Mozilla Firefox Hardening: uMatrix and Multi-Account Containers

Kyle Terrien

May 16, 2021

Contents

1 Preface: Deprecation Notice (2021-05-16)

This document is now end of life. I suspect soon Mozilla Firefox will be irrelevant too.

A lot has happened in the year since I wrote most of this material.

Most importantly, last September Raymond Hill announced end of support for uMatrix. I'm a little disappointed but not very surprised, because uBlock Origin is more mainstream and does 80% of the work without burdening the user to maintain whitelists of websites.

Secondly, Mozilla laid off at least a quarter of its employees while at the same time revealing an executive pay increase of 400% over several years. Skipping much lamentation, Mozilla and its flagship product Firefox are circling the drain more rapidly than ever before. (In this way, Mozilla has fulfilled the "rags to riches to willing its own self-destruction" story line, a surprisingly common story line in modern day literature and theater.)

Plus with all of the gratuitous (in the negative sense) user interface changes, I am finally fed up with Mozilla's continual "We know better than our users" attitude towards the dwindling remainder of their Firefox user base (currently below 5%). I thought this "revolutionary" period of lifting the codebase onto Electrolysis was over. I wonder, Mozilla: who are your real stakeholders?

Enter the next interim solution: Brave Browser. Yeah, it is Chromium-based, which unfortunately implies deferring to Google's decisions of which web standards to implement. On the bright side, being Chromium-based means that Brave will work with 95% of the Web, and the Chromium backend has an excellent profile system for security by compartmentalization. The "shields up" feature works pretty well, blocking ads, nasty scripts, and third party cookies by default. Further, Brave's political positions do not bow to the "techleft" of Silicon Valley, instead offering practical solutions to political problems on the modern web.

As far as additional security hardening goes, I know about NoScript. Maybe I will use, or maybe not. I also had a revelation about hardening tools in general. Once while cleaning up after a "cookie spill" from accidentally opening a tab in the wrong container, I looked with mild frustration into

the eyes of Saint Philomena (as I am wont to do on such occasions), and she seemed to say to me, "Why do you bother with all of these extra hurdles you build for yourself? What is the benefit?" To which, all I could answer was, "I don't know." So, I need to figure out something better. The modern web is insane, but it's wrong to let it drive you insane.

Mozilla's whirlpool of self-destruction is circling ever tighter. Please offer up your thoughts and prayers for the future of the web.

- `https://www.ghacks.net/2020/09/20/umatrix-development-has-ended/`
- `https://calpaterson.com/mozilla.html`

2 Introduction

When you browse the web, do you feel like somebody is watching you? You should! Many modern websites have Cthulhic back-ends.

Fortunately, you still have control of your web browser. So it is possible to tame your web browser. In this article, we take a look at Mozilla Firefox and two Firefox extensions: uMatrix and Multi-Account Containers. When configured properly, these tools can block the nasty parts of the web. Blocking the nasty parts of the web makes your web browser more responsive and frustrates the bad guys from profiling you.

This document is intended as a crash course on how to use uMatrix and Multi-Account Containers effectively.

3 Revision History

Version	Date	Comment
0.2	2020-02	Initial revision
0.3	2020-04	Returning after a long break
0.9	2020-04-26	First semi-public draft
0.91	2020-05-17	Add screenshot of `about:config`
0.95	2020-05-26	Proofreading pass
1.0	2020-06-01	First release
1.1	2021-05-16	Preface: deprecation notice

4 Why?

I dislike advertisers who try to profile me into a set of predefined simplified buckets. I find it demeaning to be treated as a member of one of many

poorly defined categories. Such behavior is tolerable when it is isolated in one context. For instance, once I walk out of a car dealership, the salesman who erroneously thinks I want an SUV when I actually want a small hatchback will never bother me again.

But the modern web doesn't work like that. There is no obvious way to "walk out of the store" because the web advertiser is a cross-site entity who follows me into the next store! If I search for new hiking boots on one website, then I will see ads for hiking socks on several other websites. On the web, expressing a brief interest for a specific item in location A somehow grants a advertiser a license to follow me to location B to try to persuade me to buy a related item. Compare this to the real world. In the real world, as soon as I walk out of the store, it's none of the advertiser's business to follow me to the next store. If a advertiser followed me around in the real world on the same massive scale as he does on the web, he might face a stalking charge.

Like real world creeps, the best thing you can do is refuse to interact with virtual creeps. With the right technical measures in place, it is possible to give the advertisers a big arms-crossed "No way!" gesture.

If you combine uMatrix with Firefox Multi-Account Containers, then you can create a hardened configuration of Mozilla Firefox that can both block nasty parts of the web (the advertisers) and segregate personally identifiable data (so the advertisers cannot easily identify you).

> A note about NoScript: there are plenty of guides about NoScript. I have used NoScript on a number of occasions. It was good, but I like uMatrix more. uMatrix is a lot more powerful and generalized than NoScript. I think uMatrix doesn't get the credit it deserves.

5 Platform

This guide was written with Firefox 76 in mind. The extensions used also install on the current Extended Support Release (ESR) as of this writing: Firefox 68. There is a reasonable argument for installing the ESR: fewer abrupt changes. Which version of Firefox you install is your decision.

Screenshots are on Funtoo Linux.

6 Sweep through the Preferences

Firefox has a whole host of options, including many pages of options that aren't in the settings dialog. Here are my customizations. Some of these options are security-related. Some are personal preference.

First, sanitize the UI.

