"Add '!' at the End to Make It Secure": Observing users' desire for control of password generation

Jeffrey Goldberg
1Password, Inc.

Mitchell Cohen
1Password, Inc.

Abstract

Some users appear to resist ceding full control of password generation to strong password generators. A substantial part of that resistance may arise from incorrect beliefs about what makes a password strong, but some resistance appears to also be tied to perceptions of what makes a password compatible with a site. Additionally, some people may simply find the loss of control unpleasant. This industry report does not report on any data collection beyond anecdotes of customer interaction and intuitions of our staff.

1 Introduction

Over the past several years [redacted], a password manager vendor, has been reducing the controls exposed to users for its password generator. The most recent step in this direction has been the introduction of [redacted]. Customer feedback has been intriguing and enlightening. We would like to share some of that and our thoughts and ideas on that feedback.

The clearest underlying reason for complaint is that users have unsurprisingly come to the conclusion that passwords conforming to the password complexity rules that have been inflicted on them over decades serve as a guide to what makes a password strong. These beliefs are well documented in laboratory studies [e.g., 2, 3, 1]. How these and other beliefs may interact with feelings about password generation is the subject of what follows.

USENIX Symposium on Usable Privacy and Security (SOUPS) 2021.
August 8–10, 2021, Vancouver, B.C., Canada.

2 Goals of complexity

We presume that the original intention of password complexity requirements was to flatten the distribution of created passwords. If, say, the most common password before such rules was "`password`" the hope was that whatever became most common afterwards would not be as common as the original. If some of the people originally using "`password`" switched to "`password1`" when being required to include digits and others moved to "`passw0rd`" we still have an improvement. There are ways to model the notion of flattening the distribution, but we will skip those here.

We do not explore here whether these attempts were successful, Neither do we explore whether the creators of complexity policies were explicitly aware of this goal nor whether, cargo cult-like, perpetrators of complexity requirements have entirely lost sight of the original intent. It is however useful to think in terms of the desire to flatten the curve in the discussion to follow.

3 Rules for machines are different

A good password generator will generate a uniform distribution. That is, given particular settings any password that it can generate is no more or less likely than to be generated than any other. This results in a flat distribution.

With a uniform distribution, the strength of the generator is solely a function of the number of distinct passwords it can generate. Complexity rules reduce the number of passwords (of a given length) that can be generated, and so always result in weaker password generation. There are about 457 trillion eight character password which can be made up of upper and lower case letters, digits, and a set of six symbols. If, however, we require that the password contain at least one character of each type, there are about 152 trillion eight character passwords. The complexity requirement reduces to the the set of generable passwords by two thirds.

Allowing more types of characters to appear in a uniformly generated password increases the strength; but *requiring* more types of characters reduces the strength. The same applies to requiring a specific number of some category of characters or to forbidding consecutive identical characters. This leaves us in a situation where the kinds of properties of passwords that people have been led to believe create strong passwords are counter-productive to when generating passwords uniformly.

4 Appearances matter

In addition to generating strong passwords, the generator must also create passwords which conform to the complexity requirements of the site or service it will be used with. While meeting that challenge is interesting, it is not the subject of here. What is more interesting is meeting those criteria while also appearing to do so. We found that we needed to craft our generator to not only meet compatibility and security requirements, but also to appear to do so to the user. In short we are aiming to meet and balance four criteria for generated passwords: 1. Be strong. 2. Be compatible with the requirements of the vast majority of websites. 3. Appear strong to the user. 4. Appear site compatible to the user.

Criterion 1 is the basic requirement of any password generation scheme. Criterion 2 is more interesting, and given time we would have more to say about it. It should not be surprising that we've encountered some conflict between criteria 1 and 3, but we suspect that some of the resistance we've encountered is due to failure to meet criterion 4. Finally, we are not in a position to disentangle user concerns regarding (apparent) strength and compatibility from a simple desire to maintain a sense of control over password generation.

5 Appear strong

There are 61 distinct characters which can go into a generated random password. This is 52 upper and lowercase letters, the digits, and the six most commonly accepted symbols. We exclude a total of seven digits and letters which may be visually ambiguous.

If strength were the only criterion then generating a password of length N would simply be drawing randomly and uniformly (with replacement) from that set of 61 characters N times. However, to achieve compatibility we need to, among other things, ensure that there is at least one symbol. As there are six symbols out of a set of 61 characters, this means that on average there will be about one symbol for every ten characters in the generated password.

While there there is always be at least one symbol, having so few in a generated password look like too few to some users. It doesn't give them the sense of gibberish that they expect. To a slightly lesser extent, this holds of digits as well. We allow seven possible digits, and so they show up more rarely than users expect.

6 Paraphrases

We list here a few short and distilled paraphrases of conversations we've had internally over the years with respect to our password generator.[1]

Conversation 1 (Size matters)

A: For a random password 14 characters would be strong enough for anything.

B: Yeah, but you know as well as I that users won't see them that way. We need to make them longer.

Conversation 2 (Stars shine)

A: The "*" character is less well accepted by sites than our other five symbols. So let's just go with the top most accepted five.

B: But "*" is visually much more salient. Without these the passwords won't look random enough.

Conversation 3 (The wierder the better)

A: Saying "at least one digit" gives us stronger passwords than "exactly three digits."

B: But the more digits and symbols we generate the stronger it will look.

Conversation 4 (But not too weird)

People were tripped up by character passwords in general, but especially those that had clusters of symbols and *especially* those with symbols at the beginning. So something like "`-*.gQfsdFM`" looked like it would fail.

7 Concluding remarks

We would love to see systematic studies to help test our intuitions. And if our ideas – good or bad – help spur interesting and useful research that is a good thing. We believe that it will be a long time before people stop worrying about the fine tuning and learn to love trimmed down password generators. We hope to be proven wrong.

[1] For a more complete list, see the slides.

References

[1] Tobias Seitz and Heinrich Hussmann. "PASDJO: Quantifying Password Strength Perceptions with an Online Game". In: *Proceedings of the 29th Australian Conference on Computer-Human Interaction*. OZCHI '17. Brisbane, Queensland, Australia: Association for Computing Machinery, 2017, pp. 117–125. ISBN: 9781450353793. DOI: 10.1145/3152771.3152784. URL: https://doi.org/10.1145/3152771.3152784.

[2] Blase Ur et al. ""I Added '!' at the End to Make It Secure": Observing Password Creation in the Lab". In: *Proceedings of the Eleventh Symposium On Usable Privacy and Security (SOUPS)*. Ottawa, Canada, 2015. URL: https://www.usenix.org/conference/soups2015/proceedings/presentation/ur.

[3] Blase Ur et al. "Do Users' Perceptions of Password Security Match Reality?" In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. San Jose, California, USA: Association for Computing Machinery, 2016, pp. 3748–3760. ISBN: 9781450333627. DOI: 10.1145/2858036.2858546. URL: https://doi.org/10.1145/2858036.2858546.