Replicant - Issue #1780

Update the webview apk

03/15/2017 04:44 PM - Wolfgang Wiedmeyer

Status:	New	**Start date:**	03/15/2017
Priority:	Urgent	**Due date:**	
Assignee:	Andrés D	**% Done:**	0%
Category:	Security	**Estimated time:**	0.00 hour
Target version:	Replicant 6.0 0005	**Spent time:**	0.00 hour
Resolution:		**Grant:**	
Device:			

Description

Due to #705, the webview apk in Replicant 6.0 cannot be updated. Currently, webview version 43.0.2357.134 is in use. It was released in July 2015 and has numerous security issues that were discovered since then.

Updating the webview apk would fix a lot of security issues and would ensure that websites can be visited securely using the browser shipped with Replicant or Lightning.

Related issues:

Related to Replicant - Issue #1786: Review the Chromium Webview build environ...	**Feedback**	**04/09/2017**
Related to Replicant - Issue #1947: Ask upstream F-Droid to build up to date ...	**New**	**08/22/2019**
Blocked by Replicant - Feature #1844: Select libagl/llvmpipe per app	**Resolved**	**11/02/2017**
Blocks Replicant - Issue #1949: Make web browser and email client support TLS...	**In Progress**	**08/23/2019**

History

#1 - 03/15/2017 11:34 PM - Wolfgang Wiedmeyer

- Assignee changed from Paul Kocialkowski to Wolfgang Wiedmeyer

#2 - 03/29/2017 03:05 AM - Jeremy Rand

FWIW, I've been happily using llvmpipe for about 2 weeks based on Wolfgang's instructions for Replicant 6.0. It's definitely less snappy, but I'm okay with the extra lag in return for the improved security of using Orfox. Would it be feasible to release an alternate Replicant 6.0 build with a current WebView, for the users like me who are okay with llvmpipe's current state?

#3 - 03/29/2017 03:44 AM - Kurtis Hanna

It is my understanding that newer versions of webview currently can't be used because they don't work with the software rendering.

#4 - 03/29/2017 01:16 PM - Wolfgang Wiedmeyer

> FWIW, I've been happily using llvmpipe for about 2 weeks based on Wolfgang's instructions for Replicant 6.0. It's definitely less snappy, but I'm okay with the extra lag in return for the improved security of using Orfox.

This is great to hear that llvmpipe in its current state is already usable for you!

Newer versions of the webview do indeed work with llvmpipe, at least they should. Latest versions may still introduce issues but these seem to get fixed by the Mesa or Android-x86 developers over time.

Please note that the apk is not built as part of a regular Replicant build. The apk needs to be built separately in a chromium build environment and only the final apk is committed to the source code. The apk can be installed with adb install -r webview.apk, just like a normal app. So there is no need for a completely separate Replicant build, just because of one apk file.

I didn't look into the loader code for the webview, so I don't know if it's possible to switch between the two in a similar way like with llvmpipe and the Android software renderer. At least it should be possible to additionally ship an updated webview apk as part of a Replicant 6.0 zip. Then it can be manually switched between the two with something like adb shell mv old-webview.apk old-webview.apk.bak && adb shell mv new-webview.apk old-webview.apk

#5 - 08/26/2018 08:59 PM - Kurtis Hanna

- Target version changed from Replicant 6.0 to Replicant 6.0 0004

#6 - 08/26/2018 09:26 PM - Kurtis Hanna

It would be great if we could utilize the script that selects libagl/llvmpipe on a per app basis https://redmine.replicant.us/issues/1844 and somehow configured llvmpipe by default on every app using webview. I don't know how this could be automated.

Here's a suggestions related to how one can visually inspect an app to see if it uses WebView: https://stackoverflow.com/a/19160572

#7 - 08/26/2018 09:38 PM - Kurtis Hanna

Is this the source code for the most current stable version of Webview? https://chromium.googlesource.com/chromium/src/+/master/android_webview/ If so, like Wolfgang said, the Webview apk needs to be built separately in a chromium build environment, so it'd be good if we could build it ourselves.

#8 - 08/26/2018 09:41 PM - Kurtis Hanna

These instructions reference Android 5.0.0 so it might be old https://www.chromium.org/developers/how-tos/build-instructions-android-webview

#9 - 08/30/2018 04:54 PM - Kurtis Hanna

- Target version changed from Replicant 6.0 0004 to Replicant 6.0 0005

#10 - 08/24/2019 08:28 PM - Kurtis Hanna

The old version of WebView that we use doesn't have TLS 1.2 GCM ciphersuites: https://redmine.replicant.us/issues/1949

This is yet another reason why we need to update WebView.

