Pro-activists
Alistair Miles - 2125558M
Sigrid Rein Trustrup - 2369486R
Joseph Cameron - 2117625C

A Password Checker for Minors

Children aged 5 -15 are using internet connected devices more frequently now than ever [Ofcom 2017]. As internet access becomes easier for children it becomes more important that considerations are made for them with regard to usable security. User experience and training make a considerable difference in how well a user complies with security guidelines [Adams and Sasse 1999], but children, especially as young as 5, won't have this experience which may lead them to learn bad habits or disregard security entirely. To teach minors good security habits and help them construct secure passwords which are the foundation of secure authentication online, we have designed a pro-active password that guides the user towards a secure password.

Usage Context

This proactive password checker is designed to act as a stand-alone service, where children can instantly gain feedback on their password security. This feedback will include not just security recommendations, such as avoiding common passwords etc., but also usability recommendations, such as a warning for a long password that may be too hard to remember. Along with the password recommendation information located below the password text box, these usability recommendations shall hopefully discourage bad security habits, such as writing passwords down if the password is too long to remember.

Furthermore, as this is a password checker for minors, it is vital to effectively communicate feedback in a child-friendly manner that avoids too many confusing and technical terms that may intimidate a child. Instead, feedback should be concise, clear and simple. Also, it is well known that children typically have a shorter attention span than adults [Neville 2007]. Therefore, it is not acceptable to rely on static written rules when trying to motivate a minor to develop a secure password. As a result, the feedback should be instantaneous and dynamic as the user is typing to further enable interactivity and promote a subconscious reinforcement of good security practices.

User Interface

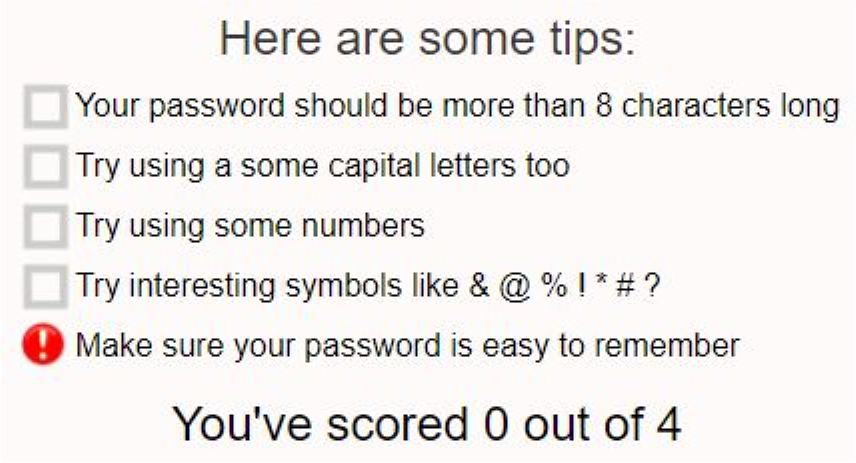
Particular care was taken when designing a clear and usable interface. The following criteria were taken into account:

- The varying importance of communications with the user
- The clarity and readability of any communication with the user
- The ease of interacting with any provided functionality

Messages that we show to the user must be easily understandable to young children so that even the youngest users are kept in-the-loop on the features that define a secure password. To do this we used encouraging and straightforward language in all of our messages, taking care to avoid technical terms like ‘special characters’ or ‘upper-case’ and complex words like ‘combination’. We also decided that colours were a suitable way of emphasising suggestions to aid communication, using the ‘red is bad, green is good’ colour scheme. One example of this is the messages beneath the password input, which range from bright red to bright green, directly communicating the scale of password strength to the user in a clear and intuitive way.

Particular care has been taken not to clutter the interface with information. There are many standards and guidelines for creating a strong password, but showing them to the user all at once would only distract them from the task. When using the password checker, the user is presented with a simple username and password input, and a condensed list of five basic requirements for a good password. This ensures the user is given at least the basic knowledge they require to start working on a password while not flooding them with so much information that they don’t know where to start.

