RTL8822be is not working anymore

12/07/2019 03:17 PM - goupille

Status: Resolved
Priority: Normal
Assignee:
Category: Hardware support
Target version: Tails_4.4
Feature Branch: bugfix/17323-fix-missing-rtw88-firmware
Type of work: Code
Blueprint:

Description

a user reported that this chipset, while working correctly in Tails 3.16, is not supported anymore: it seems that it is not even seen as a
wi-fi adapter.

Affected chip: 04:00.0 Network controller [0280]: Realtek Semiconductor Co., Ltd. RTL8822BE 802.11a/b/g/n/ac WiFi adapter [10ec:b822]

Related issues:

Blocks Tails - Feature #16209: Core work: Foundations Team

Associated revisions

Revision 87756515 - 03/07/2020 04:13 AM - CyrilBrulebois
Add tails-workarounds submodule (refs: #17323).

Revision 0ac1cc5a - 03/07/2020 05:37 PM - CyrilBrulebois
Add support for tails-workarounds (refs: #17323).

Make tails-workarounds directory available inside the chroot, so that its content can be used by any chroot local hooks.

Chroot local includes seem to be the usual way to transfer things from the Tails checkout to the chroot environment; explicitly cleaning up /tmp afterwards doesn't seem to be needed.

Revision 636fe9f2 - 03/07/2020 05:37 PM - CyrilBrulebois
Fix missing firmware for RTL8822BE/RTL8822CE (refs: #17323).

Use the tails-workarounds provided firmwares until the firmware-realtek package is updated with the patch by Sjoerd Simons (Debian#935969).

Revision 4357e918 - 03/08/2020 03:37 AM - CyrilBrulebois
Add tails-workarounds submodule (refs: #17323).

Revision 060605c0 - 03/08/2020 03:37 AM - CyrilBrulebois
Add support for tails-workarounds (refs: #17323).
Make tails-workarounds directory available inside the chroot, so that its content can be used by any chroot local hooks.

Chroot local includes seem to be the usual way to transfer things from the Tails checkout to the chroot environment; explicitly cleaning up /tmp afterwards doesn't seem to be needed.

Revision cfca4d4a - 03/08/2020 03:37 AM - CyrilBrulebois
Fix missing firmware for RTL8822BE/RTL8822CE (refs: #17323).

Use the tails-workarounds provided firmwares until the firmware-realtek package is updated with the patch by Sjoerd Simons (Debian#935969).

Revision 9c8eba6a - 03/08/2020 03:43 PM - intrigeri
Ensure we stop shipping RTL8822BE/RTL8822CE firmware once it's in Debian (refs: #17323)

Once those files are in Debian and they get updated there, if we don't notice this warning, we'll be shipping outdated firmware, which is not ideal.

I'm concerned that this warning will be hard to spot in the middle of a 15k-lines build log, that has 30+ other warnings which are all false positives.

So let's abort the build whenever we detect this situation.

Revision 8b18ac00 - 03/08/2020 04:08 PM - intrigeri
Merge branch 'bugfix/17323-fix-missing-rtw88-firmware' into stable (refs: #17323)

I'm not closing #17323 yet, because at this point we don't know if the missing firmware was the only problem.

History

#1 - 12/07/2019 03:17 PM - goupille
Bug report: 935ae832daa20a6ea1f44bdaf5b8c00d

#2 - 12/07/2019 03:32 PM - goupille
- Status changed from New to Confirmed

goupille wrote:
  Bug report: 935ae832daa20a6ea1f44bdaf5b8c00d

a second user reported this so I mark it as confirmed
This seems to be caused by a failure to load the firmware:

Dec 05 00:14:02 amnesia kernel: rtw_pci 0000:04:00.0: enabling device (0000 -> 0003)
Dec 05 00:14:02 amnesia kernel: rtw_pci 0000:04:00.0: firmware: failed to load rtw88/rtw8822b_fw.bin (-2)
Dec 05 00:14:02 amnesia kernel: firmware_class: See https://wiki.debian.org/Firmware for information about missing firmware
Dec 05 00:14:02 amnesia kernel: rtw_pci 0000:04:00.0: Direct firmware load for rtw88/rtw8822b_fw.bin failed with error -2
Dec 05 00:14:02 amnesia kernel: rtw_pci 0000:04:00.0: failed to request firmware
Dec 05 00:14:02 amnesia kernel: rtw_pci 0000:04:00.0: failed to load firmware
Dec 05 00:14:02 amnesia kernel: rtw_pci 0000:04:00.0: failed to setup chip efuse info
Dec 05 00:14:02 amnesia kernel: rtw_pci 0000:04:00.0: failed to setup chip information
Dec 05 00:14:02 amnesia kernel: rtw_pci: probe of 0000:04:00.0 failed with error -22

Note that the user for wb://935ae832daa20a6ea1f44bcfa5b8c00d had MAC spoofing disabled.

