Tails - Feature #7002
Hint user about passphrase strength when creating a persistent volume

03/31/2014 09:38 AM - tmc

<table>
<thead>
<tr>
<th>Status: In Progress</th>
<th>Start date: 03/31/2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority: Low</td>
<td>Due date:</td>
</tr>
<tr>
<td>Assignee: touss</td>
<td>% Done: 0%</td>
</tr>
<tr>
<td>Category: Persistence</td>
<td>Estimated time: 0.00 hour</td>
</tr>
<tr>
<td>Target version: Tails_4.4</td>
<td>Spent time: 0.00 hour</td>
</tr>
<tr>
<td>Type of work: Code</td>
<td>Affected tool:</td>
</tr>
</tbody>
</table>

Description
The GUI bits should be stolen from existing, well-thought solutions to the same problem, e.g. GNOME Disks.

The (Modern Perl) code lives at https://git-tails.immerda.ch/persistence-setup/.

Feel free to ask details to intrigeri if you want to tackle this, or are lost in the code.

@sajolida wants to track this :)

Related issues:
Related to Tails - Feature #7001: Hint user about the strength of their adminstration password
Confirmed 03/31/2014
Related to Tails - Feature #10647: Integrate information about persistence to persistence assistant
Confirmed 11/24/2015
Duplicated by Tails - Feature #15027: Passphrase Strength Meter for Persistence Wizard
Duplicate 12/08/2017

History
#1 - 03/31/2014 09:50 AM - intrigeri
- Subject changed from Password quality monitor for Tails persistent volume assistant UI to Hint user about passphrase strength when creating a persistent volume
- Description updated
- Category set to Persistence
- Status changed from New to Confirmed
- Type of work changed from User interface design to Code
- Starter changed from No to Yes

#2 - 03/31/2014 09:50 AM - intrigeri
- Related to Feature #7001: Hint user about the strength of their administration password added

#3 - 06/30/2017 03:43 PM - Anonymous
- Related to Feature #10647: Integrate information about persistence to persistence assistant added

#4 - 12/09/2017 06:02 AM - intrigeri
- Duplicated by Feature #15027: Passphrase Strength Meter for Persistence Wizard added

#5 - 10/18/2019 03:45 AM - touss
- Assignee set to touss

#6 - 11/23/2019 10:52 PM - touss

02/20/2020
Hi @intrigeri, I would appreciate some advice on this:
I tried to make the GUI as similar as possible with to that of the GNOME Disks. It uses Data::Password::zxcvbn to estimate the given passphrase strength (is this ok or should we use another module?). I'm not sure of the position of the warning label; would it be better to align it with the level bar?
I also added an extra commit to make mouse clicks not grab focus in Show passphrase button.

It would be helpful if you could add a screenshot of that, and then we can ask @sajolida for his opinion.

Hi, @segfault thanks for working on this!

touss wrote:

I'm not sure of the position of the warning label; would it be better to align it with the level bar?

Hi, @segfault thanks for replying.
This feature is currently implemented like image1. I suggested aligning the warning area like image2; Or would it be better to change the position of the level bar?

Cool! I'm not worried so much about the alignments of the different elements but I prefer image2.png.

I'm a bit more worried about the quality of the indication itself.

Password, password strength, and password policies are a really hot topic in usable security.

Just to give one example: https://blog.fleetsmith.com/password-security-guide/.

* I like the fact that the algorithm seems to detect weak l33t-speak and not pretend than "p4sw0rd" is stronger than "password".
I like the fact that the algorithm doesn't force the use of numbers or special characters, for example for diceware passwords.

I'm a bit concerned by the strength levels. For example "password1234" gives me "Good" in GNOME Disks and "password1234!" gives me "Strong", cf. "I Added '!' at the End to Make It Secure" :) (https://www.usenix.org/system/files/conference/soups2015/soups15-paper-ur.pdf)

Could we higher the bar to "Good" and "Strong"?

---

#12 - 11/28/2019 07:28 PM - intrigeri

Hi touss & others!

First, thanks a lot for working on this!

Wrt. how much we like the passphrase length checking algorithm, I'm glad we have sajolida's input here. touss, see the previous comments on this ticket (you were not on the list of watchers so you did not get notified of sajolida's questions).

