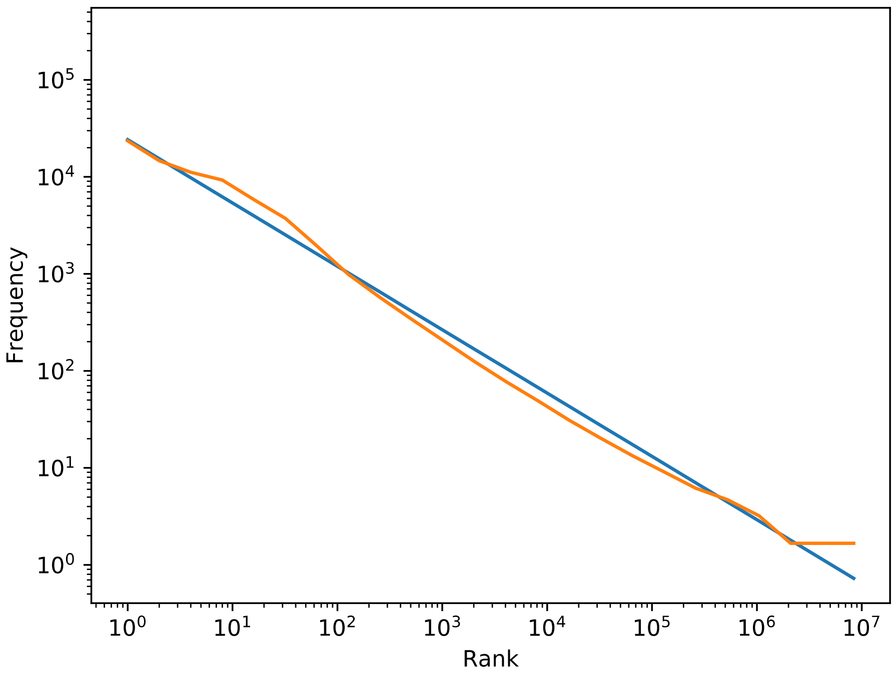
- So, do password composition policies improve user password security?
- We can find out, by:
  - Looking at password quality in real-world breached datasets for which we know the policy[1]
  - Or running lab studies[6] where users create passwords under different policies (ecological validity issues/expensive!)

Better Policy, Better Security!

Weaker Policy: Steeper Curve/Less Uniform Distribution (Length 5)



Stronger Policy: Shallower Curve/More Uniform Distribution (Length 6, 1 Digit)



But what if we don’t know the policy?

- If we don’t know the policy, we could, of course, just ask the organisation involved what it is.
- Alternatively, we could check their website and attempt to deduce password rules by trying to create an account.[7]
- These approaches can have their problems however…

Why not just ask?

Organisation might be on lockdown…

- Very often, the last thing an organisation in full damage control mode wants to do is talk about internal security decisions.
- They might accidentally incriminate themselves by revealing poor practice! GDPR makes this more likely.

…or gone entirely!

- The singles.org Christian dating website had a data breach, then ceased operations.[8]
- We can't ask them about password composition policies if they don't exist anymore!

Password Attributes

- We can imagine a password composition policy rule as a constraint on some attribute α, which is a function mapping passwords to natural numbers:

$$\alpha : Password \rightarrow \mathbb{N}$$

- Some example attributes are shown on the right here.

Attribute (α)	Description
$length(pwd)$	Length of password.
$words(pwd)$	Words (letter sequences separated by non-letters) in password.
$lowers(pwd)$	Lowercase letters in password.
$uppers(pwd)$	Uppercase letters in password.
$digits(pwd)$	Digits in password.
$symbols(pwd)$	Non-alphanumeric characters in password.
$classes(pwd)$	Character classes (lowers, uppers, digits, symbols) in password.