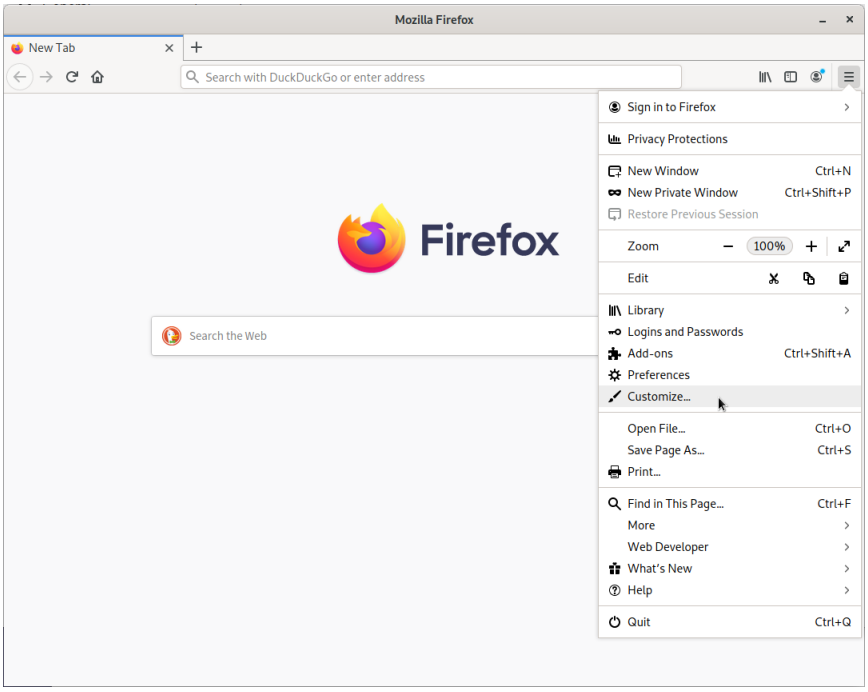

Figure 1: Where the 'Customize' option hides.

1. Go to the "Three-Bar" Menu > Customize... (See Figure 1.)
2. Remove the extra flexible spaces from the toolbar. To remove an item, either drag the element away from the toolbar or right click > Remove from Toolbar. The address bar should be larger after removing the flexible spaces from the left and right sides of the address bar.
3. Remove all undesired buttons from the toolbar.
4. At the bottom of the customization window, select Toolbars > Menu Bar. This setting displays the traditional menu bar.

5. At the bottom of the customization window, set the density to 'compact'.

Then, sweep through the preferences screens. (Edit > Preferences, Tools > Options, or "Three-Bar" Menu > Preferences)

On the General tab: Select 'Always ask where to save files', disable DRM-controlled content, and select 'Check for updates but let you choose to install them'.

On the Home tab: set your home page, disable everything on Firefox Home (the new tab page) except the search bar.

On the search tab: Set default search engine to DuckDuckGo [1] and disable search suggestions.

The Privacy/Security tab is where the guts are (Figure 2). Set Enhanced Tracking Protection to 'Custom'. Then, block all third-party cookies, block tracking content in all windows, block cryptominers, block fingerprinters, and enable 'Do Not Track'.

Further down the page (Figure 3), disable the option to save logins and passwords, select 'Use custom settings for history', and disable search and form history.

Then, **disable telemetry** (Figure 4), disable personalized extension recommendations, and **disable studies**.

The finished product looks similar to Figure 5.

7 Additional about:config Tricks

Do you remember I mentioned there are pages of hidden options in Firefox? Point your browser to `about:config` so that we can set them. (See Figure 6.) Firefox will warn you the first time you open `about:config`. This is nothing to worry about. You are a true computer geek.

I recommend setting these options in `about:config`.

`browser.urlbar.formatting.enabled = false` In the address bar, disable the emphasis on the domain name.

`browser.urlbar.trimURLs = false` Show the `http` and `https` strings explicitly instead of truncating them. The default truncation frustrates proactive security. (Is the page using HTTPS?)

