Closing the Gap: The difference between Full Knowledge and Practice in Password Cracking

Arushi Arora
(In collaboration with Ben Harsha)

May 2020

1 Problem Statement

In the field of password cracking, we often consider two types of adversaries. The first type of adversary has "perfect knowledge" of the distribution of user-selected passwords whereas the second type of adversary is one using state-of-the-art methods for cracking passwords.

Perfect Knowledge Attacker We can further categorize these into two categories. The first kind of perfect knowledge attacker has access to a database containing sampled passwords, specifically some underlying database

$$D = (pwd_1, c_1), (pwd_2, c_2), ..., (pwd_n, c_n) \tag{1}$$

where pwd_i is the i^{th} most common password and c_i is the number of times it has been observed, sorted lexicographically. The attacker does not know which password belongs to a particular user since the list is sorted. Although, the adversary knows the plain-text passwords and and their respective frequencies which could be possibly from a specific breach for instance, the LinkedIn or RockYou password breach [1]. In this work, we consider this category of Perfect Knowledge Attacker. The second type of perfect knowledge attacker knows the distribution of passwords, i.e. $Pr[x]$ for each password x, but the attacker does not know which passwords were sampled. The distribution of passwords is unknown making analysis of this attacker tough.

State-of-the-Art Attacker The second type of adversary is one using state-of-the-art methods like Neural networks [2], Probabilistic Context-Free Grammars [3], or Markov models [4] [5], and popular modern password guessing tools are John the Ripper (JTR) [1] [2] and Hashcat [3] [4].

[1] https://en.wikipedia.org/wiki/John$_{t}he_{R}ipper$

[2] https://www.openwall.com/john/

[3] https://hashcat.net/hashcat/

[4] https://en.wikipedia.org/wiki/Hashcat

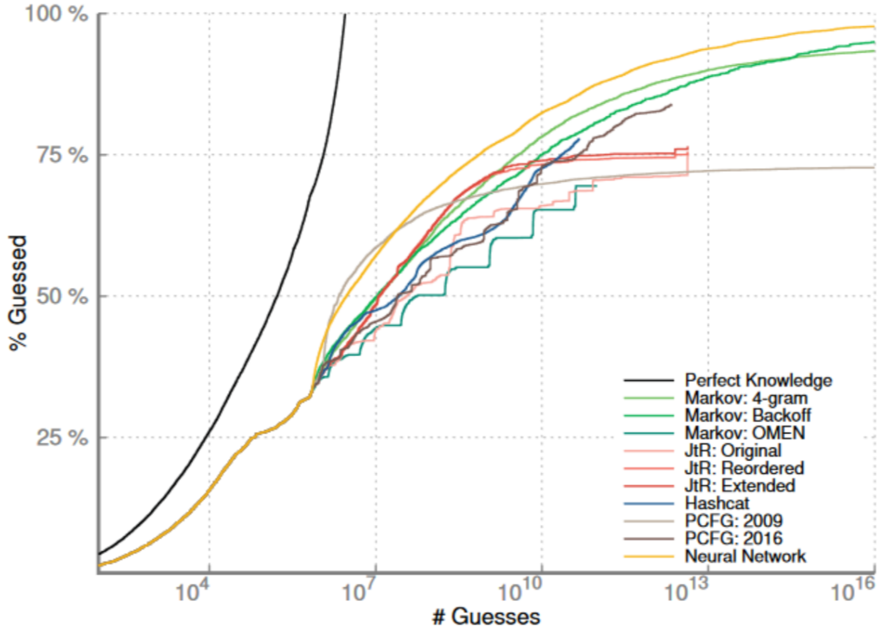

Figure 1: Password cracking percentages for the Neopets password list [6]

These two models currently show a significant performance gap, with perfect knowledge attackers having a significant advantage, e.g. cracking around twice the passwords in 10^6 guesses, over conventional cracking methods. The gap between these methods can be seen in Figure 1. Each of these two methods has its down-sides. An issue with perfect-knowledge attackers is that they overestimate cracking percentages, especially for unique passwords. Modeling a perfect-knowledge attacker as having access to these unique passwords overestimates any realistic probability of guessing them. Any practical dataset that the perfect knowledge attacker has can be considered a sample from a much larger dataset.