The five basic requirements presented to the user are as shown:



Here are some tips:

□ Your password should be more than 8 characters long

□ Try using a some capital letters too

□ Try using some numbers

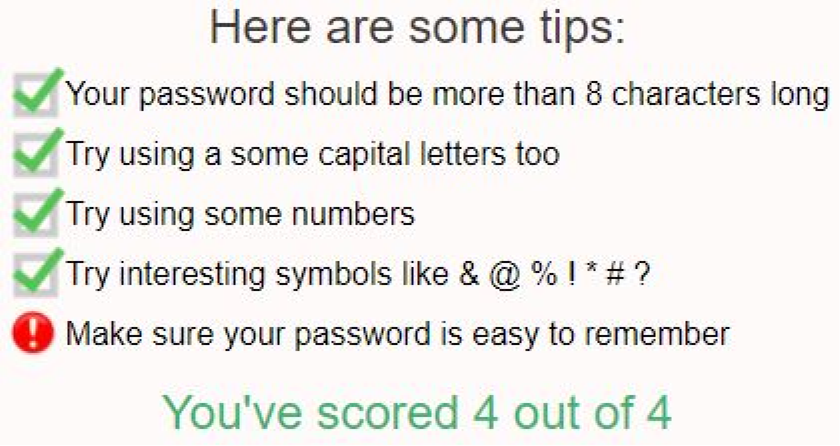
□ Try interesting symbols like & @ % ! * # ?

❗ Make sure your password is easy to remember

You've scored 0 out of 4

By checking the boxes next to each requirement and counting the number of requirements met, this adds a fun gamified element to creating a secure password. By challenging the user to meet as many requirements as possible, the user is encouraged to make a secure password without explicitly telling them they *must* use certain features to be secure. A user doesn’t need to meet every requirement for their password to be classified as strong, because explicitly enforcing too many rules may discourage the user [Beautement et al. 2008]. Instead, while meeting 3 out of 4 of the requirements gets the user a strong password, they’re still encouraged to improve it to get the maximum score.

As each requirement is met, a green tick is displayed next to it, and once they're all satisfied, the score turns green too. This is in keeping with the colour scheme used to convey password strength to not confuse the user.



Here are some tips:

✓ Your password should be more than 8 characters long
✓ Try using a some capital letters too
✓ Try using some numbers
✓ Try interesting symbols like & @ % ! * # ?
! Make sure your password is easy to remember

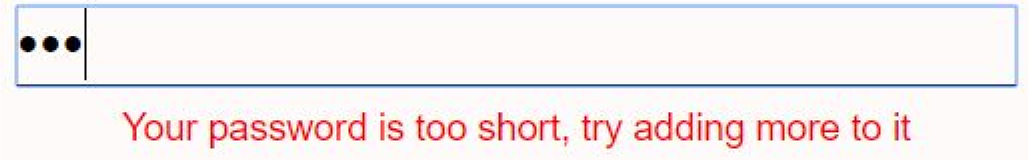
You've scored 4 out of 4

Assessing Password Strength

When assessing password strength, both technical factors and usability factors were considered.

However, before the password strength could be evaluated, the checker first checks for the absence of fundamental security flaws.

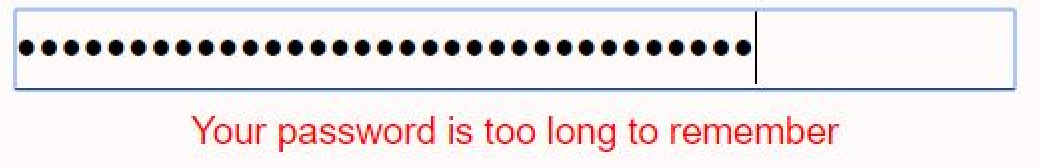
For example, if an entered password is shorter than 8 characters, a message encouraging the use of more characters is displayed, as recommended by NIST guidelines [Grassi et al. 2017] [Wisniewski 2016].