#11 - 08/24/2019 08:30 PM - Kurtis Hanna

- Blocked by Feature #1844: Select libagl/llvmpipe per app added

#12 - 08/27/2019 03:35 PM - Andrés D

- Assignee changed from Wolfgang Wiedmeyer to Andrés D

- Target version changed from Replicant 6.0 0005 to Replicant 6.0 0004

#13 - 10/20/2019 11:22 PM - Kurtis Hanna

- Blocks Issue #1960: Build release candidate image for 6.0 0004 added

#14 - 10/22/2019 02:52 AM - Kurtis Hanna

Debian, Guix, Bromite, Iridium, Inox, ungoogled-chromium, Iridium, and Vanadium all seem to have their own versions of WebView.

#15 - 10/22/2019 02:55 AM - Kurtis Hanna

- Related to Issue #1947: Ask upstream F-Droid to build up to date WebView added

#16 - 10/23/2019 02:43 AM - dl lud

I took a look at all the Chromium forks mentioned above, except for Vanadium, which I couldn't find.

After looking at the manifestos and design principles for all of them, ungoogle-chromium seemed to be more aligned with software freedom.

Then I found out that Guix (a FSDG compliant distro) distributes ungoogled-chromium after running it through a build recipe that removes a few extra files.

On Guix mailing list there is the following bold statement:

> To the best of our knowledge, ungoogled-chromium as packaged in guix is completely free

I would say this is our best bet and we should try to build WebView after Guix build recipe for ungoogled-chromium.

As for the other forks, here's what I learned about them:

- Bromite - Quite interesting for the fact that the codebase is used to build WebView. Focused on privacy and ad blocking, not on software freedom.
- Debian - Limited patch set that strives to use system libs instead of binaries but does not go as deep as ungoogled-chromium when it comes to removing Google services.
- Iridium - Tries a step on every direction. Isn't as thorough as ungoogled-chromium about ungoogling and doesn't seem to replace built-in binaries for system libs.

Edit: the proprietary codecs that Bromite brings are stuff like h264, mp3 or mp4. Which may be under some patent in some crazy legislation, but there is a free software implementation for it (which is actually distributed on FSDG distros like Parabola). Thus no issue there.

#17 - 10/23/2019 04:23 AM - dl lud

There is an ungoogled-chromium fork focused on bringing it to Android: ungoogled-chromium-android

The main dev reports success on building the WebView apk with it.

A tantalizing approach for Replicant seems to be:

1. grab ungoogled-chromium-android
2. pass it through the Guix "filters" for ungoogled-chromium
3. make it output a WebView apk.

There are some foreseeable hiccups along the way with this approach:

1. ungoogled-chromium-android targets API 24. Replicant 6 is API 23.
2. How should we re-create what Guix did on top of ungoogled-chromium?

#18 - 10/24/2019 02:11 AM - dl lud

Unobtainium is a project that, besides removing Google services and libraries from Chromium, also tried to get rid of all prebuilts. The goal was to be built from within F-Droid. Unfortunately the project has been dormant for an year now, while Chromium advanced full speed ahead. We are looking to know whether some Unobtainium patches could still be applied on top of ungoogled-chromium-android.

#19 - 10/25/2019 12:18 AM - dl lud

My proposal of approach, for the good soul that can take this task, is:

1. Start off with ungoogled-chromium-android.
2. Try to make it work with API 23.
3. Remove the remaining prebuilts while taking inspiration from Unobtainium.
4. Check if any remaining Unobtainium patches still apply on top of ungoogled-chromium-android.
5. Check if the third-party files filtered out by Guix are still in ungoogled-chromium-android at this stage.
6. Try to build everything from fdroid-server like Unobtainium does. It's a great way to pick leftover prebuilts.
7. Gather all the devs involved, write a paper, make a few slides, submit to a relevant free-software conference. From there shout out loud to the entire world how fucked up Chromium is.

Both ungoogled-chromium-android and Unobtainium devs are willing to help. Coordinate with them.

#20 - 10/26/2019 06:33 PM - dl lud

The Parabola team does not deem Guix approach to be acceptable because it hasn't been presented to the FSDG work-group for evaluation. The current recommendation for Chromium is still to: Remove program/package

Note that the above is a recommendation. Replicant may decide otherwise, but in that event we would have to prove why our approach works and we should share it with all other FSDG distros.