Inference: From Dataset to Policy

- The naïve approach here would just be to look for e.g. the shortest password in the dataset. Surely this should give us minimum password length?
- Unfortunately not, datasets like this are ‘noisy’. There are old passwords, test accounts etc. that make this approach infeasible![9]

Dataset	Compliant	Noncompliant
RockYou	32,524,461	78,587 (0.24%)
Yahoo! Voice	444,942	8,550 (1.89%)
000webhost	14,936,872	334,336 (2.19%)
LinkedIn	172,409,689	18549 (0.01%)

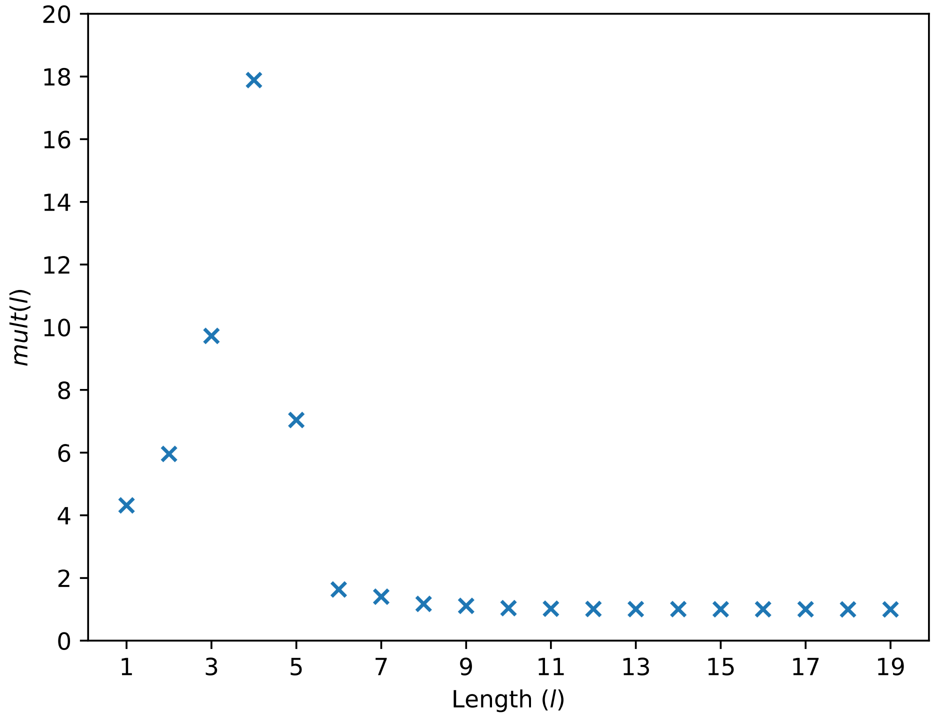
Inference: From Dataset to Policy (cont.)

- By converting our problem to one of outlier detection, we can get much more accurate results.
- We first map our chosen attribute function α over our dataset and construct a cumulative frequency series.
- We then plot the multipliers needed to reach the next cumulative frequency…

l	$f(l)$	$cum(l)$	$mult(l)$
1	314	314	4.32
2	1,042	1,356	6.00
3	6,725	8,081	9.72
4	70,506	78,587	17.89
5	1,326,965	1,405,552	7.03
6	8,488,412	9,893,964	1.64
7	6,288,016	16,181,980	—

Table 1: Frequencies $f(l)$ of passwords of different lengths l in the RockYou set, alongside their cumulative frequencies $cum(l)$ and the multiplier $mult(l)$ required to reach the cumulative frequency of the next length $cum(l+1)$.

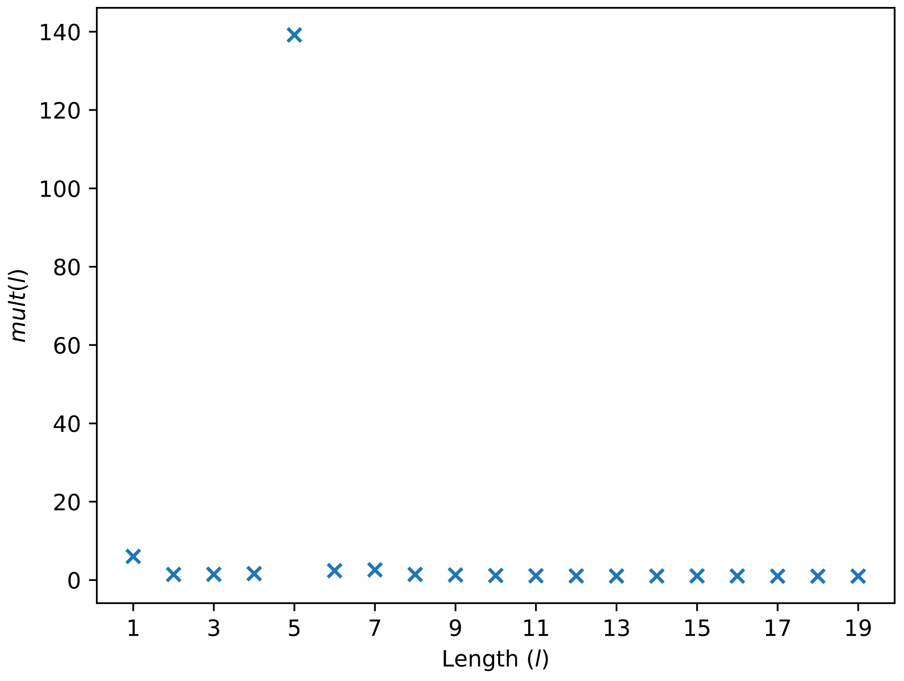
Inference: From Dataset to Policy (cont.)