Thus, the actual probability of these unique passwords is highly exaggerated and provides an unrealistic advantage to a perfect-knowledge attacker. This overestimation also exists in the case of the second perfect knowledge attacker, described before, as a real world attacker does not have perfect knowledge of the complete distribution of passwords. E.g. if a password from a true distribution D has probability $p_i = 10^{-10}$ and we take a sample $\widehat{D}, |\widehat{D}| = N = 10^6$, the estimated probabiltiy of p_i will be 10^{-6} instead of 10^{-1}, an overestimate by a factor of 10000!

It is also overly optimistic to assume an attacker can only crack as many passwords as current state-of-the-art methods can in the same number of guesses, as we might underestimate the severity of a cracking attack. In reality, an attacker is capable of running a type of hybrid attack, attempting to gain the benefits of perfect knowledge via approximating a distribution D and defaulting to current cracking methods if their approximate distribution fails. We propose to describe this hybrid attack in detail and analyze its effectiveness when applied to existing empirical data sets obtained via password breaches.

2 Motivation

Guessing the password is one of the most common way hackers can break into user accounts. Simple weak passwords and strong password cracking methods enable intruders to easily gain access and control of a computing device leaving users vulnerable. A strong password provides essential protection from financial fraud and identity theft. Therefore, it is of utmost importance that passwords are chosen with a lot of care (Although, it is possible for “simple” passwords to be fairly strong e.g. $correcthorsebatterystaple$).

Our motivation here is to understand the risks posed by offline password attacks [7][8](this is the type of attacks we are mainly considering) and in turn credential stuffing which aims to use passwords cracked from one breach to attack other accounts where the password was reused.

Other than that, our proposed approach may be used to provide feedback to users by letting them know when they select a password that may be especially vulnerable to cracking. For instance, this can be achieved by using password strength meters which when designed appropriately may encourage users to select stronger passwords [9].

We also consider the research work [10], in which the authors claim that the vulnerability of users increases as a result of password leaks. Specifically, they show that a leak that reveals the passwords of just 1% of the users provides an attacker with enough information to potentially have a success rate of over 84% when trying to compromise other users of the same website. In other words, this further strengthens our basis for the hybrid attack technique, where the idea is to have a base dictionary/database of passwords and counts, imitating the perfect knowledge attacker(type 1).

3 Related Work

1. Reasoning Analytically About Password-Cracking Software [6]:
   In this paper, the authors design and implement tools that analytically compute properties of John the Ripper and Hashcat. Their research also determines whether a particular password would be generated along with its guess count. This is done with the help of two operations, rule inversion and guess counting, with which the analyze these tools without needing to enumerate guesses.

2. Password Cracking Using Probabilistic Context-Free Grammars [3]:
   In this paper the authors propose a method that generates password structures in highest probability order using a probabilistic context-free gram-

mar approach which uses previously disclosed passwords as its training set. This grammar generates word-mangling rules which is used in password cracking.

3. Fast, Lean, and Accurate: Modeling Password Guessability Using Neural Networks [2]:
   This research proposes a neural network based password guessing mechanism that is more effective than other state-of-the-art approaches, such as probabilistic context-free grammars and Markov models.

4. A Study of Probabilistic Password Models [5]:
   In this paper, evaluation of a large number of probabilistic password models, including Markov models using different normalization and smoothing methods is carried out. The research showcases that the Markov models, when done correctly, perform significantly better than the Probabilistic Context-Free Grammar [3].

   The above stated researches (1,2,3,4), that form the basis for our project, describe password cracking using State-of-the-art models using an algorithm like JtR/Hashcat, Probabilistic Context-Free Grammars and Neural Networks that generates an ordered list of passwords. We propose a simultaneous building of a database that mimics a database used by a perfect-knowledge attacker. In other words, our approach involves a concurrent updating of the database as more and more accounts are attacked. Also, since our proposed approach considers offline password cracking attacks, the below listed papers prove to be insightful.