OK, that's tracked in Debian: https://bugs.debian.org/935969 and its 2 merged duplicates.

The missing firmware is present in the upstream Git: https://git.kernel.org/pub/scm/linux/kernel/git/firmware/linux-firmware.git/tree/rtw88.

My understanding is that the missing firmware should be added to https://salsa.debian.org/kernel-team/firmware-nonfree/blob/master/debian/config/realtek/defines.

I'm happy to treat this as FT work if it affects lots of Tails users.
#8 - 12/23/2019 01:34 PM - goupille

intrigeri wrote:

I'm happy to treat this as FT work if it affects lots of Tails users.

there were 3 reports during the first week after tails 4.1 and probably more since then

#9 - 12/28/2019 03:53 PM - intrigeri

- Blocks Feature #16209: Core work: Foundations Team added

#10 - 12/28/2019 03:56 PM - intrigeri

- Target version set to Tails_4.2
- Type of work changed from Debian to Wait

This was fixed in https://salsa.debian.org/kernel-team/firmware-nonfree but more packaging work is needed before it gets uploaded and we can upgrade the affected package. There's a small chance this happens in time for 4.2 but I'm not counting on it. Still, putting it on our 4.2 radar at least to check if the fix reached the Debian archive.

#11 - 01/06/2020 07:58 AM - intrigeri

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

#12 - 02/09/2020 08:34 AM - intrigeri

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

#13 - 03/01/2020 04:56 PM - CyrilBrulebois

Checked, and there's no upload yet.

How acceptable would it be to dump the needed files into our tree and put them in place through a local hook until the updated package reaches the archive? It's been several months already. :/

#14 - 03/02/2020 07:36 AM - intrigeri

Hi @CyrilBrulebois,

I'm glad you're caring!

How acceptable would it be to dump the needed files into our tree and put them in place through a local hook until the updated package reaches the archive? It's been several months already. :/

First, I'll share information about how we're handling this sort of hardware support issues in general, and the reason why. I do not mean it's the best and only way to approach this, nor that it's set in stone. My goal is to provide food for the collective & individual thought processes that we go through when picking a task to work on.

To put this into perspective, wrt. support for specific hardware devices, with my FT "frontdesk" hat on I see many issues reported every month. If they're tracked upstream already, in general we don't do anything ourselves besides documenting workarounds (if any is available). We could surely do something about some of these individual issues but the current capacity of the FT does not allow us to work on all of them, and it's often hard to
tell which ones matter more than others. So in general I try to focus our efforts on problems that we know affect a larger subset of our users (e.g. currently; the network adapter support regressions that seemingly were introduced in 4.1.1; the "Hot topics on our help desk" section of our monthly report is often useful to identify such higher-impact problems).

Now, for the case at hand, the workaround seems pretty cheap to implement, so even if the benefit is small, the cost/benefit could be worth it. I would not actively look for FT resources to put into this, but if that's something you'd like to fix, please go ahead! In any case, let's not spend more time debating "is it worth it?" than it would take to fix the bug :)

Implementation-wise, if possible I'd rather not add binary files to Git: it would bloat the Git repo forever and might have annoying licensing implications. Maybe we could use a build hook to wget/curl them from upstream + verify them against a hard-coded checksum? (That's essentially what we do for Tor Browser; that's what a couple packages we install do under the hood, e.g. libdvdcss & firmware-b43*, so it would not be a first.)

Cheers!

#15 - 03/02/2020 07:48 AM - intrigeri
- Related to Feature #17504: rtl8812bu added

#16 - 03/03/2020 04:54 PM - CyrilBrulebois
- Assignee set to CyrilBrulebois

#17 - 03/03/2020 05:00 PM - CyrilBrulebois
- Type of work changed from Wait to Code

intrigeri wrote:

First, I'll share information about how we're handling this sort of hardware support issues in general, and the reason why. I do not mean it's the best and only way to approach this, nor that it's set in stone. My goal is to provide food for the collective & individual thought processes that we go through when picking a task to work on.

Much appreciated.

To put this into perspective, wrt. support for specific hardware devices, with my FT "frontdesk" hat on I see many issues reported every month. If they're tracked upstream already, in general we don't do anything ourselves besides documenting workarounds (if any is available). We could surely do something about some of these individual issues but the current capacity of the FT does not allow us to work on all of them, and it's often hard to tell which ones matter more than others. So in general I try to focus our efforts on problems that we know affect a larger subset of our users (e.g. currently; the network adapter support regressions that seemingly were introduced in 4.1.1; the "Hot topics on our help desk" section of our monthly report is often useful to identify such higher-impact problems).

