I appreciate that Tails now includes native support for decrypting VeraCrypt volumes for end users.

However, there may be a legal risk in its current design which I would like to bring your attention to.

In Tails' Nautilus under 'Other Locations', it currently automatically labels a drive which may not actually be encrypted (e.g. a DBAN'ed drive with random data written on it) as 'X GB Encrypted'.

Key disclosure laws are very serious situations, with a lot of case law. Law enforcement in many jurisdictions can compel a citizen to unlock what they assert to be encrypted data based on evidence gathered by examining the user's computer equipment, and if the user doesn't give a password or denies the assertion of encryption, they are jailed.

Here are some links:


Apart from it being terrible for the user if they are jailed for denying a non-encrypted drive is encrypted, this may be a legal risk for Tails project.

If a user were put in this situation and then it was later shown that police were misled by the Tails user interface in asserting that a random data drive was encrypted, the user could plausibly sue Tails project for millions.

I guess a lawyer could demonstrate in court that another provably DBAN'ed drive (DBAN'ed right there in court) shows in Tails as 'X GB Encrypted' as well. But such situations are out of Tails’ control, influence, or even knowledge of such proceedings taking place. There may be liability issues.

I also think it is distressing to Tails users to see this false declaration of their drive being ‘Encrypted’ (before they unlock it), who rely on plausibly deniable encryption to protect their personal information.

There can be a simple fix. It should not say 'X GB Encrypted' but something like 'X GB Possibly Encrypted'.

Thanks
AnonymousUser

GVfs merge request: https://gitlab.gnome.org/GNOME/gvfs/merge_requests/55

**History**

#1 04/27/2019 03:16 AM - AnonymousUser

I now discovered another place that Tails also has this problem. Insert a USB which is wholly made up of a partitionless hidden VeraCrypt volume
(which is indistinguishable from other headerless random data), and Nautilus currently will list the USB stick (before mounting) as 'X GB Encrypted'. It may not be true at all, it again could be a non-encrypted USB stick but Tails falsely suggests it is.

---

#2 - 04/30/2019 05:43 PM - sajolida

- Subject changed from Potential issue of legal liability to Tails project with the current design of VeraCrypt support in GNOME (simple fix) to Inconsistent "Encrypted" (vs "Encrypted?") for locked VeraCrypt volumes in GNOME Files
- Category deleted (Accessibility)
- Status changed from New to Confirmed
- Priority changed from Elevated to Low
- Type of work changed from Accounting to Code

Right, we're marking locked VeraCrypt volumes as "Encrypted?" (and "Possibly Encrypted") in GNOME Disks but only as "Encrypted" in GNOME Files. See screenshot in attachment. Ideally, locked VeraCrypt volumes (and random data partitions) should be labelled as "Encrypted?" everywhere.

@segfault: Is there any reason for this inconsistency?

---

#3 - 05/01/2019 06:34 PM - segfault

- Assignee set to sajolida
- QA Check set to Info Needed

sajolida wrote:

Right, we're marking locked VeraCrypt volumes as "Encrypted?" (and "Possibly Encrypted") in GNOME Disks but only as "Encrypted" in GNOME Files. See screenshot in attachment. Ideally, locked VeraCrypt volumes (and random data partitions) should be labelled as "Encrypted?" everywhere.

@segfault: Is there any reason for this inconsistency?

The reason is that we kept the default behavior in gvfs, which is

```c
if (udisks_id_usage == "crypto") {
    volume->name = "{SIZE} Encrypted"
}
```

I could create a patch to do the following instead:

```c
if (udisks_id_usage == "crypto") {
    if (udisks_id_type == "crypto_unknown") {
        volume->name = "{SIZE} Encrypted?"
    } else {
        volume->name = "{SIZE} Encrypted"
    }
}
```

... if we decide that it's worth the effort. FWIW, gvfs upstream handled our last merge requests very nicely and in a timely manner, so I'm not expecting this to be a lot of effort.
Wow, you already came up with a patch?!?! :)  

It's very good to know that it seems simple to fix but I don't think that we should prioritize this outside of the process that we have for #14544.

---

#5 - 05/03/2019 11:16 AM - sajolida  
- Related to Feature #14544: Spend software developer time on smallish UX improvements added

---

#6 - 07/21/2019 12:37 PM - segfault  
- Assignee set to segfault

#7 - 07/21/2019 01:07 PM - segfault  
- Description updated

#8 - 09/08/2019 07:37 PM - segfault  
- Status changed from Confirmed to In Progress  
- Assignee deleted (segfault)  
- Target version set to Tails 5.0  
- Type of work changed from Code to Wait

The MR was merged into GVfs and released in 1.41.90. So this will be fixed in Tails 5.0. In case that we have to rebuild our custom GVfs package at some point during the 4.x cycle, I might backport the patch and include it, but I don't think it's worth the effort to rebuild and release a new package for this change only.

---

#9 - 09/09/2019 09:13 AM - intrigeri

The MR was merged into GVfs and released in 1.41.90. So this will be fixed in Tails 5.0.

Great!  

In case that we have to rebuild our custom GVfs package at some point during the 4.x cycle, I might backport the patch and include it, but I don't think it's worth the effort to rebuild and release a new package for this change only.

Agreed.