5. The science of guessing: analyzing an anonymized corpus of 70 million passwords [8]:
   This research reports on the largest corpus of user-chosen passwords ever studied, consisting of anonymized password histograms representing almost 70 million Yahoo! users. The authors estimate that passwords provide fewer than 10 bits of security against an online, trawling attack, and only about 20 bits of security against an optimal offline dictionary attack. They further show that no practical amount of iterated hashing can prevent an adversary from breaking a large number of accounts given the opportunity for offline search. Their study also showcases that password distributions do not seem to vary much. With all populations of users they were able to generate similar skewed distributions with effective security varying by no more than a few bits.

6. On the Economics of Offline Password Cracking [7]:
   This research provides an evidence that Zipf's law models the distribution of user selected passwords (with the possible exception of the tail of the distribution). For instance, the law closely fits the Yahoo! password frequency corpus. Also, the authors apply their framework to analyze recent large scale password breaches including LastPass, AshleyMadison,

Dropbox and Yahoo!, challenging the claim that BCRYPT and PBKDF2-SHA256 provide adequate protection for user passwords. In other words, they suggest if the password distribution follows Zipf's law then their analysis indicates that a rational attacker will almost certainly crack 100% of user passwords.

7. Convergence of Password Guessing to Optimal Success Rates [10]:
   We also consider the research work as stated in section 2, in which the authors claim that the vulnerability of users increases as a result of password leaks. Specifically, they show that a leak that reveals the passwords of just 1% of the users provides an attacker with enough information to potentially have a success rate of over 84% when trying to compromise other users of the same website. In other words, this further strengthens our basis for the hybrid attack technique, where the idea is to have a base dictionary/database of passwords and counts, imitating the perfect knowledge attacker(type 1).

4 Approach

We aim to describe a method that closes the gap between perfect knowledge and state-of-the-art attacks with a hybrid approach. In a perfect knowledge situation, an adversary simply guesses passwords in descending order of probability. What we want to investigate is how effective it is to construct a sorted database $DB = (pwd_1, c_1), (pwd_2, c_2), ..., (pwd_i, c_i)$ (where pwd_i is the i^{th} most common password and c_i is the number of times it has currently been observed) for a given salted password hash list $\pi = (s_1, \pi_1), (s_2, \pi_2), ...$ while we are cracking. An example hybrid strategy is as follows:

Algorithm 1 Hybrid-attack

```
$DB \leftarrow \varnothing$
for each salt and hash pair $(s_i, \pi_i) \in \pi$ do
    iterate through the passwords in $DB$ and calculate the salted hash of each.
    if the password is cracked and revealed to be $pwd_i$ then
        update $c_i$ in $DB$
    else switch to a secondary cracking method such as PCFG/Neural network
        check some number $B$ additional passwords
        update $DB$ with any new information
    end if
end for
```

5 Experimentation & Results

We carry out the experiment by simulating a hybrid attack considering the following datasets:

1. Wordlists: Xato, LinkedIn
2. Rulelists: Hashcat (best64, T0X1C), JtR (Spiderlabs)
3. Testsets: RockYou, LinkedIn
4. Dictionary: RockYou with frequency counts, LinkedIn with frequency counts

Procedure: We first implemented some state-of-the-art approches, specifically from [6]. We considered a combination of various wordlists, rulelists and datasets to observe and understand their behaviour. We plotted the guessability graphs as shown in figure 2-7 from the resulting guessing number files.

To simulate the hybrid attack, we followed the algorithm as stated in section 4. We considered the base dictionary to be a list of passwords with count. For instance, while considering a state-of-the-art password cracking method over RockYou dataset, we considered RockYou passwords with count file (similary for LinkedIn dataset). Then, for each salt and hash pair in a given list, we iterate through this base dictionary file, and calculate the salted hash of each password. If this password is cracked, then we update the count of password in the base dictionary. In case we were unable to to crack, we shifted to the secondary cracking method. In other words, we referred to the resulting guessing number file generated by running the corresponding state-of-the-art method in [6]. We then plotted the guessability graphs for a simple state-of-the-art versus hybrid approach.

