# Presenting New Dangers:<br>A Deep Learning Approach to Password Cracking

Annie Chen

*Tufts University, Fall 2018*

**Abstract**

Password cracking and user account exploitation is one of the largest issues in cybersecurity. State-of-the-art password guessing tools, such as HashCat and John the Ripper, enable potential attackers to check billions of passwords per second against generated password hashes. While these tools have proved to be effective in cracking passwords, recent research shows that combining deep learning techniques with these tools can produce significantly better results. Specifically, using Generative Adversarial Networks (GANs), which comprises of two neural networks, to generate high-quality password guesses can improve the existing tools to match 51%-73% more passwords than just the tools alone. This significant improvement demonstrates that this new approach using deep learning can generate numerous new passwords that were once beyond the reach of other tools. On one hand, this is an impressive result driven by researchers of deep learning and cybersecurity, and on the other, a strong warning to the community of the increase dangers of weak password authentication.

## 1. Introduction

As our society becomes increasingly dependent on technology, each individual's online presence grows and expands onto different platforms. Many web applications like email, social media (i.e. Twitter, Facebook, etc.), and online banking all require users to have an account, to be validated only through the correct username and password pair. If owners of online applications are not careful in building and maintaining a secure system, many user accounts may be subject to attack from malicious hackers. Attacks can lead to the leaking private user information taken from the application's database or attackers logging into users' accounts and exploiting the victims from there. This is why password cracking is such an integral part of cybersecurity. If attackers are able to break into a system and access a database of username and hashed passwords, it is only a matter of time before each password can be cracked. The crucial part is how quickly attackers can crack these passwords – if they can crack them before the owners of the application responds, then they get total access to the users' accounts.

This paper examines the possibility of applying deep learning techniques to password cracking to see if this can lead to more effective password guessing. Specially, we explore whether Generative Adversarial Networks (GANs) can generate more correct password guesses than current state-of-the-art tools. In password cracking, the main goal is to have as many correct password guesses as possible in the smallest number of tries. The challenge is to supply a list of high-quality probable password guesses, as the better the list, the more passwords one can crack in a smaller amount of time. In the past, this list has come from past password database leaks, words taken from a dictionary, or a combination of both with transformations on these words (i.e. `password` to `pa$$w0rd`). However, this paper investigates whether there can be another method of generating a list of passwords – through GANs.

## 2. To the Community

Though password cracking has consistently been a central topic to cybersecurity, this discovery of successfully applying deep learning to password cracking to make it even more effective has led to a new call to action. Computing power has been on the rise with the increase in data. Much of our identity is shifting or has already shifted over to the online world – where we regularly give our private information to websites, secured only through a password. Now that the reality of the bubble is bursting – that these passwords are no longer secure – a new sense of urgency arises to protect our information and online identities. This then poses the next question, if the current authentication procedures are too weak (i.e. passwords can be too easily guessed through deep learning methods), then how can authentication be strengthened? New technical measures must be introduced to ensure that passwords are not so predictable. Possibilities include pushing users to make stronger passwords of certain lengths or alphanumeric combinations, or introduce additional security measures like two factor authentication.

## 3. Background

### *3.1. Deep Learning*

Deep learning is a subset of machine learning regarding algorithms inspired by the structure and function of the human brain, formally called artificial neural networks [1]. A neural network is a simulation of the network of neurons in a brain so that the computer program will be able to "learn" and make decisions in a humanlike manner. In a program, this is done by constructing a model with different layers of mathematical processing. The "deep" in deep learning refers to the multi-layer aspect of neural networks – the fact that many layers are stacked up to construct the model. This is not seen in traditional machine learning models.

Deep learning has been used to deliver state-of-the-art accuracy in tasks such as computer vision, speech recognition, image processing, and natural language processing [2]. In the areas of privacy and security, research on deep learning has been focused on three major areas: privacy-preserving deep learning, attacks on deep learning models trained on private data, and attacks that lead to input misclassification for otherwise very accurate models [2]. Thus, the application of deep learning on password guessing is fairly new, and does not fall completely in any of the three categories. However, this paper aims to demonstrate how deep learning can be effectively applied to password cracking, and the dangers it presents to the cybersecurity world given this new development.