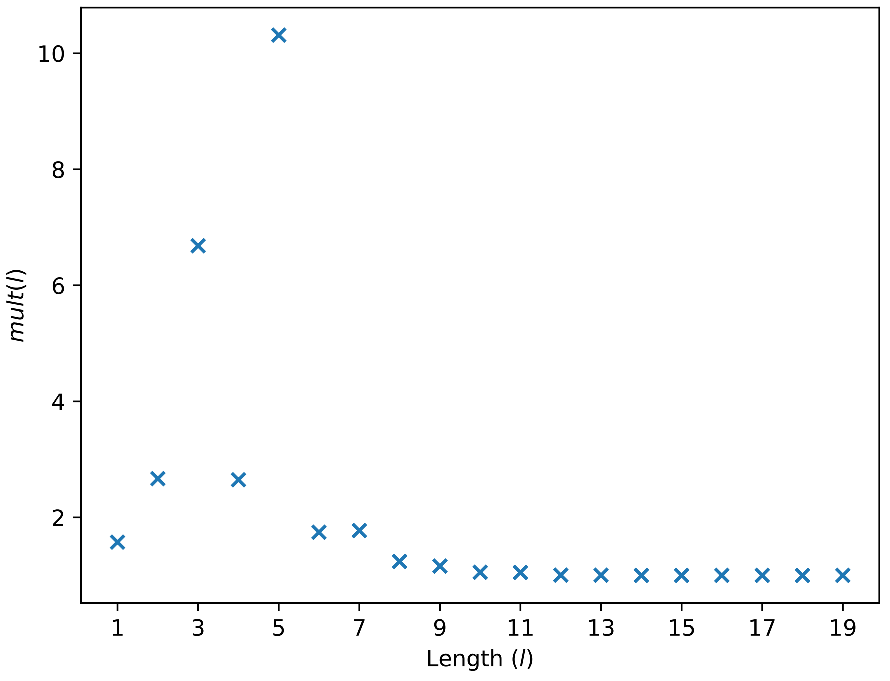
- Visualising this, we can clearly see our "big jump" outlier here.
- To get from the cumulative frequency of passwords up to length 4 to that of 5, a substantial multiplier is needed.
- Although more users have length 6 passwords (≈8m) than length 5 (≈1m) we have still correctly inferred this rule!

Some more results!

000webhost: Inferred minimum length of 6 (correct)