Figure 8, 9 and 10 represent the guessability graphs for state-of-the-art and hybrid simulation attacks over RockYou dataset. It can be clearly observed that hybrid attack, as proposed in this work, performs better in terms of percentage of password guessed w.r.t number of guesses. We conclude that Hybrid attack is a better and more effective approach to password cracking initiating the process much faster than a state-of-the-art approach. One can also note that the percentage guessed (y-axis), in these figures in somewhat nearing 30%. This could potentially be a result of hybrid attack having only 500,000 samples.

While considering the LinkedIn dataset, we also realized some downsides to the Hybrid approach. As shown in figure 11 and 12, we can see that after sometime, the state-of-the-art overtakes the hybrid attack approach. This is due to the fact that less common passwords are less likely to be in the database (the base dictionary). However, we still paid to check the entire database before we switched to the state of the art method, implying that once we start hitting uncommon passwords the hybrid attacker is less effective.

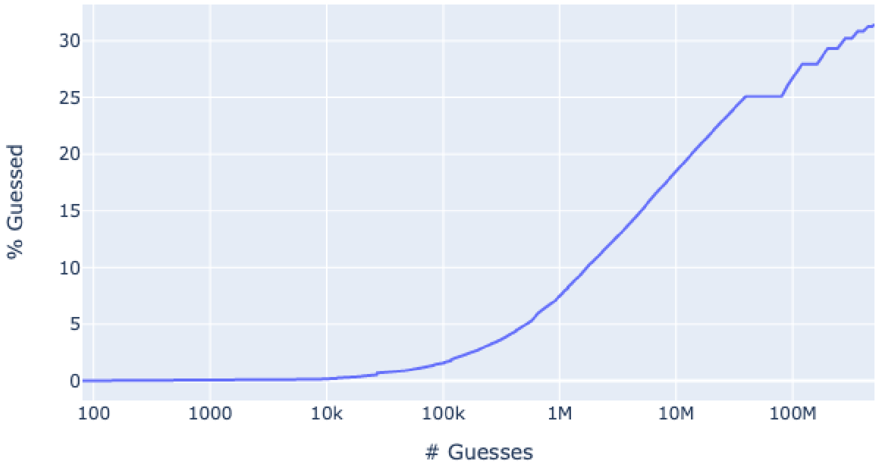

Figure 2: Wordlist: LinkedIn Rulelist: Best64 Dataset: RockYou

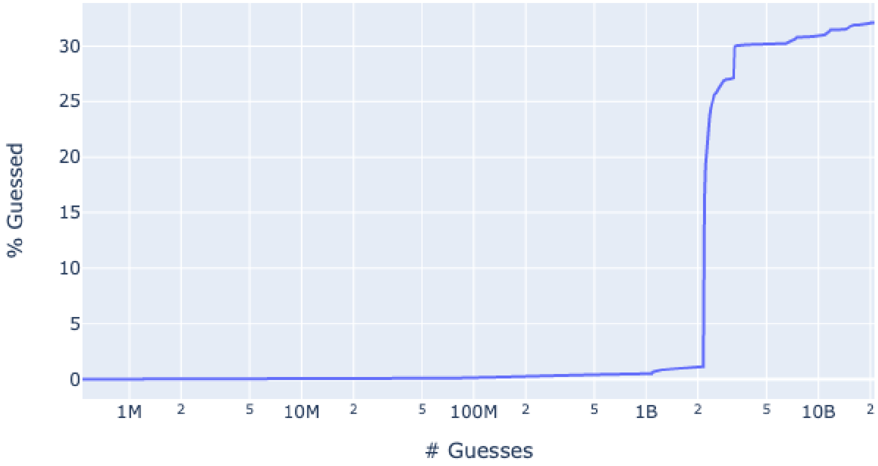

Figure 3: Wordlist: LinkedIn Rulelist: Best64 Dataset: RockYou

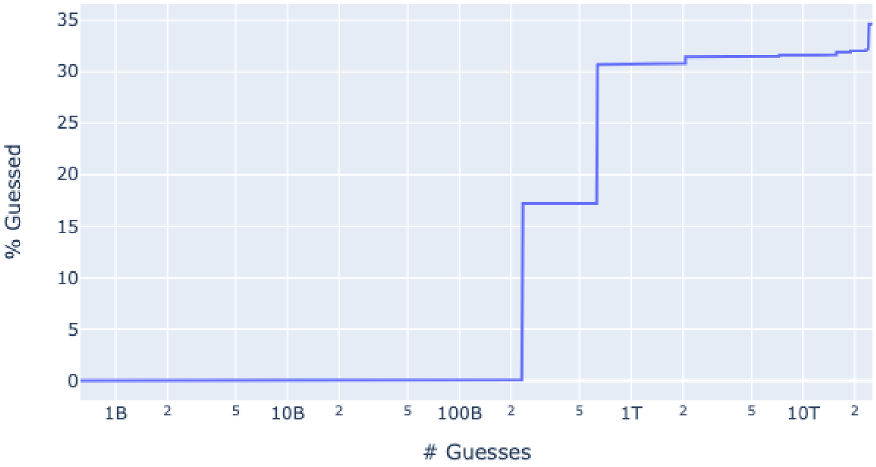

Figure 4: Wordlist: Xato Rulelist: Spiderlabs Dataset: RockYou