### *3.2. Generative Adversarial Networks*

Generative Adversarial Networks (GANs) is a recently-introduced deep learning framework that has become widely popular due to its high accuracy without the need of large data samples. Traditionally, in order to make machine learning models very accurate, they had to be trained over and over again with great amounts of data. One can imagine how costly in computation and time it is to repeat the training and to acquire this vast amount of data. But GANs have been developed so that machines can learn to teach themselves as well as produce some of this data. Now, instead of a data scientist constantly going in search of more data and then training the model, a deep learning model can do this task. This is the central idea behind the creation of GANs. To dig a little deeper in how this is actually done, consider two neural networks, with a relatively small input sample of real data. First, there is "generator" network, G, which is tasked to create training data, starting randomly and getting as realistic as it can. Then, there is a "discriminator" network, D, that tries to distinguish between real input data from the fake data generated by G. By setting these two networks up as adversaries, G tries to fool D while D tries to distinguish the samples from the real world and G. Each network tries to master its own task through thousands of iterations, with no manual intervention. In the end, there is a model that creates true-looking fakes and another that can detect most fakes from reals. A simple version of this process can be seen in Figure 1.0.

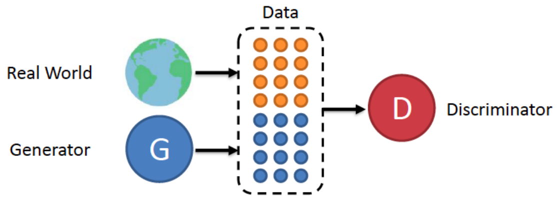

Figure 1: General GANs Architecture

A real world analogy to demonstrate the two networks at work is to consider the generator as a police sketch artist, and the discriminator as an eye witness. The sketch artist starts with a basic sketch of a suspect, and then continuously refines the image using feedback from the eye witness, until the sketch is a close approximation of the suspect's real image [3].

### *3.3. Password Guessing Tools and Methods*

State-of-the-art password guessing tools, such as HashCat and John the Ripper, can check billions of passwords per second against generated password hashes. The only way passwords can be cracked is to test candidate passwords, turn them into hashes, and check them against the unknown password. It is not possible to reverse a hashed password to retrieve the original plaintext by virtue of a hash (hash functions are irreversible). HashCat, John the Ripper, and other password guessing tools can perform a variety of attacks, with the three most popular being brute force attacks, dictionary attacks, and rule based attacks. The difference between these attacks is on how the password guesses are formed.

An exhaustive brute-force attack goes through every combination of characters up to a given length. For instance, generating passwords of length 5 will output `aaaaa, aaaab, aaaac,....` As one can imagine, the computational time increases exponentially with length. For instance, given the set of characters `[a-z], [A-Z], [0-9]` with password length up to 8, the brute-force number of computation is:

$$62^8 \approx 218e^14$$

If one is working on a workstation or multiple PCs together that can process 100,000,000 passwords/sec, this takes 25 days. If one is using distributed computing, thereby increasing computational power, and can process 1,000,000,000 passwords/sec, then this takes 60 hours [6]. In either case, it is evident that while brute-force attacks are thorough, they are not very efficient.

Dictionary-based attacks are becoming increasingly common based on their success rate and effectiveness. This type of attacks uses a list of words, either from a dictionary or commonly used passwords revealed from past password cracks, turn them into hashes, and checks them against the unknown password hash. Dictionary attacks often succeed because people have a natural tendency to choose ordinary words or common passwords with little variants.