A few notes from me:

- Depending on how much we feel this feature is important, some tools are readily available to us, while some others require more work (e.g. packaging new stuff for Debian). Of course, they won't yield the same feedback for a given passphrase.
- At first glance, I would be tempted to favor consistency over security. That is, it would feel weird to me to have the same password give different feedback here than in GNOME Disks.
- I'd like us to keep in mind that we currently have no passphrase check at all. Arguably, even the worst passphrase checker will provide better educated info than what our average user would guess by themselves: at least it'll warn against lots of obviously bad ideas. IMO this outweigh the risk for giving a false sense of security, but I may be wrong.
- GNOME Disks uses libpwquality. I don't know how good it is, but if we want to be consistent with GNOME Disks, I have a few question for touss:
  - Are there Perl bindings available for libpwquality?
  - Failing that, does libpwquality provide gobject-introspection data? If it does, we can use Gobject-Introspection to use this library.
  - Failing all of the above, I see Buster has libpwquality-tools, which ships a pwscore CLI. I optimistically assume we could use it to provide feedback similar to GNOME Disks'.

  It uses Data::Password::zxcvbn to estimate the given passphrase strength (is this ok or should we use another module?).

I did not evaluate the quality of the feedback provided by this library; I'm inclined to trust sajolida's input on this.

Upstream seems adequately maintained and the code looks sane.

The only problem I see is that this module is not in Debian, which is a requirement before we can use it; all the dependencies seem to be there though (at first glance). I'm a Debian developer myself and a member of the Debian Perl Group, so adding this module to Debian should not take more than a couple hours to me, if we decide 1. that's really the module we want; 2. it's worth this sort of effort. Then, maintenance is done collectively over there; putting a few hours/year into it would be the minimum; sky is the limit wrt. the maximum :) I'm fine with doing this work myself but I'd rather mentor touss to join the Debian Perl Group and do it themselves. It's a very nice group of people and the most active ones are very Tails-friendly, touss, what do you think?

I see libcrypt-cracklib-perl is in Buster already. Would it do the job?

Perhaps you could explain us why you picked this module specifically?
Now that this Pandora's box has been opened, it appears that it raises too many questions for this branch to make it into 4.1 ⇒ postponing. Still, I'm confident we'll manage to avoid the best becoming the enemy of the good :)

---

#14 - 11/29/2019 11:14 AM - sajolida

- At first glance, I would be tempted to favor consistency over security. That is, it would feel weird to me to have the same password give different feedback here than in GNOME Disks.

From our VeraCrypt survey, 41% of our users in 2018 didn't even know what GNOME Disks was:

https://tails.boum.org/blueprint/veracrypt/

I'm tempted to think that a lot of the ones who did know what GNOME Disks was came from the 45% of Linux users in our user base.

As we make Tails easier to use for Windows and macOS users, I expect only a minority of our target user base to be ever confronted with GNOME Disks years from now.

Another consistency that we should be worried about is between the advice that we give in our doc and what the software tells people. We're currently saying in /install/win/usb#create-persistence:

« We recommend choosing a long passphrase made of five to seven random words. »

Which is probably the shortest good advice that we can give. It also matched the NIST guidelines, referenced from the fleetsmith.com article linked above.

- I'd like us to keep in mind that we currently have no passphrase check at all. Arguably, even the worst passphrase checker will provide better educated info than what our average user would guess by themselves: at least it'll warn against lots of obviously bad ideas. IMO this outweigh the risk for giving a false sense of security, but I may be wrong.

I'm not so sure. Right now, when someone types in "password1234" we don't tell them anything. With the meter from GNOME Disks, we would explicitly tell them that it's a good password.

I think that it's not a good password in the context of Tails and we should probably not advocate for any simple combination of any 2 of the Top 500 passwords:

https://www.symantec.com/connect/blogs/top-500-worst-passwords-all-time

- GNOME Disks uses libpwquality. I don't know how good it is,
I tried zxcvbn (what a name!!!) from the python3-zxcvbn package and it tells me that "password1234" is "a very common password". I like that.

Sooooo, I would propose to:

- Do some more tests with zxcvbn, see how it compares to libpwquality, and whether we think that its results make sense in the context of Tails.

- If so, advocate for using zxcvbn in GNOME Disks as well. I haven't checked thoroughly but I would guess that this password meter is used only on the passphrase of encrypted volumes. Then it would make sense for GNOME Disks to be a bit more strict. I see several bindings for it in Debian and we might even consider writing a patch to GNOME Disks upstream. Until then, the meters in Tails and GNOME Disks might be inconsistent but I would prefer that to advocating for "password1234".

- If not, maybe we should give up on having a meter in the first place and instead move our advice from our doc to the Persistence creation app.