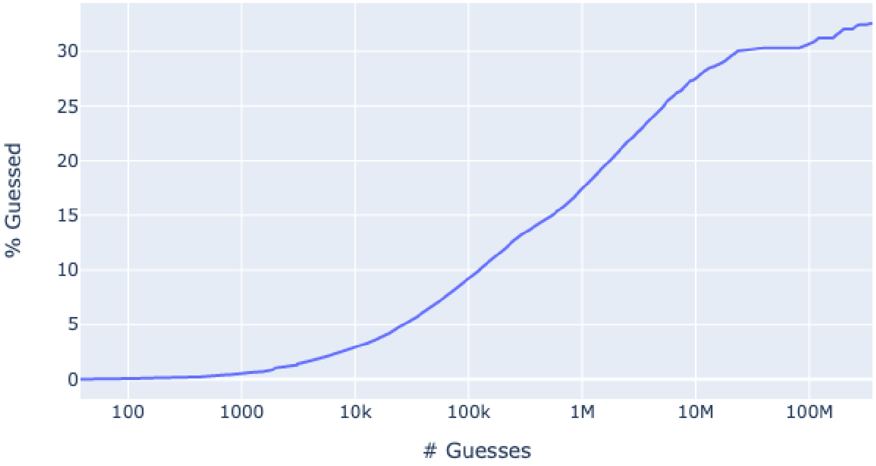

Figure 5: Wordlist: LinkedIn Rulelist: Best64 Dataset: RockYou

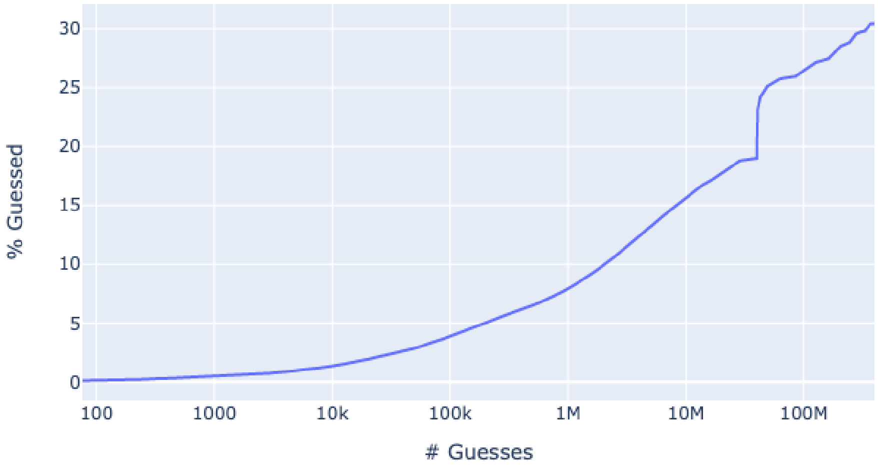

Figure 6: Wordlist: Xato Rulelist: Best64 Dataset: LinkedIn

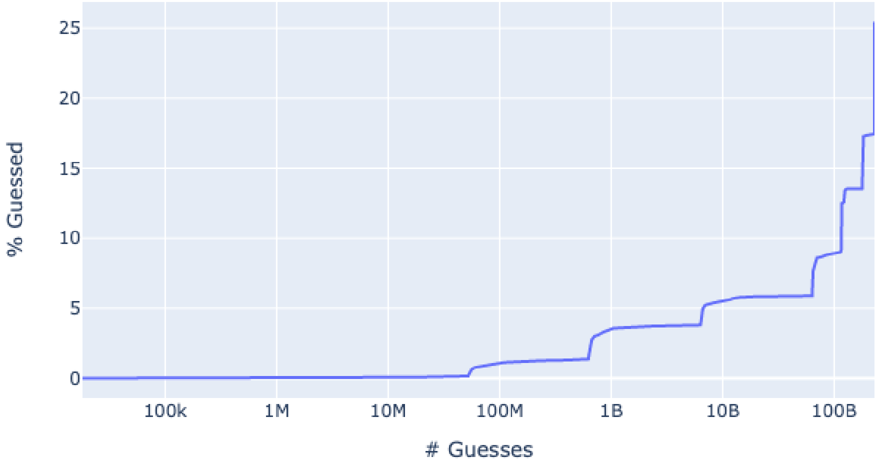

Figure 7: Wordlist: Xato Rulelist: Spiderlabs Dataset: LinkedIn

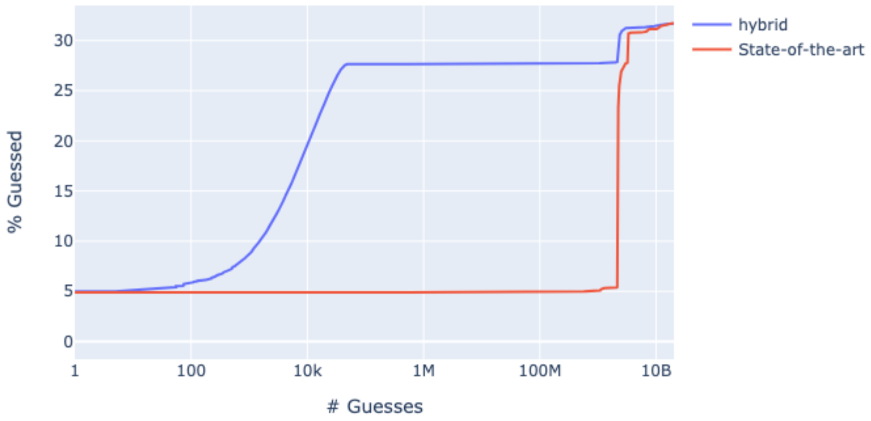

Figure 8: Comparison between Hybrid and State-of-the-Art attack simulation. The simulation uses Xato worlist, T0X1C rulelist and RockYou testset.

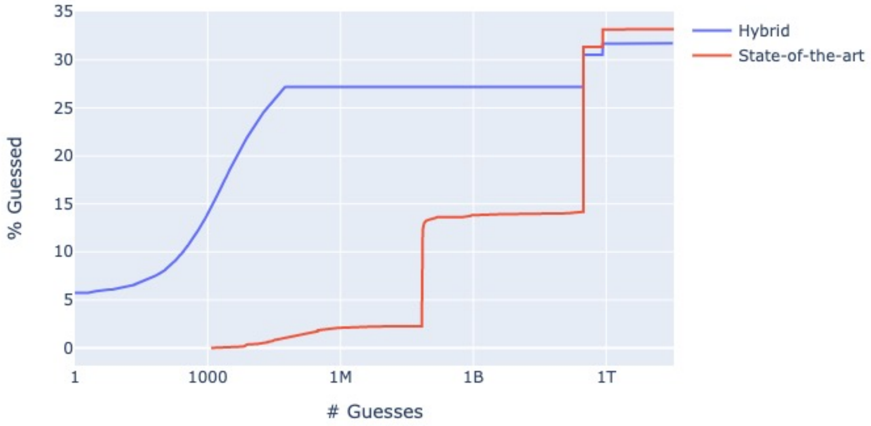

Figure 9: Comparison between Hybrid and State-of-the-Art attack simulation. The simulation uses LinkedIn worlist, Spiderlabs rulelist and RockYou testset.