●●●

Your password is too short, try adding more to it

Conversely, if an entered password is longer than 32 characters, a message warning of its length is displayed.

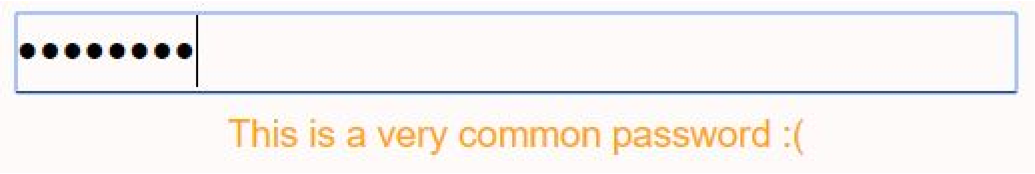


●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●

Your password is too long to remember

This message hints that the password, although being potentially very secure, may be too difficult to remember, hence, it also discourages the extremely insecure practice of writing passwords down.

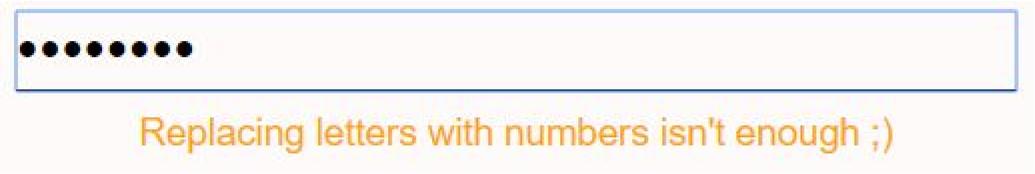
If the entered password matches a password from SecLists' set of common passwords [Haddix and Messier 2018], a warning message is displayed. We use a trie data structure that means even if our list of common passwords is thousands of entries long, lookup is perceptually instant.



●●●●●●●●

This is a very common password :(

If common number-letter replacements are used (e.g. '5' replacing 's') [Bishop and Klein 1995], the following message is displayed.



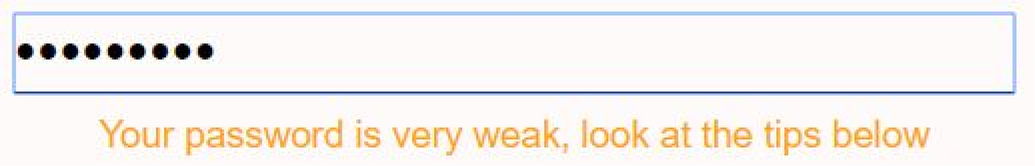
●●●●●●●●

Replacing letters with numbers isn't enough ;)

Once these basic flaws are confirmed to be non-existent, the strength of a password is calculated. Initially, before checking, the strength of a password is assumed to be 0. From that baseline, the final strength is then determined by the presence of the following properties within the password.

If an entered password is 8 characters or longer, the password strength is increased by 1. If an entered password contains a mixture of lower and upper case characters, the password strength is increased by 1. If an entered password contains a mixture of alphabetic characters and numeric characters, the password strength is increased by 1. If an entered password contains a mixture of alphanumeric characters and special characters (symbols such as !@£ etc.), the password strength is increased by 1. Finally, if an entered password contains at least two special characters, the password strength is increased by 1.