`browser.urlbar.openViewOnFocus = false`

[1] `https://duckduckgo.com/`

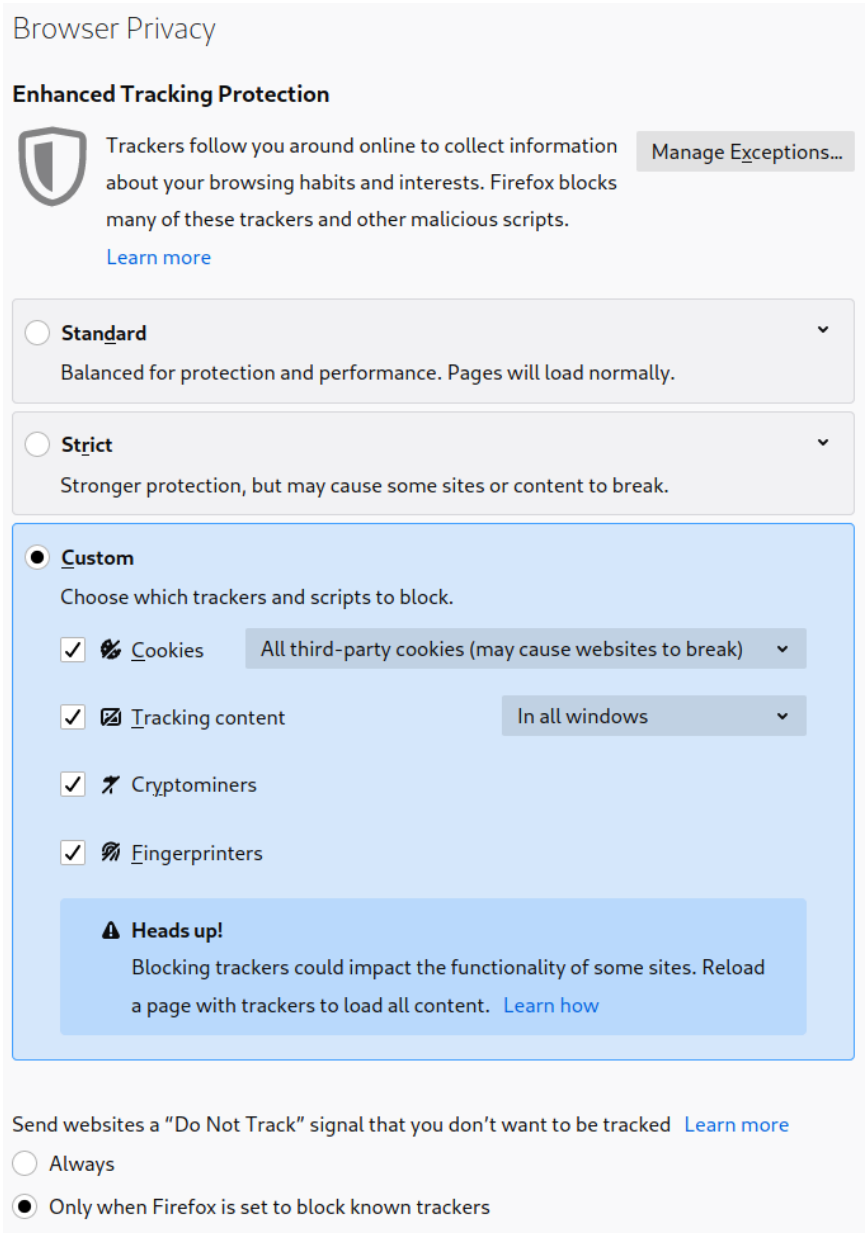

Figure 2: Privacy & Security > Enhanced Tracking Protection

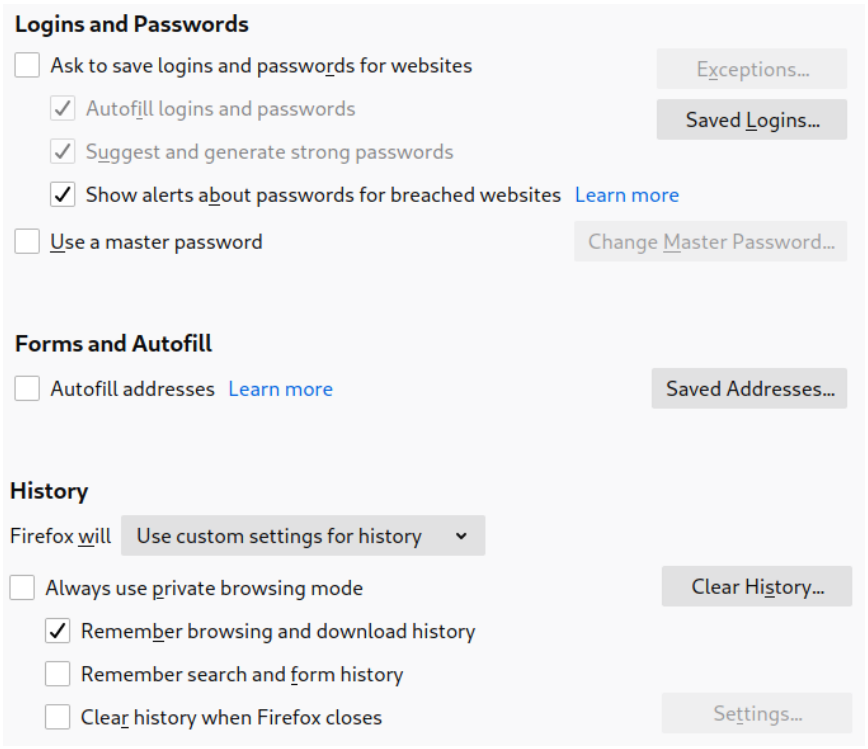

Figure 3: Disable passwords and autofill. The breached website warning might be useful.

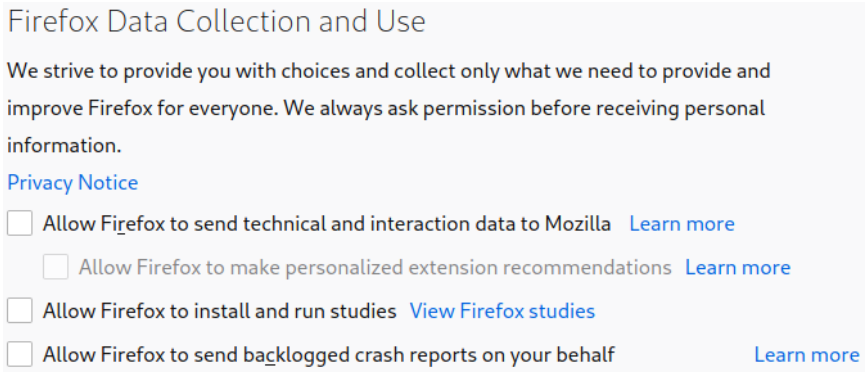

Figure 4: Disable telemety.