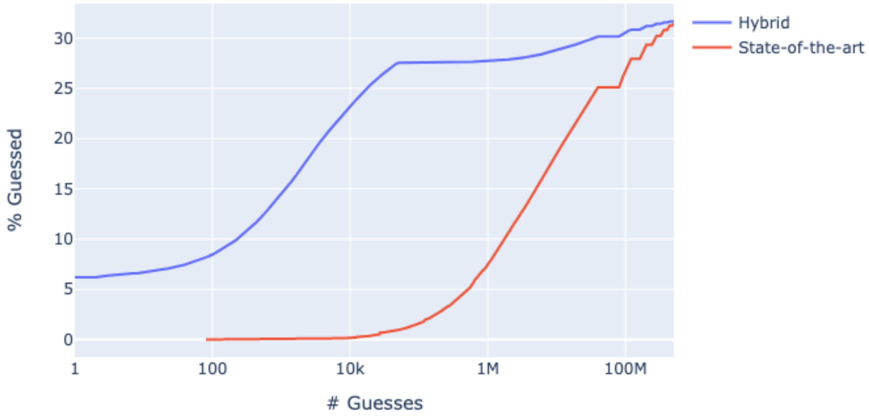

Figure 10: Comparison between Hybrid and State-of-the-Art attack simulation. The simulation uses LinkedIn worlist, Best64 rulelist and RockYou testset.

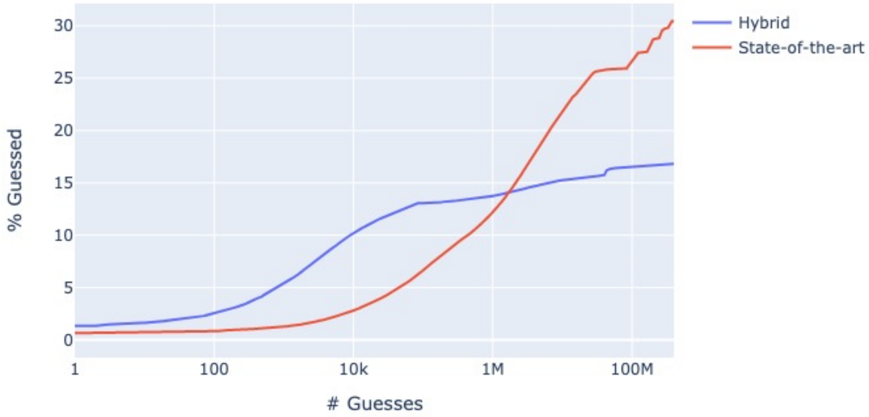

Figure 11: Comparison between Hybrid and State-of-the-Art attack simulation. The simulation uses Xato worlist, Best64 rulelist and LinkedIn testset.

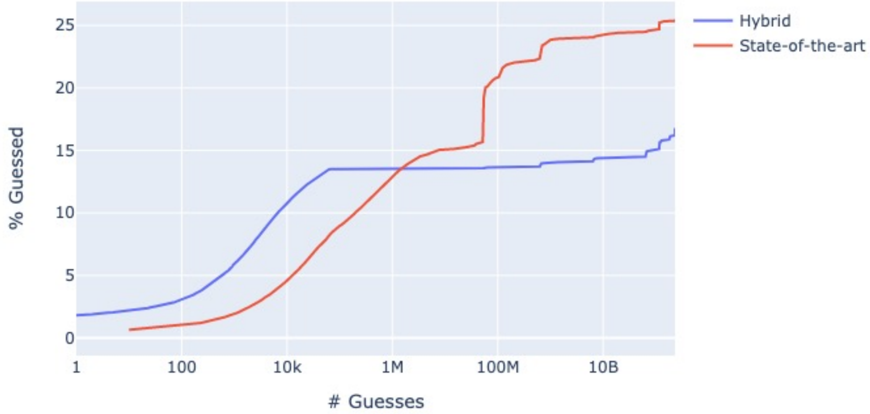

Figure 12: Comparison between Hybrid and State-of-the-Art attack simulation. The simulation uses Xato worlist, Spiderlabs rulelist and LinkedIn testset.

