Haketilo - Feature #90

Make the 0.1 release

09/06/2021 02:39 PM - koszko

Status:	Closed	**Start date:**	09/06/2021
Priority:	High	**Due date:**	
Assignee:		**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

Right now what we have left to do is:

1. ~~Make it impossible to check "allow" option for page with payload, as suggested in~~ #15
2. ~~Try to implement the Ajax approach from~~ #13 ~~and make build.sh generate a random file for it~~
3. ~~Re-enable the data: URLs blocking facility (now without the intrinsics blocking)~~
4. ~~Document basic Hachette usage~~
5. ~~Bump version in manifest.json to 0.1~~
6. Spread the word

I figured out it's going to be simpler to list it all in a single issue. 1-4 can be completed in any order. I'll start with 2. and (unless you really want to do it) 1.

Additional issues that are important but were not listed at first:

- ~~Drop all CSP for pages with payload, leave CSP intact for all other ones~~
- ~~Limit the processed length of path and domain parts of a URL (regards both Hachette and Hydrilla)~~
- ~~Test compatibility with LibreJS~~
- ~~(Yet to be decided) change extension's name (to "Haketilo"?) as per recent request~~
- ~~Make a srevice that generates Haketilo Chromium builds with different secret~~
- Provide a signed .xpi (not urgent)

History

#1 - 09/06/2021 04:54 PM - koszko

- Description updated

Instead of implementing 2. as specified in the description, I did something else. Effect is as wanted - build.sh generates a Chromium manifest with random value. Details in #13

#2 - 09/06/2021 08:51 PM - koszko

- % Done changed from 0 to 20

- Description updated

3. is now ready, as noted in #78

#3 - 09/07/2021 12:05 AM - jahoti

That leaves me with 4, I suppose, which is probably just as well; the current (limited) state of the testing suite is also in need of documentation.

(Come to think of it, testing compatibility with LibreJS might also be useful for 0.1; I'll do that first.)

#4 - 09/10/2021 05:15 PM - koszko

- % Done changed from 20 to 60

- Description updated

"allow" option, CSP behavior and URL length limits are now on koszko branch

#5 - 09/10/2021 10:07 PM - koszko

I started documenting Hachette usage. I uploaded the screenshots I made, so if you happen to come there while I sleep, you can easily continue where I left off :)

EDIT: It's (of course) on User manual and its subpages. Also, at some point we'll upload prebuilt versions of Hachette here. For now we can leave some empty placeholders in the doc for where file links will be...

#6 - 09/11/2021 05:03 AM - jahoti

> Also, at some point we'll upload prebuilt versions of Hachette here.

On that note (and your breakthrough with CRX on #13), do we want to sign releases? I'm assuming we can't for Chromium-based browsers regardless; with Mozilla, by constrast, I can get signatures with only free software (assuming they haven't broken anything recently) and with some more reverse engineering it would even be possible to submit to the add-ons store.

I'm working on the user documentation now, in any case.

#7 - 09/11/2021 11:38 AM - koszko

jahoti wrote:

> On that note (and your breakthrough with CRX on #13), do we want to sign releases?

Yes. And I'd like to use a separate key for that. Firstly, because it is going to be more vulnerable as it will be accessible to the service that generates signed .crx files, secondly, because at some point we might want to release or share it (consider how handling of signed apps in Android stops F-Droid from building programs by itself like a proper software repository should).

So, I see it this way:

- One key will be used to create Chromium .crx releases of the extension and nothing else.
- My personal PGP key will be used to sign:
  - source tarballs
  - packages for distros (.deb and similar)
  - the key above

> I'm assuming we can't for Chromium-based browsers regardless;

What do you mean by that? Merely generating a valid .crx file requires signing it. Installing such .crx file will require users to toggle a browser configuration flag, but for many users this will still be preferable to installing as Unpacked Extension.

> with Mozilla, by constrast, I can get signatures with only free software (assuming they haven't broken anything recently) and with some more reverse engineering it would even be possible to submit to the add-ons store.

Seems great :)

> I'm working on the user documentation now, in any case.

□□

EDIT: changed the signing plans as there seems to be no way to self-sign Mozilla extensions

#8 - 09/11/2021 11:54 AM - jahoti

Wait- is it possible to sign XPIs with our own key? If so that would be much better than relying on Mozilla.