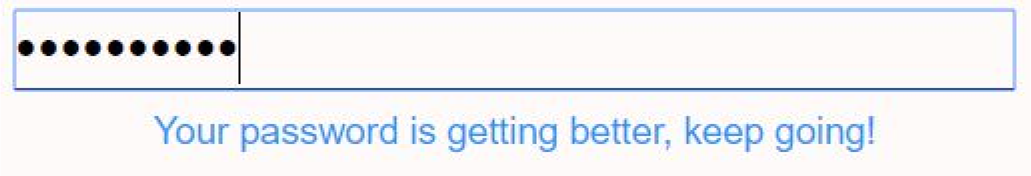
Once these factors have been analysed, the displayed feedback message directly corresponds to the numerical password strength score that has been calculated. A password strength that is less than 2 results in the following feedback message:



●●●●●●●●●

Your password is very weak, look at the tips below

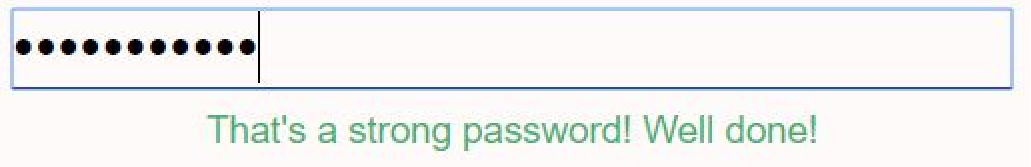
A password strength that is equal to 2 results in the following feedback message:



●●●●●●●●●●

Your password is getting better, keep going!

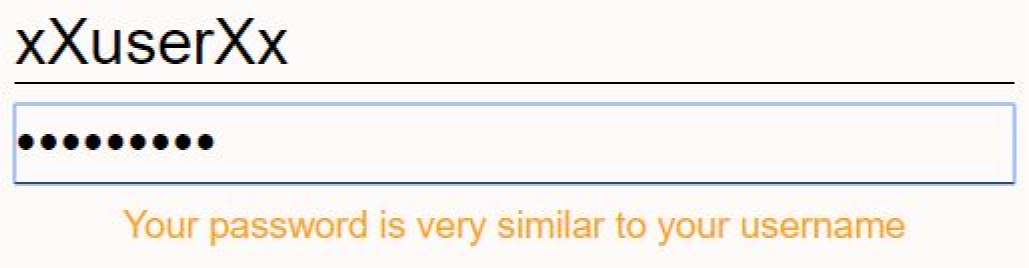
A password strength that is greater than 2 results in the following feedback message:



●●●●●●●●●●●

That's a strong password! Well done!

As an added precaution, users are able to enter their username as well as their password. This lets our application check for the similarity between the two [Bishop and Klein 1995]. We've implemented a longest common subsequence algorithm that allows us to compare two strings. If a username and password are more than 50% similar (contain more than 50% of the same characters in the same order) the following message is shown:



xXuserXx

●●●●●●●●●

Your password is very similar to your username

Test Cases

User Reasoning	Entered Password	Output
A user thinks they'll enter an obvious password with a 'hiding in plain sight' mentality	password	This is a very common password :(

A user tries to increase their score by replacing some letters with numbers	pa5sw0rd	Replacing letters with numbers isn't enough ;)
Starting from scratch, a user follows the first step of the recommended tips	cooldinosaur	Your password is very weak, look at the tips below
Continuing from the first tip, a user tries to follow the second tip	CoolDinosaur	Your password is getting better, keep going!
Now filled with motivation, a user adds some numbers to their password	C00lDinosaur2001	That's a strong password! Well done!
Now determined to see what it would take to get the maximum score, a user adds some special characters	#C00lDinosaur200!	That's a strong password! Well done! and You've scored 4 out of 4
A user randomly hits keys in an attempt to try and increase their score	dsfajisdnfoiua1234dnfalwemn123kjnllo9ufuol4	Your password is too long to remember
A user makes no attempt to engage and tries a rude word as their password	butts	Your password is too short, try adding more to it
In an effort to make a very easy-to-remember password, a user tries to make a small variation on their username	Username: xXuserXx Password: XuserX123	Your password is very similar to your username
A user tries to turn a common password into something the system will accept	Password123!	That's a strong password! Well done! and You've scored 4 out of 4

As shown, the last test case fails as the user receives praise for their strong password, and manages to get the maximum score when in reality the password entered is extremely weak. To catch cases like this, more complex algorithms would be required to detect how spread out the features of a password are. However, implementing this would also require communicating the issue to the user. This would be particularly challenging, as simple vocabulary to describe the issue of badly distributed features doesn't exist. Furthermore, the addition of complicated rules such as special character placement or upper case character placement may bring a child closer to the compliance threshold [Beautement et al. 2008], which may negatively impact on a child's experience of employing good security practices with the application.