#21 - 10/26/2019 07:14 PM - dl lud

Addition: Guix developers did present something, their guile code for the package. I'm trying to understand why it is not enough.

Guix developers stand by their approach, and reply that all freedom related issues pointed onto their version of ungoogled-chromium have been solved.

#22 - 10/27/2019 01:27 AM - dl lud

After a long review of most email threads on GNU-linux-libre that address the Chromium subject, with special focus on the latest thread that spun up when Guix added their build of ungoogled-chromium. I, Andrés D and Kurtis Hanna concluded that Guix's approach is reasonable. Guix devs actually posted a plain English description for it with all subsequent rebuttals being suspicions and not actual findings of non-free content.

Here's our revised proposal to approach this task:

1. Start off with Guix's source code for ungoogled-chromium, i.e. after being cleaned by their build script.
2. Run Ubuntu license check script on top of it.
3. Check if any "BlockedOn" issue from the original Chromium bug still applies (hint: most of them should be related to third-party code that was removed).
4. Try to build WebView out of it (will probably fail).
5. Cherry pick all the necessary patches from ungoogled-chromium-android and Unobtainium.
6. Make it work with API 23.
7. Try to build everything from fdroid-server like Unobtainium does. It's a great way to pick leftover prebuilts.
8. Send recipe to be peer-reviewed at GNU-linux-libre, written in plain English, and explaining how it addresses Luke's concerns.

#23 - 10/30/2019 03:24 AM - Kurtis Hanna

My understanding of the discussion related to this on IRC was that, for our 6.0 0004 release, we will build and provide an up to date version of webview using ungoogled-chromium's tree after running it through a Guix build recipe that removes a few extra files.

I think that once webview is built and the apk is integrated into our build system we will be ready to release 6.0 0004.

While not directly related to this ticket, I think it was decided in Replicant 9 webview will be removed until we can find a way to use gecko instead of chromium as the underlying engine, but still keep feature parity with the webview API.

Please correct me if this is wrong.

#24 - 10/30/2019 03:48 AM - dl lud

Hi Kurtis,

I think you are wrong.

There is no one available to complete the task of providing an up-to-date WebView in due time for the Replicant 6 0004 release. Therefore the approach agreed upon on IRC was:

1. Release Replicant 6 0004 with the old WebView, alongside with a warning stating it's security holes. This warning should advise people to use a Gecko based browser such as Fennec F-Droid, which although slow, will now work on 0004 due to the usage of llvmpipe software renderer. The old WebView should only be used on apps that render trusted content.
2. Replicant 9 will be released without WebView only if there is neither: a) a liberated WebView or b) a shim that envelops GeckoView and makes it API-compatible with WebView (the Upstream page has more details on this).

IMHO the GeckoView shim, although much better, is not accomplishable in due time even for Replicant 9 (unless some new contributor appears, or someone from Mozilla steps in). We must thus strive to complete the WebView/Chromium liberation approach suggested above in time for the Replicant 9 release. It seems to be our best hope.

#25 - 10/30/2019 03:56 AM - dl lud

Another possible approach for Replicant 9, would be releasing it without WebView, and fork the most important apps that depend on WebView to use GeckoView instead. This approach would be almost madness as too many apps depend on WebView. It would be impossible for the small Replicant team to maintain this. It would only work if the app maintainers themselves perceive GeckoView as a better alternative and start using it upstream.

#26 - 12/11/2019 11:00 PM - Denis 'GNUtoo' Carikli

- Target version changed from Replicant 6.0 0004 to Replicant 6.0 0005

Since we'll have llvmpipe work by default and that it would delay a lot the release we can move that to be done later if we choose to do it for Replicant 6.

#27 - 01/04/2020 07:47 PM - Kurtis Hanna

The quick and dirty approach to update webview in Replicant 6 is to:

1) grab ungoogled-chromium-android
2) make it compatible with API 23
3) compile

The reason why ungoogled-chromium-android is not compatible with API 23 was described by the author here:
https://github.com/wchen342/ungoogled-chromium-android/issues/7#issuecomment-545561123

#28 - 01/04/2020 07:50 PM - Kurtis Hanna

- Blocks Issue #1949: Make web browser and email client support TLS 1.2 added

#29 - 01/06/2020 09:52 PM - Kurtis Hanna

- Blocks deleted (Issue #1960: Build release candidate image for 6.0 0004)