Tails - Bug #12093
Feature # 5464 (Resolved): Revamp Greeter interface
Feature # 8230 (Resolved): Greeter revamp: Phase 1

Missing "Read only" option for persistence in new Greeter
12/28/2016 07:42 AM - intrigeri

<table>
<thead>
<tr>
<th>Status:</th>
<th>Rejected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Elevated</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Category:</td>
<td>Persistence</td>
</tr>
<tr>
<td>Target version:</td>
<td>Tails_3.0</td>
</tr>
<tr>
<td>Feature Branch:</td>
<td></td>
</tr>
<tr>
<td>Type of work:</td>
<td>Discuss</td>
</tr>
<tr>
<td>Blueprint:</td>
<td></td>
</tr>
</tbody>
</table>

**Start date:** 12/28/2016  
**Due date:**  
**% Done:** 100%  
**Estimated time:** 0.00 hour  
**Spent time:** 0.00 hour  
**Assignee:**  
**% Done:** 100%  
**Estimated time:** 0.00 hour  
**Spent time:** 0.00 hour  
**Category:** Persistence  
**Estimated time:** 0.00 hour  
**Spent time:** 0.00 hour  
**Affected tool:** Greeter  

**Description**
As the attached video shows, the new Greeter doesn't seem to allow unlocking a persistent volume in read-only mode. Is that by design and on purpose?

**Related issues:**
- Related to Tails - Feature #12055: Update test suite for Greeter revamp, phase 1  
  Resolved  12/21/2016  
- Related to Tails - Feature #12361: Document the removal of read-only persistence  
  Resolved  03/17/2017  

**Revision 90b8b989 - 01/30/2017 12:14 PM - anonym**
Test suite: drop usage and tests of read-only persistence.

We won't have it in Tails 3.0–beta1 since the Greeter doesn't have that option, and it's not even sure we'll reintroduce it since it's apparently quite buggy and not widely used.

Will-fix: #12055  
Refs: #12093  

**History**

**#1 - 12/28/2016 07:42 AM - intrigeri**
- Blocks Feature #12055: Update test suite for Greeter revamp, phase 1 added

**#2 - 12/31/2016 05:47 PM - alant**
intrigeri wrote:

As the attached video shows, the new Greeter doesn't seem to allow unlocking a persistent volume in read-only mode. Is that by design and on purpose?

That's a good question... and I think that the answer is unclear. Read-only persistence was not in the final mockups that were agreed for implementation (https://tails.boum.org/blueprint/greeter_revamp_UI/design_rationale_phase1/#index2h3):

Greeter.Encrypted.Storage.png

But it was not totally removed by design. The clearest messages I found about it are a Tails-UX discussion of september 2014 (https://tails.boum.org/blueprint/greeter_revamp_UI/design_rationale_phase1/#index2h3):  

Greeter.Encrypted.Storage.png

03/12/2020
And I suggest again to move the "Read only" option elsewhere (in advanced options). I don't know anybody using it.

We already dismissed that option because there are real usecases (e.g. needing to have access to its keyrings but wanting to keep the option to shutdown Tails by removing the USB stick without loosing data).

Having "real usecases" for some option is not a good argument for having it on the first screen. There are serious use cases for all the options in the advanced screen as well. The question here is rather a question of priority: we need to find a balance between having as little things on the first screen as possible, but still cover the vast majority of use cases.

I'm pretty sure that the "read-only" option is less popular than the "Windows camouflage" for example, or the "Tor bridge mode".

And regarding "prioritizing the best impact", I think that the read-only option fits in the category of "few users frequently" at most. And for me that goes on the "advanced" screen.

My only doubt is that it would be an option related to persistence that would appear in other place than the "Enable persistence" options... But I believe that this slight argument against this proposal is counterbalanced by the advantage of getting a more straightforward screen for the vast majority of uses.

So we could add this to advanced options in case there is persistence in the media, something like "Read-only encrypted storage : When encrypted storage is unlocken, only allow read-only access to it" or something like that...

What do you think?
Read-only persistence was not in the final mockups that were agreed for implementation

OK, good to know.