In any case, as I was about to write just minutes ago, it seems we don't have a choice anyway; all attempts at installing an unsigned extension on LibreWolf thus far have failed, suggesting version 90.0.2 at least copied Firefox Stable's behavior of completely banning them :(.

#9 - 09/11/2021 12:22 PM - koszko

jahoti wrote:

> Wait- is it possible to sign XPIs with our own key? If so that would be much better than relying on Mozilla.

I just realized it probably isn't. I earlier blindly assumed it works similarly to Chromium

> In any case, as I was about to write just minutes ago, it seems we don't have a choice anyway; all attempts at installing an unsigned extension on LibreWolf thus far have failed, suggesting version 90.0.2 at least copied Firefox Stable's behavior of completely banning them :(.

Do you mean some older LibreWolf versions allowed that? Anyway, this doesn't match the overall attitude of the LibreWolf project. Perhaps we can contact the devs about that? How about browsers from ethical distros? I once tested installation from .xpi but don't remember on which browser, probably Parabola Iceweasel 75. It worked at least there

#10 - 09/11/2021 12:35 PM - jahoti

> Wait- is it possible to sign XPIs with our own key? If so that would be much better than relying on Mozilla.

> I just realized it probably isn't. I earlier blindly assumed it works similarly to Chromium

That makes sense- I was hopeful, yet unfortunately it seems Mozilla and Google do use very different systems.

> Do you mean some older LibreWolf versions allowed [using unsigned extensions]?

I'm not entirely sure, having not tried except with my one version.

> Anyway, this doesn't match the overall attitude of the LibreWolf project. Perhaps we can contact the devs about that?

That would be a good idea, actually; there are allowances in the source code for that use case, and it's just dawned on me that a comment I read recommending LibreWolf as a way to circumvent that policy is probably only a few weeks old.

Perhaps the problem is that I'm using an AppImage; I will try obtaining something more sensible in the morning and test with that.

> How about browsers from ethical distros? I once tested installation from .xpi but don't remember on which browser, probably Parabola Iceweasel 75. It worked at least there

They should work; IceCat 60 does, and Tor Browser allows it to be enabled in the advanced settings. I did take some screenshots with them, in fact, only to stop when IceCat refused to allow screenshots of the context menu for installation and Tor prominently warned against installation of unverified xpis (which will not appear if we distribute a signed one).

#11 - 09/11/2021 12:51 PM - koszko

- Description updated

Interesting. The flag that enables unverified installs is supposedly still supported in developer edition of Firefox:
https://support.mozilla.org/en-US/questions/1134589

Also, I wonder how Firefox in mainstream GNU/Linux distros behaves. Are they based off the developer version or not? Also, disrtos like Debian actually have extensions in their repositories, so there is surely *some* way to install there.

Although I believe we have the same view about the thing, I thought it'd be good to state it explicitly:
When possible with free software, we are willing to provide a Mozilla-signed XPI as well as distribute in extension stores. However, we also want to direct users towards browsers and installation methods that don't make them reliant on Mozilla or Google. Installation from extension stores will be the unrecommended approach (unless an ethical store comes up, of course)

EDIT: As to LibreWolf, it seems to have been a bug caused by Mozilla changing the handling of a flag responsible for signatures enforcement. Everything should be working again in newer LibreWolf builds:
https://gitlab.com/librewolf-community/browser/arch/-/issues/49

EDIT2: Btw, by mentioning you can get signatures with only free software, you mean you have a way to create an actual Firefox account, right? Could you share it?

#12 - 09/12/2021 12:41 AM - jahoti

> Interesting. The flag that enables unverified installs is supposedly still supported in developer edition of Firefox:
> https://support.mozilla.org/en-US/questions/1134589

It is indeed.

Actually, that reminds me- I have a copy of the developer edition somewhere here! I'll add it to the list of browsers and do some testing with it soon.

> Also, I wonder how Firefox in mainstream GNU/Linux distros behaves. Are they based off the developer version or not? Also, disrtos like Debian actually have extensions in their repositories, so there is surely some way to install there.

In a trivial sense they are based on the developer version; the source code of all the Firefox builds is the same, it's just the build parameters that get altered. I would strongly suspect they do support unsigned extensions given that (according to the original Bugzilla issue I found yesterday) distro repositories were the specific use case being considered, yet that would require specific checking (it's also possible they just distribute the signed extensions and package the signatures when building from source).