An additional measure that can be combined with dictionary attacks is to use a rule-based approach on top of the dictionary list. This consists of generating password guesses from transformations based on the given rule. For example, with the word `hello`, an example rule can be to concatenate numbers onto the word, creating the variation of `hello123`. Such transformations on the word is currently the main way of generating new forms of possible high-probability passwords. However, inherently with this method, the new generated passwords are limited to the set of fixed rules. In other words, the new passwords created are based on the structure the rule identifies. This means that these rules create the variations that are believed to occur, though different patterns and structures of passwords may be present that have not been identified by these rules. Thus, to address this issue, a deep learning approach has been proposed to autonomously extract password properties that these rules do not encode.

## 4. The Application of Deep Learning in Password Guessing

### *4.1. Motivation of PassGAN*

PassGAN is a novel approach introduced in 2017 as the first password guessing technique based on GANs. The purpose of PassGAN is to replace the human-generated password rules in a rule-based attack with a deep learning model to autonomously identify password structures and properties to create better password guesses [4]. This model takes a list of real leaked passwords and generates a new list of potential passwords that have the highest probabilities of being real passwords. Prior to the introduction of this approach, new high-probability passwords were primarily generated through rule-based attacks. This, however, is limited to the human-constructed rules that it is based on, which naturally come from human-made observations on the properties of passwords (i.e. concatenating `123`, or replacing characters for symbols like `h@ck3r`). Thus, the limitation of the rule-based attack boils down to the limitation of human observation. To address this problem, PassGAN was built as a deep learning model to extract patterns or properties of passwords that is otherwise indistinguishable to the human eye. A further attribute of using deep learning is that the model learns these features autonomously, without *apriori* knowledge on the structure or other properties of passwords.

### *4.2. Implementation and Architecture of PassGAN*

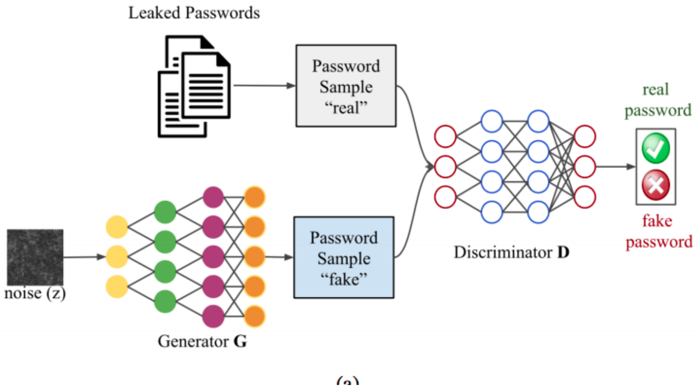

Figure 2: Overview of PassGAN's architecture [4]

As shown in Figure 2.0, the discriminator D processes passwords from the training dataset, which is comprised of real passwords from leaked databases as well as password samples produced by the generator G. Based on the feedback from D, G finetunes its network to produce password samples that are close to the training set, with no direct access to the training set. Ideally, in the final iterations of training the model, the generator G would be outputting such high quality password samples that the discriminator D would not be able to tell between the real sample and the "fake" one. Applying GANs for password guessing is an interesting approach because in theory, if the password samples generated by G are so realistic that the discriminator D is not able to distinguish them apart, then these high-quality password guesses are theoretically just as probable as other real passwords.

The specific implementation details of PassGAN can be found documented in the Hitaj et al. paper [4].

### *4.3. Analysis and Evaluation*

The evaluation of PassGAN is done in comparison to HashCat Best64 and gen2 and John the Ripper Spiderlab, two tools with different sets of rules to generate password guesses. The underlying evaluation is really to determine whether the properties that PassGAN autonomously extracts from passwords can compete with state-of-the-art human-generated rules [4]. After running on two different datasets (RockYou and LinkedIn), results show that PassGAN was able to generate "at least the same number of matches" as the rule-based approach on HashCat and John the Ripper [4]. Other metrics such as probability density estimates were also evaluated to understand the performance of PassGAN as a whole. Results show that PassGAN is able to correctly estimate the probabilities of many of the 50 most frequent passwords in the training set [4]. This is an impressive result as being able to correctly measure the frequency of passwords allows PassGAN to match common passwords within a small number of guesses.