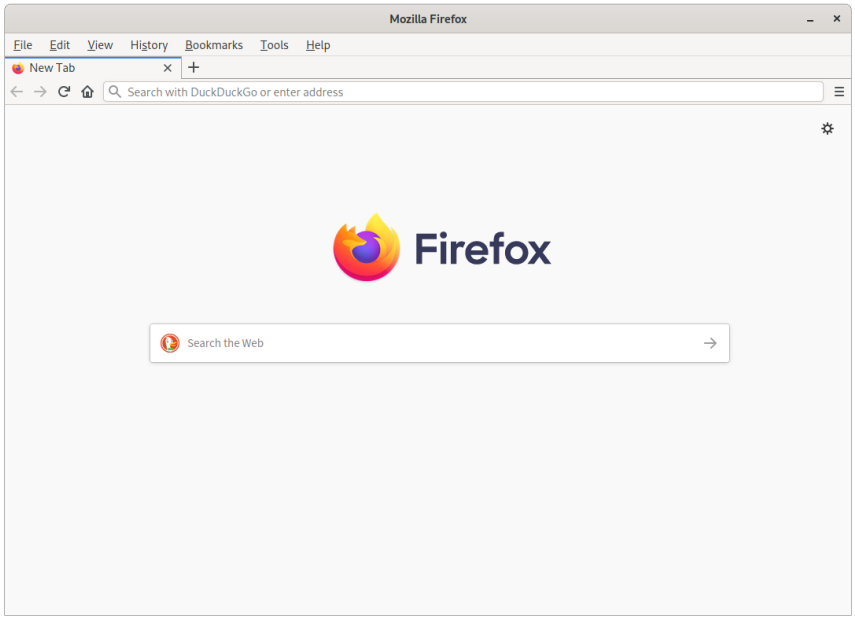

Figure 5: Mozilla Firefox sans unecessary distractions

`browser.urlbar.update1 = false`

`browser.urlbar.update1.interventions = false`

`browser.urlbar.update1.searchTips = false`

`browser.urlbar.update1.view.stripHttps = false` Rollback the so-called "enhancements" made to the address bar in Firefox 75. I don't like things popping up in my face. (Big thank you to Dedoimedo [2].)

`extensions.pocket.enabled = false` Disable Pocket integration. A text file of URLs works well enough.

`extensions.screenshots.disabled = true` Disable the screenshot service that uploads screenshots to Mozilla. The screenshot button in Developer Tools [3] and 'Save Page As...' are good offline alternatives.

`media.peerconnection.enabled = false` Disable WebRTC, the UDP-based peer-to-peer outgrowth of HTTP.

[2] `https://www.dedoimedo.com/computers/firefox-75.html`
[3] `https://developer.mozilla.org/en-US/docs/Tools/Taking_screenshots`

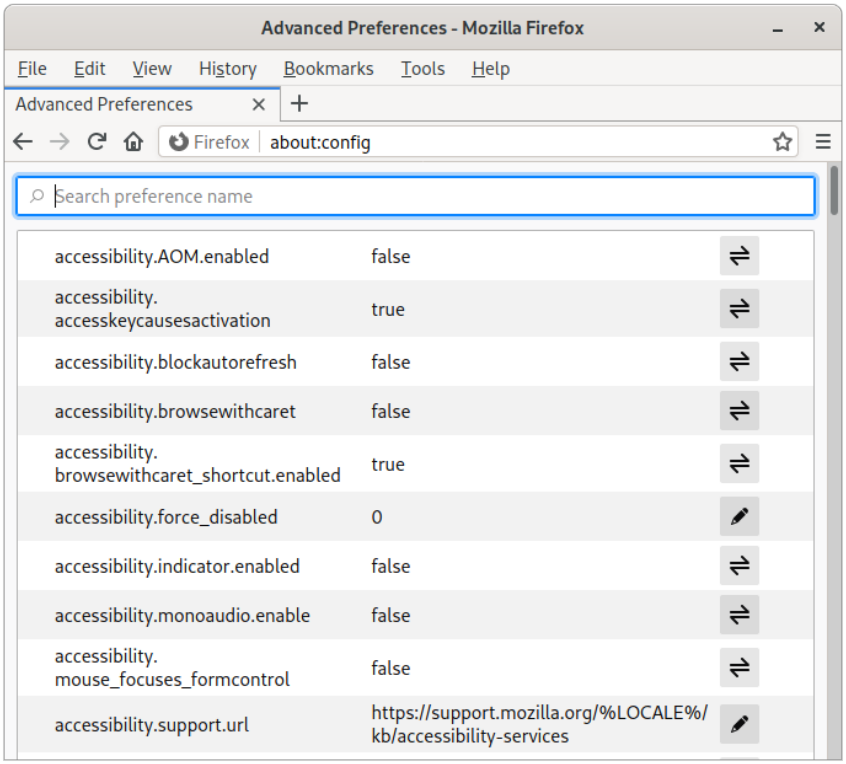

Figure 6: `about:config`, where the many hidden options of Mozilla Firefox reside. There be dragons here!

8 Tool #1: Firefox Multi-Account Containers

Download and install the Firefox Multi-Account Containers extension [4] from Mozilla. This is a nifty tool that lets you segregate your data (cookies and local storage) into multiple **containers** (also called "contexts" in other parts of the browser).

Data written in one container cannot be accessed from another container. So, if you create one container for Facebook and another container for work, then Facebook cannot access data in the work container and vice versa. The pattern is called security by compartmentalization, or less formally "the Great Wall of China approach."

Click the new Multi-Account Containers button on the toolbar (Figure 7), and create containers as you wish.

Try not to obsess over containers at this point. We will install uMatrix to further control which sites are allowed to store cookies.

I like to use the default container as an ephemeral area that can be easily wiped. So, any login cookies that I care about go into a Firefox container.

My primary advice: create isolated containers for each social media website. I.e. one for Facebook, one for LinkedIn, and another for Google. Social media sites started a bad trend of convincing other website operators to include "like" and "share" widgets on their web pages. These widgets phone home to the social media site about the web page you are viewing. So, if you visit such a web page, the social media site automatically knows about it.

Think about how often you see these "share" widgets. Do you really want Facebook knowing that you looked up how to tie a noose out of curiosity? Probably not!

9 Tool #2: uMatrix

Now, we come to the blocking tool: the venerable uMatrix [5]. Go ahead and install uMatrix.

After installing uMatrix, you will see the uMatrix button on the toolbar. Click the uMatrix button to open the uMatrix control panel. (See Figure 8.)