> Although I believe we have the same view about the thing, I thought it'd be good to state it explicitly:
>
> > When possible with free software, we are willing to provide a Mozilla-signed XPI as well as distribute in extension stores. However, we also want to direct users towards browsers and installation methods that don't make them reliant on Mozilla or Google. Installation from extension stores will be the unrecommended approach (unless an ethical store comes up, of course)

Indeed, we can both agree with that! If there were a way to distribute a Google-signed CRX that could be installed without enabling developer mode that too would be good to include; it sounds like that isn't possible, however.

> EDIT: As to LibreWolf, it seems to have been a bug caused by Mozilla changing the handling of a flag responsible for signatures enforcement. Everything should be working again in newer LibreWolf builds:
> https://gitlab.com/librewolf-community/browser/arch/-/issues/49

Thanks for tracking that down! I'll try with a newer build, then, and go from there.

> EDIT2: Btw, by mentioning you can get signatures with only free software, you mean you have a way to create an actual Firefox account, right? Could you share it?

Yes, and yes! Just let me test that it still works and make it somewhat usable and I'll upload it here.

#13 - 09/12/2021 11:13 AM - jahoti

I'm working through testing the Mozilla account-generation script now. I've removed the signing functionality rather than clean it up, as it's particularly terrible and can already be done with Mozilla's web-ext; however, if you'd rather not go near NodeJS and NPM, I can clean up that as well.

As for LibreWolf, while I didn't try too hard, it unfortunately seems the AppImage for version 92 remains broken. I'll try a few more things and if that doesn't work, perhaps contacting the devs will be needed...

#14 - 09/13/2021 08:02 AM - koszko

> Also, disrtos like Debian actually have extensions in their repositories, so there is surely some way to install there.

> (it's also possible they just distribute the signed extensions and package the signatures when building from source).

While technically possible, it would probably go against Debian policy (or at least *unwritten* policy). When I think about it, a distro that did it this way would need to be a very dirty one...

> Indeed, we can both agree with that! If there were a way to distribute a Google-signed CRX that could be installed without enabling developer mode that too would be good to include; it sounds like that isn't possible, however.

However, it might still be useful to distribute such CRX. I recall that installation of user-signed add-ons is disabled in some (Losedows?) official builds of Chrome/Chromium, so Google-signed CRX + developer mode could be a viable solution there.

> I'm working through testing the Mozilla account-generation script now. I've removed the signing functionality rather than clean it up, as it's particularly terrible and can already be done with Mozilla's web-ext; however, if you'd rather not go near NodeJS and NPM, I can clean up that as well.

Although I'd rather not go near NPM, it seems API is well documented. I should be able to work with it using curl or something.

> As for LibreWolf, while I didn't try too hard, it unfortunately seems the AppImage for version 92 remains broken. I'll try a few more things and if that doesn't work, perhaps contacting the devs will be needed...

Right. That issue was Arch-specific. Perhaps it only got fixed there and not for other kinds of builds? While it is surely possible to set up a chroot in which a LibreWolf .pkg.tar.xz could be installed, it's a shame we won't be having an option to recommend to casual users...

#15 - 09/13/2021 09:01 AM - jahoti

> (it's also possible they just distribute the signed extensions and package the signatures when building from source).
> While technically possible, it would probably go against Debian policy (or at least unwritten policy). When I think about it, a distro that did it this way would need to be a very dirty one...

True! It is very unlikely; on the other hand, Debian does distribute Firefox trademarks using an exception specifically for them that doesn't solve the issue of software freedom (as far as I know).

> Indeed, we can both agree with that! If there were a way to distribute a Google-signed CRX that could be installed without enabling developer mode that too would be good to include; it sounds like that isn't possible, however.

> However, it might still be useful to distribute such CRX. I recall that installation of user-signed add-ons is disabled in some (Losedows?) official builds of Chrome/Chromium, so Google-signed CRX + developer mode could be a viable solution there.

If we can then we definitely should create and distribute such a CRX, perhaps even through the official store if that can be made to work ethically for us without too much hassle. However, don't they require developers to identify themselves, create a Google account, and pay a fee?

> I'm working through testing the Mozilla account-generation script now. I've removed the signing functionality rather than clean it up, as it's particularly terrible and can already be done with Mozilla's web-ext; however, if you'd rather not go near NodeJS and NPM, I can clean up that as well.

> Although I'd rather not go near NPM, it seems API is well documented. I should be able to work with it using curl or something.