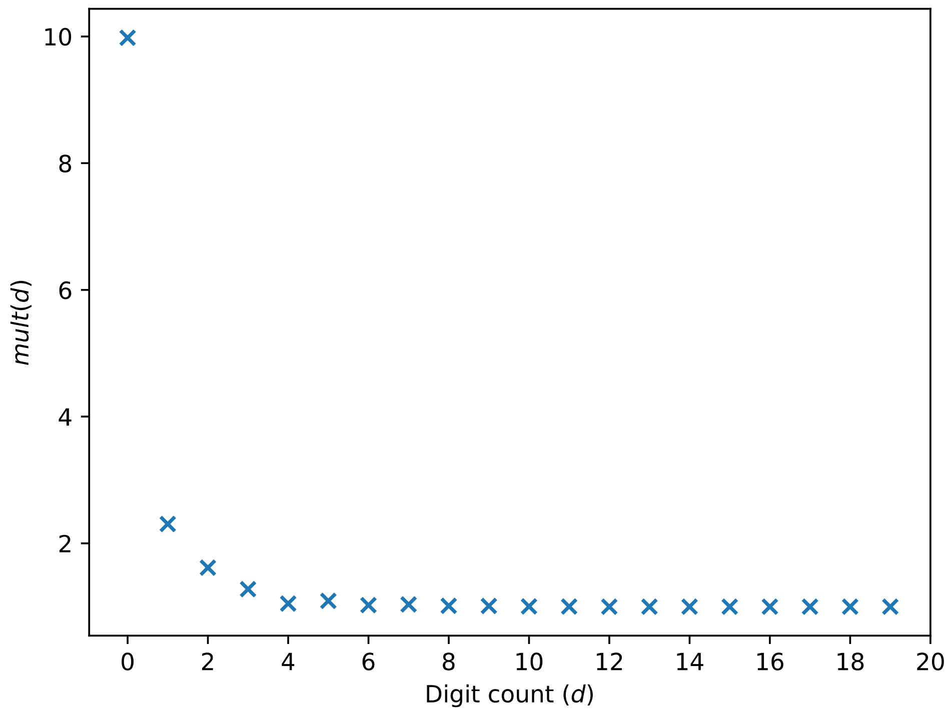


Yahoo!: Inferred minimum length of 6 (correct)

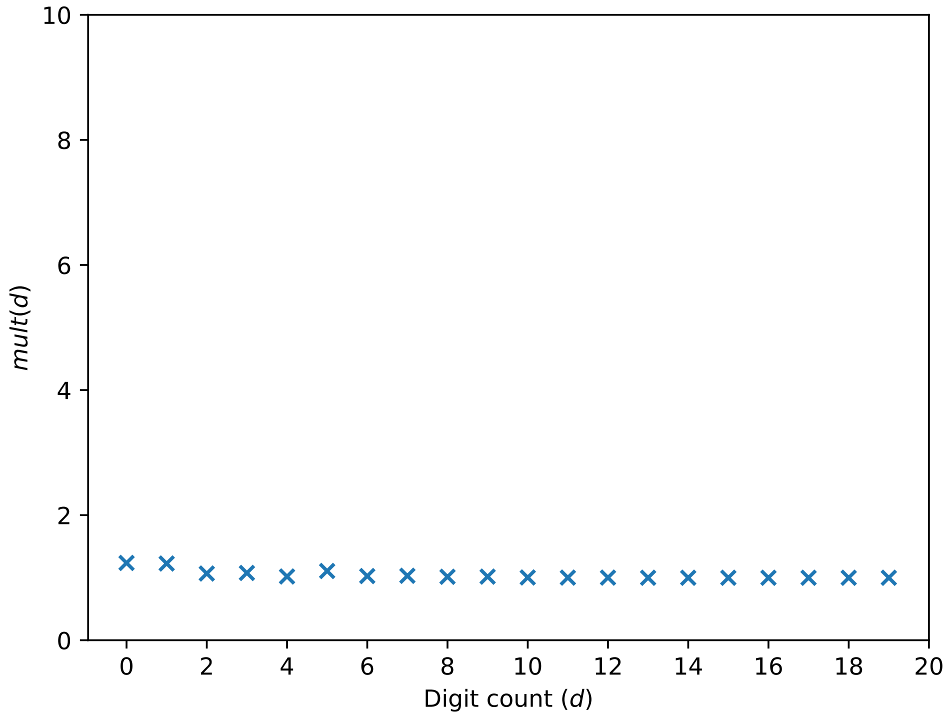


We're not limited to length, either!

- For example, if we swap our attribute α for a function that gives the number of numeric digits d in a password, we can infer constraints on that!
- The 000webhost mandates at least 1 digit in passwords, giving us this spike in $mult(d)$ at $d = 0$.



Inferring the Absence of Constraints



- By setting a threshold on what we consider an ‘outlier’ we can also infer the absence of constraints.
- RockYou, for example, had no requirement for digits in passwords, meaning all multipliers were very low (see left).

Why should we care?