A more reasonable, but still exploitable solution would be to increase the number of common passwords stored that the checker checks against. This would ensure that, even if an insecure password meets all the requirements, it would likely be detected as a common password.

Challenges

Finding relevant literature regarding password checkers aimed at communicating with children was a challenge. There was little research to be found on the subject of how children think when faced with creating passwords. However, there has been an enormous amount of research produced on how growing up with technology affects children. The immersion in the use of technology many children born today experience help them acquire decent to fairly advanced technology competencies[Chaudron 2015]. Taking this into account, the password checker follows well-established standards with regard to the design of familiar registration/login pages. Furthermore, the fact that ”Children learn from observation”[Chaudron 2015], indicates that following standards like this are a safe way to think of design in relation to children as well.

Another challenge was deciding on how to best phrase the feedback displayed by the password checker. The challenge lay in conveying the right message and simultaneously considering that the intended users are children. To achieve this, more difficult words were swapped with more child-friendly words that children are more likely to have heard in an educational context. E.g. capital letters and numbers rather than upper-case letters and integers/digits. Furthermore, we decided to have the dynamic checklist to support the written feedback that will give children who have limited vocabulary or are weak readers a helping hand by visualising the strength.

Deciding on how to thoroughly check a password was also a challenge. The recommendations NIST [Grassi et al. 2017] have for creating strong passwords were used as a template for this password checker. These recommendations include checking for uppercase letters, lowercase letters, digits, symbols, and also checking that the password is 8 or more characters. This password checker restricts the password selection process further by not allowing commonly used passwords or common letter to number substitutions, e.g. i as 1 and s as 5. Keeping in mind that there is a limit to how much a user is willing to comply with many strict rules[Beautement et al. 2008], the password checker does not force the user to comply with all of the rules mentioned above. Rather, it allows the user exclude one of the character sets. E.g. non-use of uppercase letters: “cat2cute*”.

Set-Up

The password checker is provided as an HTML page with functionality implemented in a javascript file. To use the application all a user needs to do is open the ‘checker.html’ file in any modern web browser.

Workload Report

Every member of the team worked on both the implementation of the password checker and subsequently this report. All team members contributed equally.

Bibliography

Adams, A. and Sasse, M.A. 1999. Users are not the enemy. *Communications of the ACM 42*, 12, 40–46.

Beautement, A., Angela Sasse, M., and Wonham, M. 2008. The compliance budget. *Proceedings of the 2008 workshop on New security paradigms - NSPW ’08*.

Bishop, M. and Klein, D.V. 1995. Improving system security via proactive password checking. *Computers & Security 14*, 3, 233–249.

Chaudron, S. 2015. *Young Children (0-8) and Digital Technology: A Qualitative Exploratory Study Across Seven Countries*. Publications Office of the European Union.

Grassi, P.A., Fenton, J.L., Newton, E.M., et al. 2017. *NIST Special Publication 800-63B*. National Institute of Standards and Technology.

Haddix, J. and Messier, D. 2018. SecLists. *GitHub*. https://github.com/danielmiessler/SecLists.

Neville, H.F. 2007. *Is This a Phase?: Child Development & Parent Strategies from Birth to 6 Years*. Parenting Press, Inc.

Ofcom. 2017. *Children and Parents: Media Use and Attitudes Report*. Ofcom.

Wisniewski, C. 2016. NIST’s new password rules – what you need to know. *Naked Security*. https://nakedsecurity.sophos.com/2016/08/18/nists-new-password-rules-what-you-need-to-know/.