6 Barriers

1. Implementation of each ”state-of-the-art” password-cracking methodology, as they have been proposed and developed by a different researchers and hence research labs that use a programming language as per their need.
2. Finding the right dataset for the bootstrapping phase.
3. Collecting additional password data to include in our initial database.
4. Handling and formatting large datasets and files for analysis. Making our code compatible with previous existing state-of-the-art repositories.
5. Having patience! Experimentation takes so long!

7 Future Work

1. In the near future we would simulate hybrid attack over various other datasets with altering the sample size for the base dictionary.
2. In future, we would consider and include optimization strategies like maximum guess limits, skipping most of the guessing steps in a brute forcing phase, how long we should run before starting to use the dictionary, or other minor optimizations.

3. Also, introducing techniques to make hybrid attack perform well for uncommon passwords could also be carried out.

4. We would also want to consider the second type of perfect knowledge attacker for our analysis. Peiyuan Liu's course project analyzes the lower bound of the number of cracked user passwords in an arbitrary sample dataset and would be a boost for this approach.

5. We also plan to plot the guesswork graphs for the results we obtained.

6. In the coming weeks, we also would want to test our approach starting with an empty dictionary, or a dictionary initialized with just 5% of the data taken from Linkedin/RockYou dataset. Also, we would want to observe how hybrid attack performs when we choose a base dictionary with frequently used passwords (lets say with passwords having a frequency count > 3).

7. We also plan to write a research paper to present our project work (after we perform the above stated experiments).

References

[1] D. Mirante and J. Cappos, "Understanding password database compromises," *Dept. of Computer Science and Engineering Polytechnic Inst. of NYU, Tech. Rep. TR-CSE-2013-02*, 2013.

[2] W. Melicher, B. Ur, S. M. Segreti, S. Komanduri, L. Bauer, N. Christin, and L. F. Cranor, "Fast, lean, and accurate: Modeling password guessability using neural networks," in *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pp. 175–191, 2016.

[3] M. Weir, S. Aggarwal, B. De Medeiros, and B. Glodek, "Password cracking using probabilistic context-free grammars," in *2009 30th IEEE Symposium on Security and Privacy*, pp. 391–405, IEEE, 2009.

[4] M. Dell'Amico and M. Filippone, "Monte carlo strength evaluation: Fast and reliable password checking," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 158–169, 2015.

[5] J. Ma, W. Yang, M. Luo, and N. Li, "A study of probabilistic password models," in *2014 IEEE Symposium on Security and Privacy*, pp. 689–704, IEEE, 2014.

[6] E. Liu, A. Nakanishi, M. Golla, D. Cash, and B. Ur, "Reasoning analytically about password-cracking software," in *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 380–397, IEEE, 2019.

[7] J. Blocki, B. Harsha, and S. Zhou, "On the economics of offline password cracking," in *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 853–871, IEEE, 2018.

[8] J. Bonneau, "The science of guessing: analyzing an anonymized corpus of 70 million passwords," in *2012 IEEE Symposium on Security and Privacy*, pp. 538–552, IEEE, 2012.

[9] B. Ur, P. G. Kelley, S. Komanduri, J. Lee, M. Maass, M. L. Mazurek, T. Passaro, R. Shay, T. Vidas, L. Bauer, N. Christin, and L. F. Cranor, "How does your password measure up? The effect of strength meters on password creation," in *USENIX Security 2012: 21st USENIX Security Symposium* (T. Kohno, ed.), (Bellevue, WA, USA), pp. 65–80, USENIX Association, Aug. 8–10, 2012.

[10] H. Murray and D. Malone, "Convergence of password guessing to optimal success rates," *Entropy*, vol. 22, no. 4, p. 378, 2020.