I hope that I'm not being too much of a downer here @touss. It's so cool to see you working on the Persistence app, it's one of our key software and we could improve it so much!

#15 - 11/29/2019 12:12 PM - sajolida
- Description updated

#16 - 01/06/2020 03:16 AM - touss
- File libpwquality_strenght_distribution.png added
- File top500combined.png added

Hi @intrigeri and @sajolida, thanks for all the questions/information provided and sorry for the late reply.

Could we higher the bar to "Good" and "Strong"?

You mean make the bar green for Good and Strong?

- Are there Perl bindings available for libpwquality?

I didn't find any perl bindings available for libpwquality.
Failing that, does libpwquality provide gobject-introspection data? If it does, we can use Gobject-Introspection to use this library.

Cool, I was not aware of this possibility, but it doesn't provide.

Failing all of the above, I see Buster has libpwquality-tools, which ships a pwscore CLI. I optimistically assume we could use it to provide feedback similar to GNOME Disks'.

I made some tests and I think that's possible to use.

I'm fine with doing this work myself but I'd rather mentor touss to join the Debian Perl Group and do it themselves. It's a very nice group of people and the most active ones are very Tails-friendly. touss, what do you think?

This would be awesome! I would love to work on this

I see libcrypt-cracklib-perl is in Buster already. Would it do the job?

The problem is that this library doesn't have a function that returns a password score like libpwquality and zxcvbn has. This won't help to implement the strength bar.

Perhaps you could explain us why you picked this module specifically?

Because I couldn't find any Perl bindings for libpwquality, I started to look for a Perl module that could replace libpwquality. Then I came across with the David Farrel article about zxcvbn, that led me to the Daniel Wheller paper and a few other documents that suggested me that it could be worth the try at least for testing purposes (the password_strength function in zxcvbn returns a score key that made it easier to implement the strength bar).

I couldn't find any information about the algorithm used by libpwquality in their GitHub repo, though I didn't read the code.

I took a look at the source code and found that Gnome Disks uses the default values provide by libpwquality that resulted in the following checks:
1 check if the password is palindrome
2 check if the password has minlen 8
3 whether the password (with possible modifications) matches a word in a dictionary. Currently the dictionary check is performed using the cracklib library.
then it calculates the password score if it passes in these tests: https://github.com/libpwquality/libpwquality/blob/master/src/check.c#L592 although I didn't figure out exactly how it works

I could find an article comparing the password meter from different online services and the Dropbox algorithm seems to be among the strictest ones: https://www.ndss-symposium.org/wp-content/uploads/2017/09/06_3_1.pdf

To compare the results obtained by zxcvbn in this paper I wrote a script to see the strength label that Gnome Disks would assign for some of the dictionaries tested in this paper:
I also tested what strength label we would get for every 2 of the top 500 passwords combined for libpwquality, zxcvbn and got the following result:

Sooooo, I would propose to:

- Do some more tests with zxcvbn, see how it compares to libpwquality, and whether we think that its results make sense in the context of Tails.

What more tests we can do?

I hope that I'm not being too much of a downer here @touss. It's so cool to see you working on the Persistence app, it's one of our key software and we could improve it so much!

Far from it, i'm learning a lot and it's been very exciting. Thanks for sharing your knowledge!
I also tested what strength label we would get for every 2 of the top 500 passwords combined for libpwquality.zxcbn

Here is the list that I used:
https://salsa.debian.org/touss-guest/pwcheck/raw/master/500-worst-passwords-combined.txt

#18 - 01/07/2020 06:00 PM - CyrilBrulebois

- Target version changed from Tails_4.2 to Tails_4.3

#19 - 02/11/2020 03:25 PM - anonym

- Target version changed from Tails_4.3 to Tails_4.4

Files

<table>
<thead>
<tr>
<th>File Name</th>
<th>Size</th>
<th>Date</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>image1.png</td>
<td>81.2 KB</td>
<td>11/28/2019</td>
<td>touss</td>
</tr>
<tr>
<td>image2.png</td>
<td>88.7 KB</td>
<td>11/28/2019</td>
<td>touss</td>
</tr>
<tr>
<td>libpwquality_strength_distribution.png</td>
<td>27.4 KB</td>
<td>01/06/2020</td>
<td>touss</td>
</tr>
<tr>
<td>top500combined.png</td>
<td>18 KB</td>
<td>01/06/2020</td>
<td>touss</td>
</tr>
</tbody>
</table>