But it was not totally removed by design.

"Interesting".

So we could add this to advanced options in case there is persistence in the media, something like "Read-only encrypted storage : When encrypted storage is unlocken, only allow read-only access to it" or something like that...

What do you think?

I'm not going to engage in this UX discussion here, so I'm going to raise the question on tails-ux@.
Given [https://mailman.boum.org/pipermail/tails-ux/2017-January/003331.html](https://mailman.boum.org/pipermail/tails-ux/2017-January/003331.html) we won't block on this for #12055 as far as 3.0~beta1 is concerned. Deadline for a final decision (+ consensus on mockups if applicable) is March 16.

#7 - 01/30/2017 11:02 AM - intrigeri

- Priority changed from High to Normal

#8 - 01/30/2017 11:02 AM - intrigeri

- Priority changed from High to Normal

#9 - 03/17/2017 12:17 PM - intrigeri

- Priority changed from Normal to Elevated

#10 - 03/17/2017 05:36 PM - alant

- Status changed from Confirmed to Rejected
- Assignee deleted (intrigeri)

sajolida wrote on tails-ux mailing list:

My gut feeling was that we could remove it and focus on issues that are affecting better identified scenarios and more users.

Still, I did some stats on the WhisperBack reports received in 2016:

- 515 had persistence read-write (98%)
- 11 had persistence read-only (2%)
- 857 had no persistence

4 reports were from m...@r...:

- a0e7629eb892236b0e07ebfa782111__-__2016-08-20_1756.asc
  Explaining that she uses read-only persistence because some months ago she lost some text in Writer with a read-write persistence and is now scared about losing text again.
  Later on in the thread she reaches the conclusion that she probably had a faulty SD card or messed up with her setup.
- 1518328556a7d72ae059327d7dc168c4c__-__2016-08-20_1823.asc
  Asking for how to download files from the Unsafe Browser.
- 8496e755d887e6b165c694967e5a42__-__2016-08-20_1830.asc
  Reporting that Icedove does not work with accents in passwords.
  I wouldn't be surprised if Icedove behaved in a weird way when it's file system is read-only. The user stopped answering the thread and didn't lead the ticket to resolution.
- b25ef0f1966d668be776707676f1978b__-__2016-08-20_1807.asc
  Reporting that the upgrade check files with read-only persistence. I couldn't find an answer from help desk.

2 reports were from l...@y...:

- 74c7c775a98619acbc087d1166cb6299__-__2016-01-13_1524.asc
  Reporting an error while upgrading to 2.0 RC1
- bee48a231db757dd227c65f235a8f6161__-__2016-02-13_1555.asc
  Reporting about failed upgrade checks. Unclear if it was related to read-only persistence.

And the 5 other reports:

- 2627e58e07c8eacbb72712169491ddd__-__2016-11-25_1238.asc
  Someone reporting a bug with the Places menu when in read-only.
- 45199ebbf40252424053e17e52a867a7__-__2016-05-31_0847.asc
  Someone asking about how to save the timezone.
- 6cc189b9a2bedac0e6395ccc5d8fedb__-__2016-05-22_0113.asc
  Someone reporting that KeePassX is not read by Orca.
- 9fbe3576bb54572887b2698e05801a5__-__2016-11-23_0304.asc
  Someone reporting about the absence of on-screen keyboard in Greeter.
- f94fb33dd617191b4d0b4995327c82__-__2016-03-25_0621.asc
  Reporting problems with super custom persistence: ~/.config, ~/.cache.
So from these 11 reports, 4 were from someone using read-only when she really shouldn't (m...@r...), 3 were about stuff that otherwise work well with read-write persistence (74c7c75, bee48a2, 2627ef5), 1 was from someone who's trying hard to shoot herself in the foot (f94fb83).