To make it explicit: my interest in this particular topic was prompted by the fact it indeed seemed like a regression introduced by the "hot fix" for Mac machines, and it seemed to affect several chipsets. That and my stupid "after-sales service" urges… :D

Now, for the case at hand, the workaround seems pretty cheap to implement, so even if the benefit is small, the cost/benefit could be worth it. I would not actively look for FT resources to put into this, but if that's something you'd like to fix, please go ahead! In any case, let's not spend more time debating "is it worth it?" than it would take to fix the bug :)

I think at this point, it wouldn't make sense for me to walk away, now that I have enough context to handle this properly, so that'll be some extra FT work I didn't plan but can happily do.

Implementation-wise, if possible I'd rather not add binary files to Git: it would bloat the Git repo forever and might have annoying licensing
implications. Maybe we could use a build hook to wget/curl them from upstream + verify them against a hard-coded checksum? (That's essentially what we do for Tor Browser; that's what a couple packages we install do under the hood, e.g. libdvdcss & firmware-b43*, so it would not be a first.)

ACK. I was definitely fishing for ideas, and adding binary stuff to the git repository made me twitch when I wrote it, but I thought you'd get the general idea. :)

And given it's an interim solution, no need to mirror stuff on our website like we do for Tor Browser, I assume.

Will work on it!

#18 - 03/04/2020 03:53 AM - CyrilBrulebois
Except direct wget/curl don't get any DNS resolution; stealing the "let's set all HTTP/HTTPS proxy variables" dance from 10-tbb doesn't work either since HTTPS connections are denied. See Tor Browser where we need http:// for the same reason…

@intrigeri:
Do we have a somewhat generic HTTP-not-redirecting-to-HTTPS service we could use for such tiny hacks, instead of relying on individual developer's machines? (ab)using torbrowser-archive seems wrong, but maybe maintaining a duplicate service for very infrequent hacks isn't much better because of the extra sysadmin time?

(BTW, cachewebsite is wow!)

#19 - 03/05/2020 07:50 AM - intrigeri
- Related to deleted (Feature #17504: rtl8812bu)

#20 - 03/05/2020 07:59 AM - intrigeri
Hi @CyrilBrulebois,

To make it explicit: my interest in this particular topic was prompted by the fact it indeed seemed like a regression introduced by the "hot fix" for Mac machines, and it seemed to affect several chipsets.

FWIW (not that it matters much anymore), I doubt that's the case: this very issue was created a week before that hot fix was released.

In our January report, help desk said that the issues possibly introduced in 4.1.1 were: #17388, #17430, #17418.

intrigeri wrote:

Implementation-wise, if possible I'd rather not add binary files to Git: it would bloat the Git repo forever and might have annoying licensing implications. Maybe we could use a build hook to wget/curl them from upstream + verify them against a hard-coded checksum? (That's essentially what we do for Tor Browser; that's what a couple packages we install do under the hood, e.g. libdvdcss & firmware-b43*, so it would not be a first.)
ACK. I was definitely fishing for ideas, and adding binary stuff to the git repository made me twitch when I wrote it, but I thought you'd get the general idea. :)

Except direct wget/curl don't get any DNS resolution; stealing the "let's set all HTTP/HTTPS proxy variables" dance from 10-tbb doesn't work either since HTTPS connections are denied. See Tor Browser where we need http:// for the same reason…

Ah, right!

Do we have a somewhat generic HTTP-not-redirecting-to-HTTPS service we could use for such tiny hacks, instead of relying on individual developer's machines? (ab)using torbrowser-archive seems wrong, but maybe maintaining a duplicate service for very infrequent hacks isn't much better because of the extra sysadmin time?

We also have {.tagged.snapshots,time-based.snapshots}.deb.tails.boum.org that don't redirect to HTTPS for the same reason. If we end up (ab)using one of these things, I slightly prefer torbrowser-archive because there's some minimal versioning thanks to git-annex.

Other options I can think of:

- bypass apt-cacher-ng
  - pro: hopefully very cheap to implement, which is good for a temporary hack
  - drawback: breaks the offline build option; I'm not aware of anyone using it these days and I have no idea if it still works, so IMO it would be acceptable to do that as a temporary hack, and see if anyone complains
- add linux-firmware.git as a Git submodule
  - pro: consistency, as it's one of our most common ways to track & pull third-party code/files during the build
  - cons: I dunno, maybe it's huge?