Very probably! The only potential complication I can see is the authorization part, which requires generating JSON Web Tokens. They might be much more common that I thought, in which case there's no issue at all; if not, however, the script I'll upload in the next submission has a function jwt_header that generates a suitable value for the HTTP Authorization header using your key (jwtkey) and secret (jwtsec).

> As for LibreWolf, while I didn't try too hard, it unfortunately seems the AppImage for version 92 remains broken. I'll try a few more things and if that doesn't work, perhaps contacting the devs will be needed...

> Right. That issue was Arch-specific. Perhaps it only got fixed there and not for other kinds of builds?

It was too! That would go some way to explaining it, especially given none of the other repositories have a corresponding issue (except maybe Gentoo's).

> While it is surely possible to set up a chroot in which a LibreWolf .pkg.tar.xz could be installed, it's a shame we won't be having an option to recommend to casual users...

It really is. I will try and contact the LibreWolf devs in the hope that they can either confirm this was fixed or go and fix it.

#16 - 09/13/2021 09:12 AM - jahoti

- File mozoid.py added

OK, the Firefox account generation/management script is attached. Some notes:

- It depends on librecaptcha, requests, and cryptography.
- There is no actual password involved as such- instead, the script takes a 64-character hexadecimal string (no '0x' at the front, of course). While the actual process generates this from a user password by "stretching" it (I think it's fairly easy to find the exact process by searching "mozilla stretched password" or similar), finding a suitable library and its usage seemed too hard when this was an option ;).
- The error handling is still terrible; hopefully it's at least informative enough to let you know what to do, but feel free to ask otherwise.
- Creating an account can be done with mozoid.py -c; for help, mozoid.py -h should work.

Ultimately, this will hopefully end up as a fix for Haketilo and be much less painful for it!

(except on some Mozilla browsers where several mozilla domains are privileged pages)

#17 - 09/13/2021 09:46 AM - koszko

jahoti wrote:

> OK, the Firefox account generation/management script is attached.

Thanks a lot!

> (except on some Mozilla browsers where several mozilla domains are privileged pages)

Right, in the documentation (at the end of Mozilla installation instructions, perhaps also in some other place(s)) we should advice the users of non-harderned Mozilla browsers to empty the extensions.webextensions.restrictedDomains property in about:config

EDIT: Btw, from the email I conclude you've started the rebranding of Hachette. If so, I shall focus on renaming the Redmine project (if possible) and configuring Apache to use another domain for it (with redirect from the old one).
EDIT2: I see you already renamed the project. Good it was so easily doable. I'll focus on Apache then :)

#18 - 09/14/2021 03:25 AM - jahoti

> Right, in the documentation (at the end of Mozilla installation instructions, perhaps also in some other place(s)) we should advice the users of non-harderned Mozilla browsers to empty the extensions.webextensions.restrictedDomains property in about:config

In the documentation at least this done, yet it probably does indeed belong in multiple locations.

> I see you already renamed the project. Good it was so easily doable.

Indeed- I discovered some time ago I had that power, and was slightly terrified of what I could do until a good use for it came along ;).

> I'll focus on Apache then :)

Thank you for doing that (and fixing the link from the Hydrilla project description)- hopefully now enough traces are cleared to make everyone happy.

As a rather unimportant aside, however, we have yet to establish a clear difference between "Haketilo" and "Haketilo □□". Is it simply a matter of stylistic preference, or is there a distinction in meaning that it might be helpful to give to these?

#19 - 09/14/2021 03:27 AM - jahoti

- % Done changed from 60 to 70

- Description updated

#20 - 09/14/2021 04:24 PM - koszko

> As a rather unimportant aside, however, we have yet to establish a clear difference between "Haketilo" and "Haketilo □□". Is it simply a matter of stylistic preference, or is there a distinction in meaning that it might be helpful to give to these?

I meant no difference in meaning for these. Actually, if we made them mean something different, it would be misleading to ppl