The uMatrix popup panel [6] shows elements on the current page by domain (in rows) and type of element (in columns). Green items are allowed

[4] `https://addons.mozilla.org/firefox/addon/multi-account-containers/`
[5] `https://addons.mozilla.org/firefox/addon/umatrix/`
[6] `https://github.com/gorhill/uMatrix/wiki/The-popup-panel`

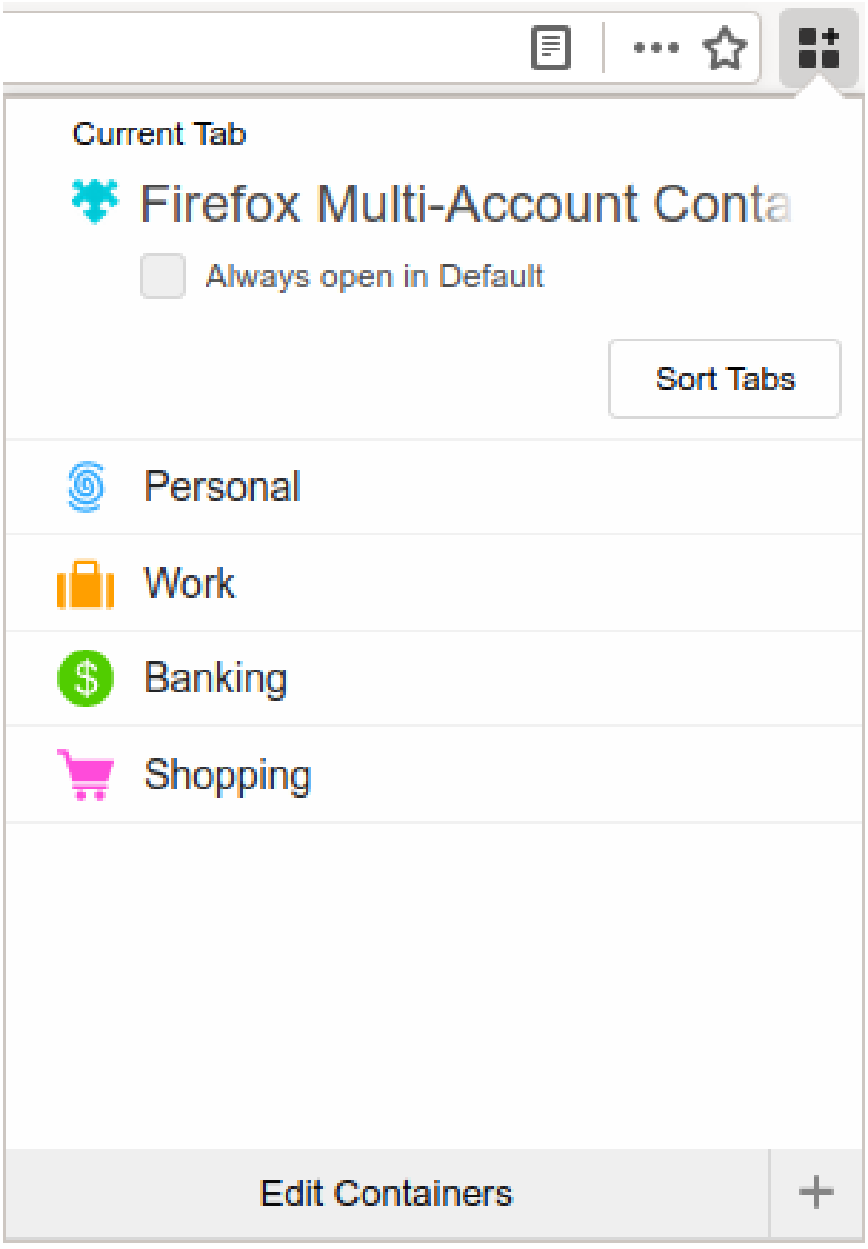

Figure 7: Firefox Multi-Account Containers: default configuration

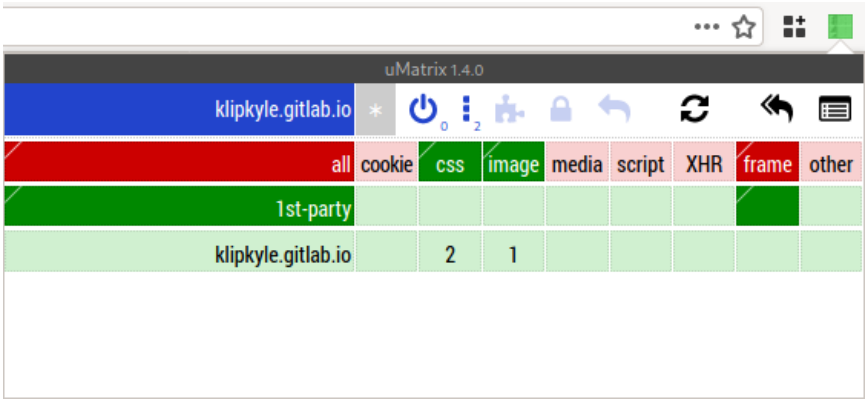

Figure 8: uMatrix control panel: default configuration

through, and red items are blocked. Dark green indicates an explicit whitelist rule. Dark red indicates an explicit blacklist rule. All other pale cells indicate inherited rules. You can set a whitelist or blacklist rule by clicking on the corresponding cell. The number on each cell is the number of items allowed or blocked by the cell's rule.

The categories (columns) are as follows:

Cookie Key-value data used to identify you to a site. Some cookies are legitimate, and some are used by advertisers to track you.

CSS Cascading Style Sheets (CSS) define formatting rules for the page.

Image Images on the page.

Media Audio, video, and plugins.

Script Executable code (JavaScript) that runs inside the browser. Scripts are used on interactive pages and web applications. Malicious scripts send data to advertisers. Very malicious scripts may try to lock you out of the browser.