These 3 options seem acceptable to me.
scope of this particular issue, but it would be great if I, and we, had a generic way to implement them without entering the "nasty territory". (For what it's worth, I'm not tracking the extra time spent on this topic, I've capped this issue at 2 hours.)

intrigeri wrote:

FWIW (not that it matters much anymore), I doubt that's the case: this very issue was created a week before that hot fix was released. In our January report, help desk said that the issues possibly introduced in 4.1.1 were: #17388, #17430, #17418.

It's entirely possible I mixed things up a little, and the vague feeling I had was "woops, maybe the hotfix broke many chipsets, let's look at that one I'm seeing on our plate".

We also have [:tagged.snapshots.time-based.snapshots].deb.tails.boum.org that don't redirect to HTTPS for the same reason. If we end up (ab)using one of these things, I slightly prefer torbrowser-archive because there's some minimal versioning thanks to git-annex.

Yeah, I thought about the deb.tails.boum.org one but that didn't seem a good fit... I thought about torbrowser-archive but given it only had version numbers directly under the root directory, adding some workarounds/ top-level directory seemed a little sad.

Other options I can think of:

- bypass apt-cacher-ng
  - pro: hopefully very cheap to implement, which is good for a temporary hack
  - drawback: breaks the offline build option; I'm not aware of anyone using it these days and I have no idea if it still works, so IMO it would be acceptable to do that as a temporary hack, and see if anyone complains

I'm skipping this proposal because I'm tempted to build on the second one.

- add linux-firmware.git as a Git submodule
  - pro: consistency, as it's one of our most common ways to track & pull third-party code/files during the build
  - cons: I dunno, maybe it's huge?

These 3 options seem acceptable to me.

It seems a bit heavy for this specific case, seeing 350+ MB worth of .git.

But I'm wondering whether having some tails-workarounds git submodule would be suitable for the random workarounds we might need over time? This would mean being able to drop anything at once without breaking the history in the main repository, if there were any issues (e.g. mainly we don't like tracking binary contents of the main repo, or maybe more annoyingly redistributability disputes, or whatever). That could also be used to beta-test changes involving updated or new Debian packages that we wouldn't need to upload to deb.tails.boum.org, and that we wouldn't have to self-host (with all the keyring fun that involves), the latter being what I ended up doing when testing the feasibility of 4.1.1. And that wouldn't involve more sysadmin-side setup (e.g. no need to duplicate what we do for torbrowser-archive).

I can propose that plan on tails-foundations@ / tails-dev@ to avoid having you spend more time on this/burden you with making a call.

.oO( thinking about it a little more, not sure whether I have the appropriate rights to create a new repository at the moment anyway... I've never needed to do that yet... )
Hi,

But I'm wondering whether having some tails-workarounds git submodule would be suitable for the random workarounds we might need over time? This would mean being able to drop anything at once without breaking the history in the main repository, if there were any issues (e.g. mainly we don't like tracking binary contents of the main repo, or maybe more annoyingly redistributability disputes, or whatever).

Sounds good to me!

That could also be used to beta-test changes involving updated or new Debian packages that we wouldn't need to upload to deb.tails.boum.org, and that we wouldn't have to self-host (with all the keyring fun that involves), the latter being what I ended up doing when testing the feasibility of 4.1.1.

I don't understand this part. More specifically, I don't understand what is the limitation of our custom APT repo (deb.t.b.o) that makes workarounds desirable or even needed. Could you please clarify?

.oO( thinking about it a little more, not sure whether I have the appropriate rights to create a new repository at the moment anyway… I've never needed to do that yet… )

Indeed, currently only sysadmins can create new repos: [https://tails.boum.org/contribute/git/#creating-a-new-repository](https://tails.boum.org/contribute/git/#creating-a-new-repository)
I've created this to unblock your progress:

git clone tails@git.tails.boum.org:tails-workarounds

---

#23 - 03/07/2020 05:38 PM - CyrilBrulebois
- Status changed from Confirmed to In Progress

Applied in changeset tails@87756515f1fa51bc30fbb64aa93b7d67b343c384.

#24 - 03/07/2020 10:55 PM - CyrilBrulebois
I'd be happy to get a review on the bugfix/17323-fix-missing-rtw88-firmware branch.

Maybe @intrigeri or @segfault would have a chance before 4.4?

I'll need to concentrate on merging Tor Browser soon anyway, so if for some reason we miss the window for 4.4, this can likely wait until 4.5 anyway.