We're left with 3 reports that would otherwise come from people who might have a good reason to be in read-only and are not complaining about its malfunctioning. I didn't take the time to sanitize as well the number of reports from people with read-write persistence, so we cannot compare this number of 3 with anything else to judge the portion of users relying on read-only for good reasons. But seeing that:

- The most vocal user was actually using read-only while she actually shouldn't have and ran into other problems as a consequence.
- The absolute number of 3 is extremely small.
- We don't have solid user scenarios ourselves to defend this feature.

I'm also in favor of removing it for the time being.

intrigeri, spencer and me seems to aggre with this. I'm thus rejecting this issue.

---

#11 - 03/17/2017 05:47 PM - alant
- Related to Feature #12361: Document the removal of read-only persistence added

#12 - 04/23/2017 08:55 AM - sajolida
People who still need very badly to have a read-only persistence could still use a SD to USB adapter and a miniSD card which still have hardware read-only toggle like this one: https://images-na.ssl-images-amazon.com/images/I/4177OdJ0ZsL_.AC_US218_.jpg.

The extra cost is ~5€.

#13 - 04/24/2017 07:11 AM - intrigeri
People who still need very badly to have a read-only persistence could still use a SD to USB adapter and a miniSD card which still have hardware read-only toggle

I don't know how the software we ship behaves when it faces a filesystem mounted read-write, backed by storage that rejects writes. Some might work similarly to read-only persistence, some might just crash.

#14 - 04/25/2017 09:41 AM - anonym
- Status changed from Rejected to In Progress
intrigeri wrote:

People who still need very badly to have a read-only persistence could still use a SD to USB adapter and a miniSD card which still have hardware read-only toggle

I don't know how the software we ship behaves when it faces a filesystem mounted read-write, backed by storage that rejects writes. Some might work similarly to read-only persistence, some might just crash.

When I tested this in a VM (by adding <readonly/> to the USB disk's section) the final mount points (e.g. ~/Persistent, ~/.gnupg) were read-only †. This will work for some applications, but it will work really poorly with applications that constantly writes e.g. Icedove. I guess it could be argued that users shouldn't use such applications when read-only persistence is enabled, but we certainly don't document this. So, IMHO, we cannot advertise sajolida's workaround without trying to explain this problem, which is a bit complex. OTOH, we could quite easily improve the situation: if we detect that the read-only flag is set for the persistent medium, then we enable read-only persistence (note that this is something we could do now even if we never re-introduce the user-visible Greeter option for read-only persistence). In fact, one way to look at the read-only persistence feature is that it is needed to properly support read-only switches on the persistent media.

I'm reopening this ticket to check what you think about this solution. Thoughts?

† This is because the /sys/block/sda/ro flag is set to 1. If some real hardware has a read-only switch that doesn't set that flag I'd expect major breakage, but I'm quite hopeful we won't see that since this flag is something that storage manufacturers seem to get right (unlike the removable flag).

#15 - 04/26/2017 03:31 PM - sajolida
- Assignee set to sajolida

#16 - 04/27/2017 10:47 AM - intrigeri
Meta: we've received very, very little feedback from users who suffer from the removal of this option in the 3.0~ series. This tends to confirm that very few people are using it. In general I'd rather see us (Foundations team, UX folks) focus on matters that bring value to more users than this one, so I won't spend any time on it unless a project-wide decision says this feature is important, should be prioritized, with a clear definition of its goals, scope and support level. Sorry to be the spoilsport here.

#17 - 04/28/2017 12:47 PM - sajolida
- Status changed from In Progress to Rejected
- Assignee deleted (sajolida)
I fully agree with intrigeri. Though, in general, I'm not sure that we should rely only on negative feedback from beta versions to check the value of a feature but here it was not our only indicator (we analyzed WhisperBack reports as well) and I'm not sure what we could do best with our current ressources.

I proposed the idea of documenting SD to USB adapters as a cheap workaround but I'm dropping the idea since it seems to be more complicated that I thought initially.

So: rejecting again :)

#18 - 05/27/2017 07:06 AM - intrigeri
- % Done changed from 0 to 100

Files
02_25_52_Booting_Tails_from_a_USB_drive_with_an_enabled_persistent_partition_and_reconfiguring_it.mkv 797 KB 02/12/2016 intrigeri