XHR XMLHttpRequest. XMLHttpRequests are resource requests that scripts initiate after the page finished loading. XMLHttpRequests are common in interactive web applications.

Frame A web page embedded within a web page. The embedded web page can be on the same domain or a different domain.

Other Anything else.

All rules are applied within a per-domain context. To create a more detailed or more general rule, click on the desired part of the domain name to which you wish to apply the rule. Click on the * to configure global rules. (See Figure 9.)

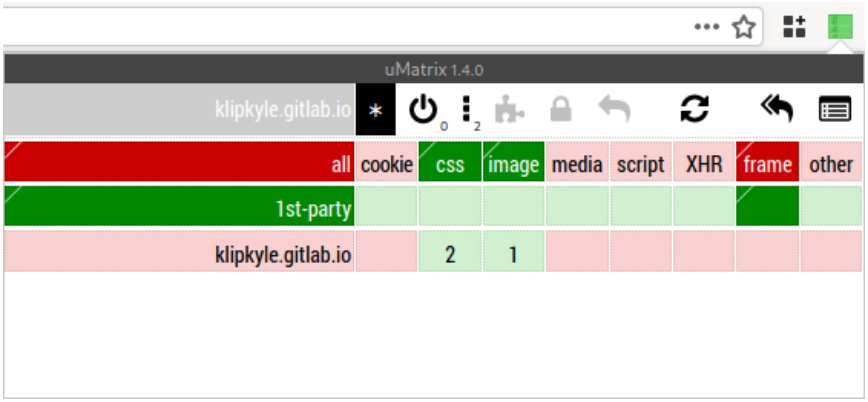

Figure 9: Default global rules in uMatrix

uMatrix is a general purpose tool, so we must configure it in a desirable way. Here is how I configure uMatrix.

First, go to the uMatrix settings screen. Click on the gray caption bar at the top of the uMatrix window. The uMatrix dashboard will appear.

On the 'Settings' tab (Figure 10), enable 'Delete blocked cookies' and 'Delete local storage content set by blocked hostnames'. uMatrix does something a little strange with cookies and local storage: by default uMatrix allows all cookies and local storage to be set but will send neither cookies nor local storage to blocked domains. The rationale behind this quirk is to allow you to inspect the cookies' contents to decide whether you want to keep them. I don't care much for cookies I block, so it is safe to go ahead and delete them.

Next, move to the 'Assets' tab (Figure 11) and update the advertising blocklists. uMatrix blocks advertising domains by default.

> Firefox is unaffected by the new restriction in Google Chrome's v3 manifest which caps the number of filters an add-on can apply. Yet another reason to use Firefox!

Finally, open the 'My Rules' tab (Figure 12). This tab contains a plain text listing of all rules in uMatrix. uMatrix has a permanent section and