#25 - 03/07/2020 10:56 PM - CyrilBrulebois
- Assignee deleted (CyrilBrulebois)

#26 - 03/08/2020 06:41 AM - intrigeri
- Status changed from In Progress to Needs Validation
- Feature Branch set to bugfix/17323-fix-missing-rtw88-firmware

#27 - 03/08/2020 06:43 AM - intrigeri
- Assignee set to intrigeri

I'd be happy to get a review on the bugfix/17323-fix-missing-rtw88-firmware branch.

ACK. I've updated metadata accordingly.

Maybe @intrigeri or @segfault would have a chance before 4.4?

I'll give it a shot this morning. Ideally I'd like 1+ affected user to confirm it fixes the problem, but given the timing I'm not going to block on that.

I'll need to concentrate on merging Tor Browser soon anyway, so if for some reason we miss the window for 4.4, this can likely wait until 4.5 anyway.

Agreed.

#28 - 03/08/2020 07:49 AM - intrigeri
Hi!
@goupille,

Ideally I'd like 1+ affected user to confirm it fixes the problem, but given the timing I'm not going to block on that.

I've asked one such user (wb://935ae832daa20a6ea1f44bdfa5b8c00d) to try this branch.

goupille wrote:

there were 3 reports during the first week after tails 4.1 and probably more since then

I've found one single report (935ae832daa20a6ea1f44bdfa5b8c00d) that got classified as such (at least, in a way that's visible to me). Could you please point me to these other reports? Unless perhaps 935ae832daa20a6ea1f44bdfa5b8c00d was the only one with contact info, in which case: forget it!

@CyrilBrulebois,

I'd be happy to get a review on the bugfix/17323-fix-missing-rtw88-firmware branch.

I'll give it a shot this morning.

Code review passes. Great job! I'm now building an image to verify that the firmware indeed lands where I would expect.

Just one comment: I'd rather see "WARNING: Firmware for $file found, maybe this hook could be dropped" be a fatal build error, to avoid shipping outdated firmware once Debian includes and they get updated there. I'm concerned that this warning will be hard to spot in the middle of a 15k-lines build log, that has 30+ other warnings which are all false positives. If you agree, I can make this trivial change myself before merging.

In passing, I've verified that we don't need rtw8822c_wow_fw.bin: it's only needed to resume a suspended system using Wake-on-LAN over Wi-Fi, which I don't expect Tails users will ever try :)
#29 - 03/08/2020 04:09 PM - intrigeri
- Status changed from Needs Validation to In Progress

Applied in changeset tails|9c8eba6a847d9d22af847b63a76d32e2759bc0ed.

#30 - 03/08/2020 04:11 PM - intrigeri
- Target version changed from Tails_4.4 to Tails_4.5

Merged after turning the warning into a fatal build error, as agreed on XMPP with kibi.
I'm not closing this yet, because at this point we don't know if the missing firmware was the only problem. I'll come back to it once I get an answer from affected user or from help desk!

#31 - 03/09/2020 02:39 PM - goupille
intrigeri wrote:

Ideally I'd like 1+ affected user to confirm it fixes the problem, but given the timing I'm not going to block on that.

I've asked one such user (wb://935ae832daa20a6ea1f44bdfa5b8c00d) to try this branch.

goupille wrote:

there were 3 reports during the first week after tails 4.1 and probably more since then

I've found one single report (935ae832daa20a6ea1f44bdfa5b8c00d) that got classified as such (at least, in a way that's visible to me). Could you please point me to these other reports? Unless perhaps 935ae832daa20a6ea1f44bdfa5b8c00d was the only one with contact info, in which case: forget it!

they weren't wp reports, I just resent them your email to the other user (i've cc'ed you so you should have their contact as well now)

#32 - 03/09/2020 02:51 PM - intrigeri

Hi!
I just resent them your email to the other user

Thank you. I see 2 such messages presumably from you today.

(i've cc'ed you so you should have their contact as well now)

That part did not work: I see the Cc header to me, but no To header. I suspect you used x-resend-cc, which may not work in the way you would expect (at least, it does not work the way I thought): https://0xacab.org/schleuder/schleuder/issues/345

one of them was encrypted for all the recipients, I assumed at least this one should have worked...
well, I'm not sure what is the difference between x-resend and x-resend-cc, then, I give it a try every now and then and still end up resending everything manually...
any way, you should have received those emails again with the users contact :)

I've emailed the two users for which I have contact info, asking them to test with Tails 4.4.

I've emailed the two users for which I have contact info, asking them to test with Tails 4.4.

One of these 2 messages was rejected due to an over-quota mailbox.
#37 - 03/18/2020 08:56 AM - intrigeri

- Status changed from In Progress to Resolved
- Assignee deleted (intrigeri)
- Target version changed from Tails_4.5 to Tails_4.4