#21 - 09/14/2021 07:20 PM - koszko

- Description updated

In case anyone's wondering how I automatized the generation of Chromium builds with different secres, it's this script (CC0-licensed):

```
#!/bin/bash

echo "Content-type: application/zip"
#echo "Content-type: text/plain"
echo "" # End of headers

HAKETILO_VERSION="$(echo "$0" | sed 's/^.*haketilo-\(.*\)\.sh$/\1/')"
HAKETILO_ZIP="$(echo "$0" | sed 's/\.sh$/.zip/')"
WORKING_DIR=/tmp/haketilo_cgi/$(($(date +%k) % 2))
EXTENSION_DIR="haketilo-chromium-$HAKETILO_VERSION"

#echo version $HAKETILO_VERSION
#echo dir $WORKING_DIR
#echo zip "$HAKETILO_ZIP"
#echo ext dir "$EXTENSION_DIR"

mkdir -p $WORKING_DIR
TMPDIR="$(mktemp -d $WORKING_DIR/XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX)"
#echo tmpdir "$TMPDIR"
#echo pwd "$PWD"

trap 'cd /; rm -rf "$TMPDIR"' EXIT

cp "$HAKETILO_ZIP" "$TMPDIR"
cd "$TMPDIR"
WORKING_ZIP="$(echo *)"
MANIFEST="$EXTENSION_DIR/manifest.json"
unzip "$WORKING_ZIP" "$MANIFEST" > /dev/null

#echo manifest "$MANIFEST"
OLD_DUMMY_NAME="$(grep -oE 'chromium-key-dummy-file-[^"]*+' "$MANIFEST")"
#echo dummy file "$OLD_DUMMY_NAME"
zip -d "$WORKING_ZIP" "$EXTENSION_DIR/$OLD_DUMMY_NAME" > /dev/null

NEW_DUMMY_NAME="$(dd if=/dev/urandom bs=32 count=1 2>/dev/null | base64)"
NEW_DUMMY_NAME=$(echo chromium-key-dummy-file-$NEW_DUMMY_NAME | tr / -)
#echo new dummy file "$NEW_DUMMY_NAME"
NEW_DUMMY="$EXTENSION_DIR/$NEW_DUMMY_NAME"
touch "$NEW_DUMMY"
sed -i "s/$OLD_DUMMY_NAME/$NEW_DUMMY_NAME/" "$MANIFEST"
zip "$WORKING_ZIP" "$NEW_DUMMY" "$MANIFEST" > /dev/null
cat "$WORKING_ZIP"
```

#22 - 09/14/2021 08:22 PM - koszko

- % Done changed from 70 to 90

- Description updated

OK, it seems all that's important is ready. Documentation will never be perfect but it's already sufficiently good. I haven't made a signed .xpi yet, but I will make one when I find another couple of minutes...

So, I believe we can start asking people to come and use Haketilo :)

#23 - 09/14/2021 11:09 PM - jahoti

> As a rather unimportant aside, however, we have yet to establish a clear difference between "Haketilo" and "Haketilo □□". Is it simply a matter of stylistic preference, or is there a distinction in meaning that it might be helpful to give to these?

I meant no difference in meaning for these. Actually, if we made them mean something different, it would be misleading to ppl

Very true- thank you for bringing me back to sense!

> In case anyone's wondering how I automatized the generation of Chromium builds with different secres, it's this script (CC0-licensed):

Thanks for sharing that- I genuinely would have been. If you haven't done so already, I'll try and add artifact generation and some other perks to the build script using this.

> So, I believe we can start asking people to come and use Haketilo :)

YES!!

#24 - 09/15/2021 06:49 AM - jahoti

Note: the 0.1 release is missing the default repository :/.

#25 - 09/15/2021 12:23 PM - jahoti

After a somewhat embarrassing length of time, I've come to the realization the script you posted doesn't actually do most of the CRX packaging, only modify an existing file. I'll still continue working on it in any case (it turns out that is yet another prerequisite for the test suite).

#26 - 09/15/2021 02:45 PM - koszko

> Note: the 0.1 release is missing the default repository :/.

My fault. Updated Releases.

> If you haven't done so already, I'll try and add artifact generation and some other perks to the build script using this.

I'm certainly focusing on something different now :)

> After a somewhat embarrassing length of time, I've come to the realization the script you posted doesn't actually do most of the CRX packaging, only modify an existing file.

Indeed, I only did the generation of an *unpacked* extension build (which is, btw, a quite common way of distributing Chromium builds of extensions...). Since generation of an actual .crx using Google's script requires almost none programming effort on our side, we can quite easily add that feature, though :)

> I'll still continue working on it in any case (it turns out that is yet another prerequisite for the test suite).

□□

#27 - 09/18/2021 06:19 AM - jahoti

- % Done changed from 90 to 100

- Status changed from New to Closed

"andyprough" has offered some outlets for spreading the news at https://trisquel.info/en/forum/announcing-haketilo-01-first-release-favor#comment-160449, BTW.

Files

mozoid.py	11.9 KB	09/13/2021	jahoti