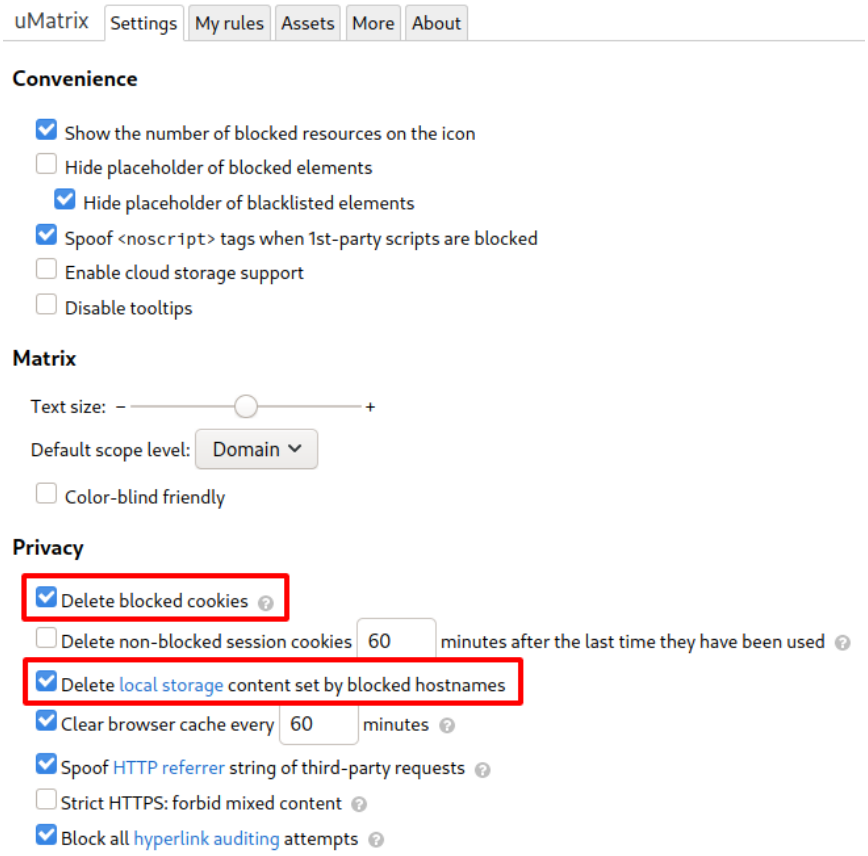

Figure 10: Settings tab of the uMatrix dashboard

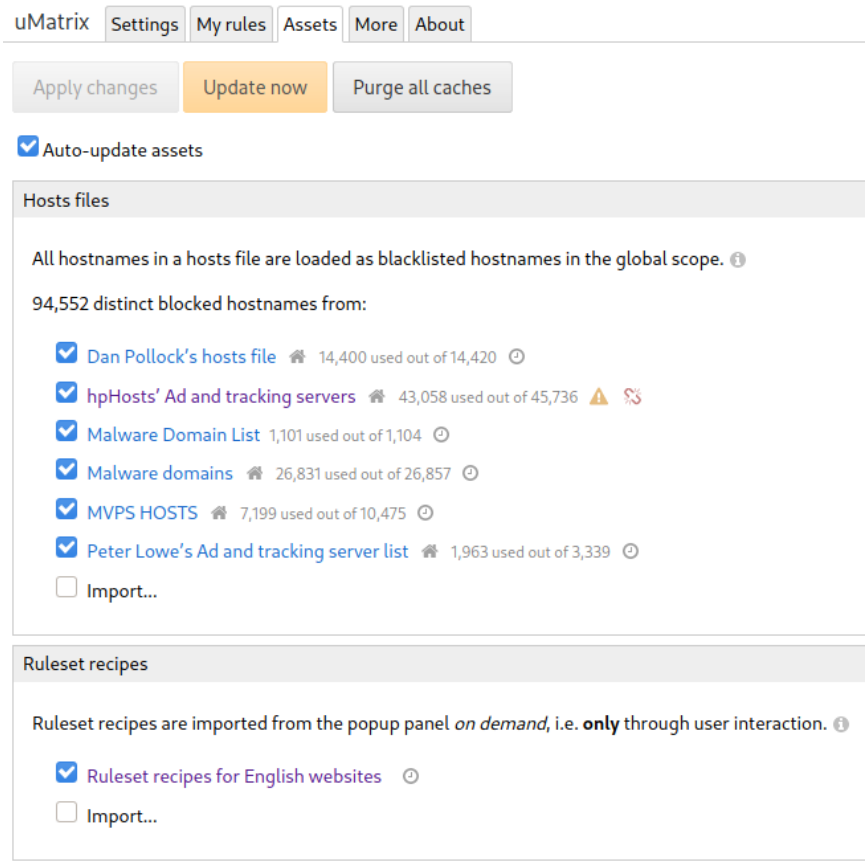

Figure 11: Assets tab of the uMatrix dashboard

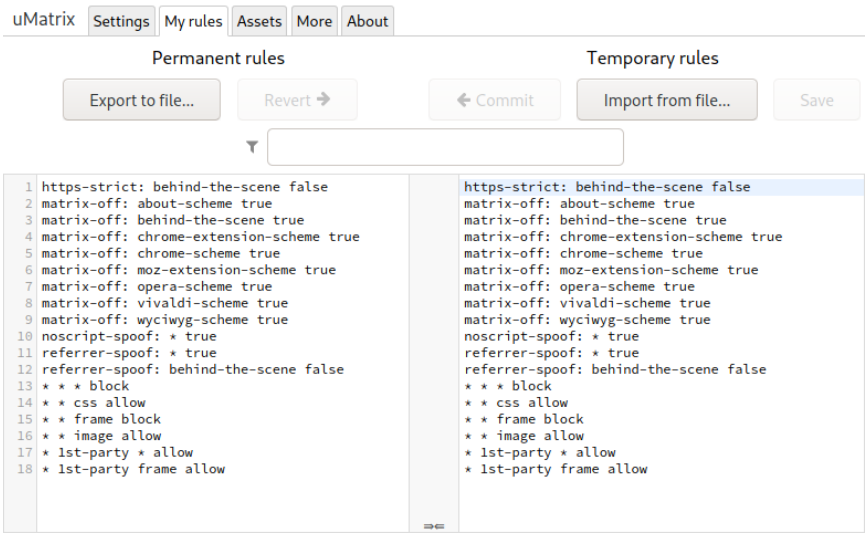

Figure 12: 'My rules' tab of uMatrix dashboard, showing the default set of rules

a temporary section. You can make changes in the temporary section and revert them easily. Also, you can import and export the rule list.

Now, close the uMatrix Dashboard and load a regular web page.

> Do not load mozilla.org because mozilla.org is administratively whitelisted.

Open uMatrix and select * to configure global rules. (See Figure 13.) Remove the first-party whitelist and the first-party frame whitelist. Then, globally block cookies and frames by clicking on the 'cookie' and 'frame' column headers.

The global rules should look as above. Try to match your configuration with mine.

If you open the uMatrix Dashboard, you will see that the corresponding text rules are as follows.

```
* * * block
* * cookie block
* * css allow
* * frame block
* * image allow
```

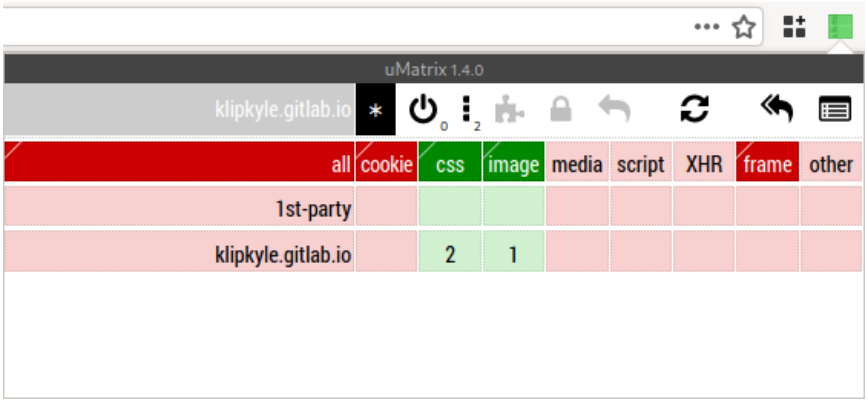

Figure 13: My global rules in uMatrix

Rationale: I want to block everything by default except the bare minimum required to render a static web page (images and CSS). If I trust a website, I will explicitly whitelist the domain.

Cookies are especially dangerous because they allow a site (or advertiser) to identify me. So, I want to block cookies, even on trusted domains. If I want to allow a website to identify me, then I will explicitly whitelist cookies for the domain.