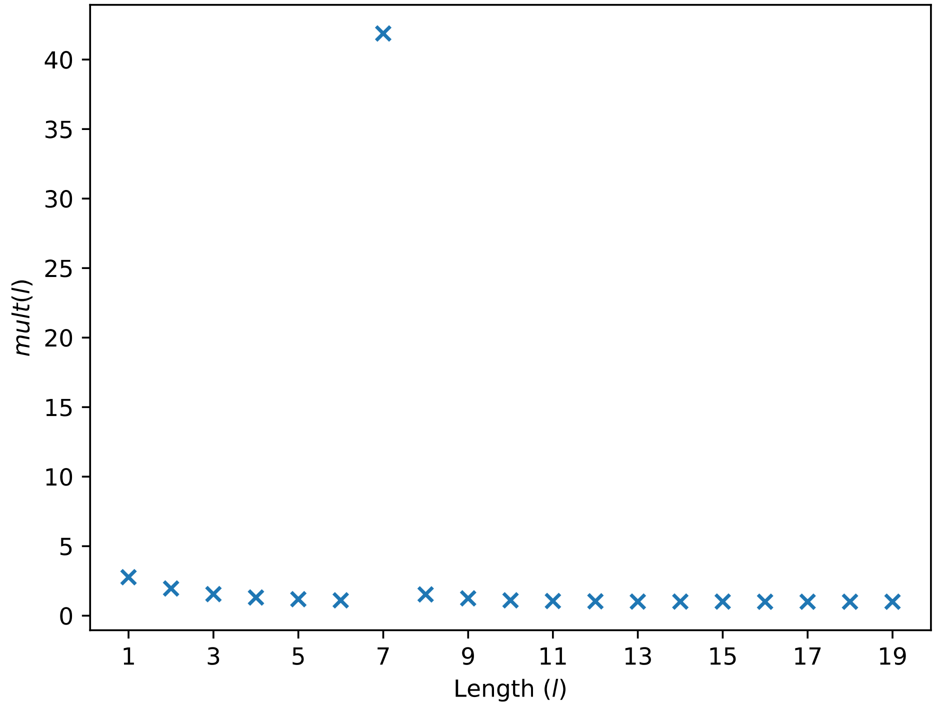
- For password data breaches for which the policy is not known, it is now possible to attempt to easily infer it!
- We're applying this in our research now, to increase the quality of the datasets we're using in our work by filtering out non-password artefacts.

Saving us from bungled data!

- After the data has been compromised, the party responsible may run processing scripts on it to e.g. change its file format for easy resale.
- This can introduce non-password artifacts into the data if, for example, passwords containing spaces are split into more than one record.

Saving us from bungled data! (cont.)

- We filtered the LinkedIn dataset according to a *2class8*[10] policy (at least 8 characters long, at least 2 character classes) and intentionally introduced errors.
- Passwords were split along commas/spaces, creating 404,547 extra records.
- We were able to use our approach to recover the original *2class8* policy.

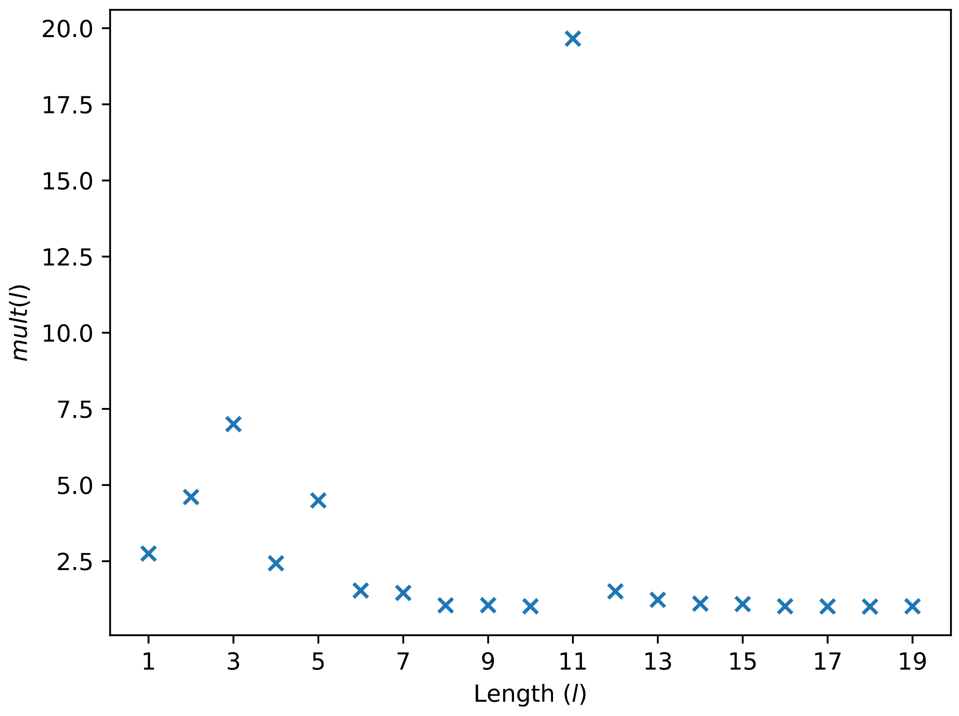


Detecting padded data!

