Add '!' at the End to Make It Secure

Observing users' desire for control of password generation

Jeffrey Goldberg & Mitchell Cohen

What 1Password does

- We make a password manager
- It has a password generator
- We used to expose lots of knobs and controls for users to tune and configure the generator
- We have been progressively removing those controls over the course of years

Tourism Vancouver / Fraanz Morzo Photography

A generated password should …

1. Be strong
2. Appear strong
3. Be site compatible appear
4. Appear site compatible

Strength (real and imagined)

People don't know what makes a password secure

(Everyone in this room already knows this)

"I added '!' at the end to make it secure." P34

felt that "usually numbers and a symbol will make the password strong.

[Ur et al. 2015]

A good reason for complexity rules

Well, it might have been a good reason at some point in the past

- Instructions to users for password creation are intended to flatten the curve
- If `password` is the most common password requiring digits will get some of those **people** to switch to `password1` and others will switch to `passw0rd`.
- Each of `password1` and `passw0rd` will be less common than the original.

Reasons or Rationalizations?

And other questions we won't answer

- Did such rules ever work?
- Did the original creators explicitly understand the good reasons?
- Have today's perpetrators of complexity requirements lost sight of the original intent?

We don't address any of those questions today

Rules for machines are different

Uniform generation

All passwords are equal (probability)!

- People are far more likely to pick the most popularly picked passwords. (Not wholly a tautology)
- Generator will pick uniformly from the full set of possible passwords
- Complexity rules (may once have) help(ed) people pick (slightly) more uniformly
- Complexity rules only reduce the set of possible passwords a machine would draw from

A (contrived) Example

- Given a set of 68 characters to draw from there are 457 trillion eight character passwords that can be generated.
- If we require that there be at least one character from each of the four character classes then there are 152 trillion eight character passwords that can be generated.
- This complexity requirement cuts the number of possible passwords to 1/3 the original set of possibilities.

The kinds of requirements that people believe make a password strong make generated passwords weaker

Site compatibility

Machines need default rules

For when the generator doesn't know specific requirements for a site

- At least one uppercase letter
- At least one lowercase letter
- At least one digit
- At least one symbol
- No consecutive identical characters (coming soon)

Conversations

Lossing control

(Some of these are more fictional than others)

Customer: Hey, what happened to all of the controls for the generator?

Us: Can you let us know what you are trying to achieve?

C: I need to say how many digits are in the password for a site.

U: Does the site require a specific number of digits?

C: I want 4 digits in my password.

U: Does our default generated password not work on the site?

C: Why did you take the controls away? I need them.

Starting with a digit

(Some of these conversations are less fictional than others)

C: How much stronger would the generator be if it required that passwords start with a digit?

U: It would actually be weaker. By a lot.

C: Really?

U: We understand why you might think otherwise, but it turns out that the kinds of things people have been told to make their passwords strong would make weaker passwords when told to a password generator. Here’s some math.

C: Cool beans! Thanks guys.

Seeing stars

Some conversations are internal
(And why we need you all to do user studies)

Team member 1: If we only used the single most commonly accepted symbol then we will have much more compatibility. The strength reduction is not significant at these lengths.

TM2: It won’t show up enough times and will look weak.

TM1: Ok, here are the five most broadly accepted symbols. The “*” is less commonly accepted than any of them. We don’t need it.

TM2: The “*” is much more visibly salient than the other symbols and so people are more likely to notice that there are symbols.

This discussion played out in public in https://github.com/1Password/spg/issues/14 and https://github.com/1Password/spg/pull/22

Mistraining users

Precedent is hard to break

TM1: Saying “at least one digit” gives stronger passwords than saying “exactly three.”

TM2: For more than a decade we’ve given users a slider that allows them to ramp up the number of digits, implying the more the better.

TM1: Eww. Well I guess it’s time to try to untrain them.

TM2: Yep. We don’t want to be having this conversation 10 years from now.

Gambler's fallacy?

If it looks too weird it may not be compatible

TM2: 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.

Intuition based on years of customer interaction and reading the research is a fine thing, but it tells us that it will take a while for users to stop worrying and learn to love the generator

Jeffrey Goldberg
Principal Security Architect
1Password
jeff@1Password.com

Mitchell Cohen
Director, Product Management
1Password
mitchell@1Password.com