Additionally, the results show that the best overall performance was when multiple techniques were combined, specifically when PassGAN and HashCat were used jointly. When the output of PassGAN and the output of HashCat were combined as a set, 51-73% more passwords were matched than with HashCat alone [4]. This signifies that many of the successful password matches from PassGAN and HashCat were unique, in order to create this larger set of successful password guesses. The implication of this is that PassGAN is able to extract a considerable number of password properties that the rules used with HashCat were not able to encode. This indicates that PassGAN is able to generate many new passwords beyond the reach of HashCat and other tools.

## 5. Applications

### *5.1. The Emergence of a New Password Cracking Tool*

From the results, it is clear that PassGAN can be an efficient and accurate password cracking tool – perhaps one good enough to slowly replace current popular tools. In the future, it is possible that those in the cyber-security field will start shifting from using password cracking tools based on human-generated rules, HashCat Best64 and gen2 and John the Ripper Spiderlab, and use machine learning tools like PassGAN instead, for improved accuracy of password guesses. In addition to improving the accuracy percentage, PassGAN also offers a distinct advantage in being able to create passwords indefinitely, whereas for instance, HashCat is only able to produce up to 650 million passwords [7]. Thus, though PassGAN is a new project in the field that most cyber-security specialists have not started using due to its novelty and lack of official publication as a tool, it shows promising results as a first step in using GANs in password cracking.

### *5.2. Significance*

Though it is an impressive achievement that authors of PassGAN were able to create a tool with autonomous learning skills of password rules, it is surely a warning to the public and cyber-security specialists on the weaknesses of passwords. PassGAN proves the possibility of using machine learning in this realm, and demonstrates the power of it, and is only in its first few iterations of development. There are huge implications to this tool, as some foresee a drastic improvement with the increase of computing power and availability of data in the future.

## 6. Conclusion

While PassGAN alone does not beat current state-of-the-art tools by a large margin, it strongly demonstrates that deep learning can be integrated into existing tools to gain better results as a whole. Its promising results show that with improvements upon this model, for instance through training it with a wider variety of datasets, it could become a very powerful tool. Academically, this is an interesting application of deep learning, but realistically, it can be a dangerous one. Though PassGAN has been developed under an academic setting for research purposes, it is possible that malicious attackers adopt a similarly powerful tool, allowing them to crack passwords from a database leak quickly and then exploit the victim's account. Furthermore, given the common occurrence of reusing the same passwords with little variations on different accounts, the danger of such password attacks further grows. This is not an issue only for cyber-security specialists to be aware of, but the general public as well. It serves as a reminder for system programmers to reinforce password strength restrictions to encourage stronger, less guessable, passwords. Alternatively, other measures like two-factor authentication can improve the chances of keeping the system secure even more. As for the general public, with this awareness of malicious attackers being able to quickly crack passwords, safeguarding one's personal cyber-security should become and remain as a top priority

## Bibliography

[1] Jason Brownlee. *What is Deep Learning?*
https://machinelearningmastery.com/what-is-deep-learning

[2] NVIDIA Developer. *Deep Learning.*
https://developer.nvidia.com/deep-learning

[3] Jai Vijayan. *PassGAN: Password Cracking Using Machine Learning*
https://www.darkreading.com/analytics/passgan-password-cracking-using-machine-learning/d/d-id/1329964

[4] B. Hitaj, P. Gasti, G. Ateniese, and F. Pérez-Cruz. *"Passgan: A deep learning approach for password guessing,"* CoRR, vol. abs/1709.00440, 2017
https://arxiv.org/abs/1709.00440

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. "Generative adversarial nets," in *Advances in neural information processing systems*
https://arxiv.org/abs/1709.00440

[6] The Home Computer Security Center. *Password Recovery Speeds*
https://web.archive.org/web/20180412051235/http://www.lockdown.co.uk/?pg=combis=articles

[7] Matthew Hutson. *Artificial intelligence just made guessing your password a whole lot easier*
https://www.sciencemag.org/news/2017/09/artificial-intelligence-just-made-guessing-your-password-whole-lot-easier