- The size of a password data breach (i.e. the number of records it contains) often dictates the price cybercriminals are able to obtain for it.
- For this reason, such data may be padded with other password data from elsewhere to artificially inflate its value.

Detecting padded data! (cont.)

- Using the LinkedIn dataset filtered for *2word12* instead, we intentionally padded it with several smaller data breaches:
  - Elitehacker ($n = 1{,}000$)
  - Hak5 ($n = 2{,}987$)
  - Singles.org ($n = 16{,}248$)
  - Faithwriters ($n = 9{,}709$)
- Again, our approach permitted recovery of the *2word12* policy.



Our Tool: *pol-infer*

- We built a tool that implements this methodology called *pol-infer*.
- All scatter plots shown in this talk were generated using it!
- Here's the GitHub link: https://github.com/sr-lab/pol-infer

References

1. Matt Weir, Sudhir Aggarwal, Michael Collins, and Henry Stern. 2010. Testing metrics for password creation policies by attacking large sets of revealed passwords. In Proceedings of the 17th ACM conference on Computer and communications security (CCS '10). ACM, New York, NY, USA, 162-175. DOI: https://doi.org/10.1145/1866307.1866327

2. Doug Gross. 2012. Yahoo hacked, 450,000 passwords posted online – cnn. Online https://edition.cnn.com/2012/07/12/tech/web/yahoo-users-hacked (Accessed on 04/10/2019).

3. Charlie Osborne. 2015. 000webhost hacked, 13 million customers exposed — zdnet. Online https://www.zdnet.com/article/000webhost-hacked-13-million-customers-exposed/ (Accessed on 04/10/2019).

4. Nik Cubrilovich. 2009. Rockyou hack: From bad to worse — techcrunch. Online https://techcrunch.com/2009/12/14/rockyou-hack-security-myspace-facebook-passwords/ (Accessed on 04/10/2019).

5. Matt Burgess. 2016. Check if your linkedin account was hacked — wired uk. Online https://www.wired.co.uk/article/linkedin-data-breach-find-out-included (Accessed on 07/26/2019).

6. Saranga Komanduri, Richard Shay, Patrick Gage Kelley, Michelle L. Mazurek, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Serge Egelman. 2011. Of passwords and people: measuring the effect of password-composition policies. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11). ACM, New York, NY, USA, 2595-2604. DOI: https://doi.org/10.1145/1978942.1979321

7. Peter Mayer, Jan Kirchner, and Melanie Volkamer. A second look at password composition policies in the wild: Comparing samples from 2010 and 2016. In Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017). Santa Clara, CA: USENIX Association, 2017, pp. 13–28.

8. Darren Pauli. 2009. Exposed web site a reminder for use of multiple passwords — network world. Online https://www.networkworld.com/article/2263760/exposed-web-site-a-reminder-for-use-of-multiple-passwords.html (Accessed on 07/25/2019).

9. Patrick Gage Kelley, Saranga Komanduri, Michelle L. Mazurek, Richard Shay, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Julio Lopez. 2012. Guess Again (and Again and Again): Measuring Password Strength by Simulating Password-Cracking Algorithms. In Proceedings of the 2012 IEEE Symposium on Security and Privacy (SP '12). IEEE Computer Society, Washington, DC, USA, 523-537. DOI: https://doi.org/10.1109/SP.2012.38

10. Richard Shay, Saranga Komanduri, Adam L. Durity, Phillip (Seyoung) Huh, Michelle L. Mazurek, Sean M. Segreti, Blase Ur, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2016. Designing Password Policies for Strength and Usability. ACM Trans. Inf. Syst. Secur. 18, 4, Article 13 (May 2016), 34 pages. DOI: https://doi.org/10.1145/2891411