I also want to block frames because frames allow another domain to load inside the context of the current domain. So life from a security standpoint can become complicated rather quickly when frames are involved. If I want to use frames, I will explicitly whitelist them.

Now let's navigate to a page that requires JavaScript. By "requires," I mean the page looks conspicuously empty if JavaScript is disabled.

Point your web browser to `https://www.kassandravasquez.com/`

Open uMatrix. You will see that almost everything is blocked. (See Figure 14.)

This site requires scripts. I trust the site owner, so let's whitelist the domain. Click the row header `kassandravasquez.com` so that it turns dark green. (See Figure 15.)

Click the reload button. The page with all of its pretty pictures should display correctly. Notice how all the other extra domains (e.g. pinterest.com) still have their scripts blocked.

What about cookies? Notice how when whitelisting a domain, cookies and frames are still blocked. If you want to give a website permission to identify you (e.g. to log in to the website), then you need to enable cookies

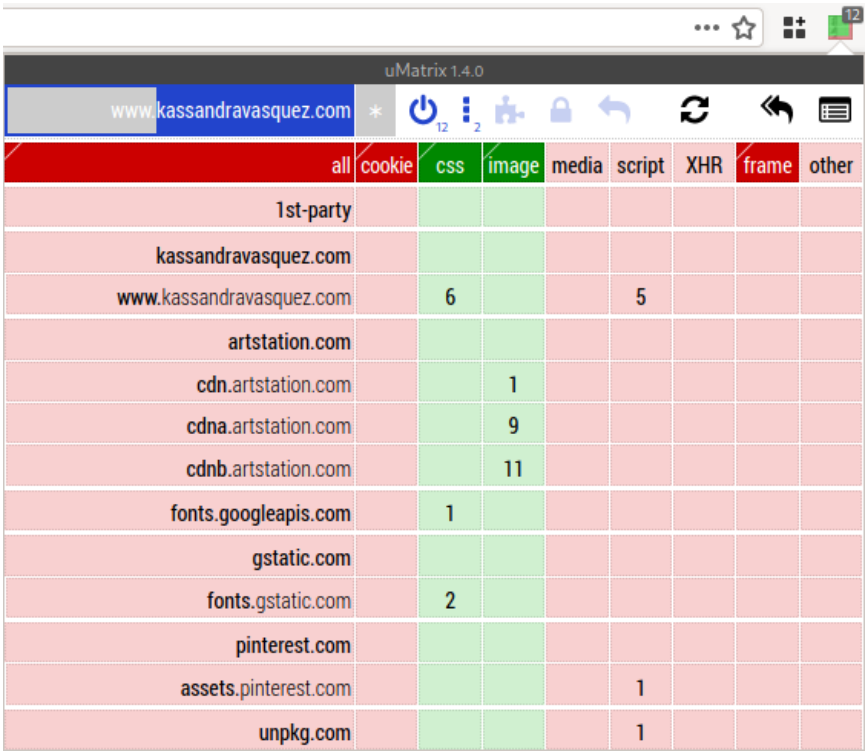

Figure 14: Almost everything is blocked.

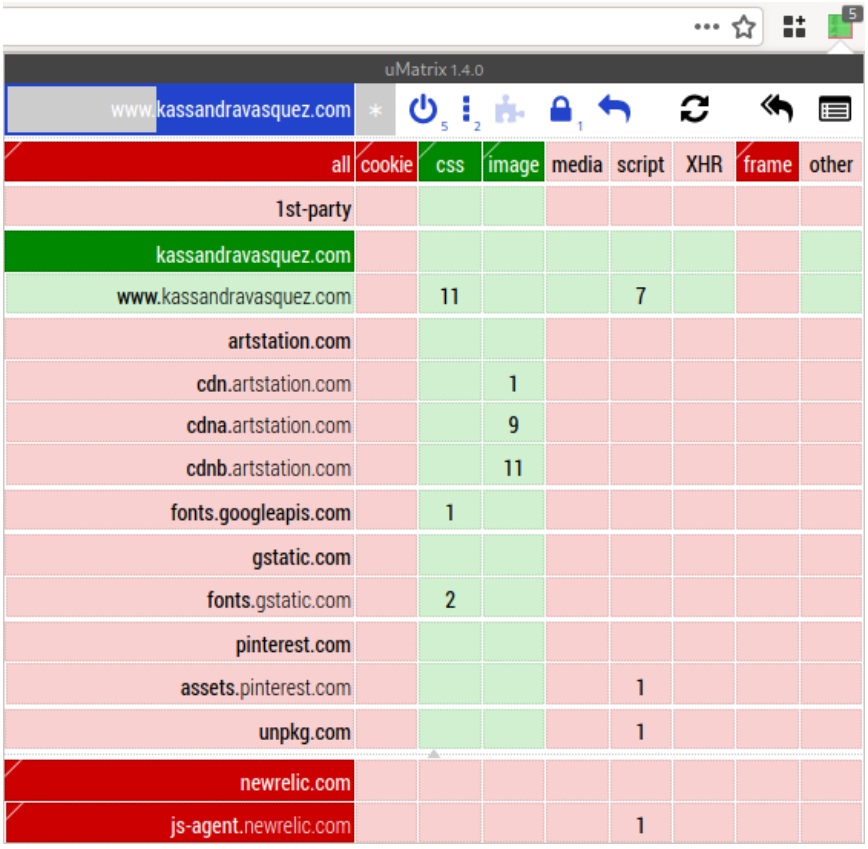

Figure 15: Whitelisting the domain

for the domain. To enable cookies for the domain, click on the cookies cell next to the domain `kassandravasquez.com` and turn it dark green. (See Figure 16.)

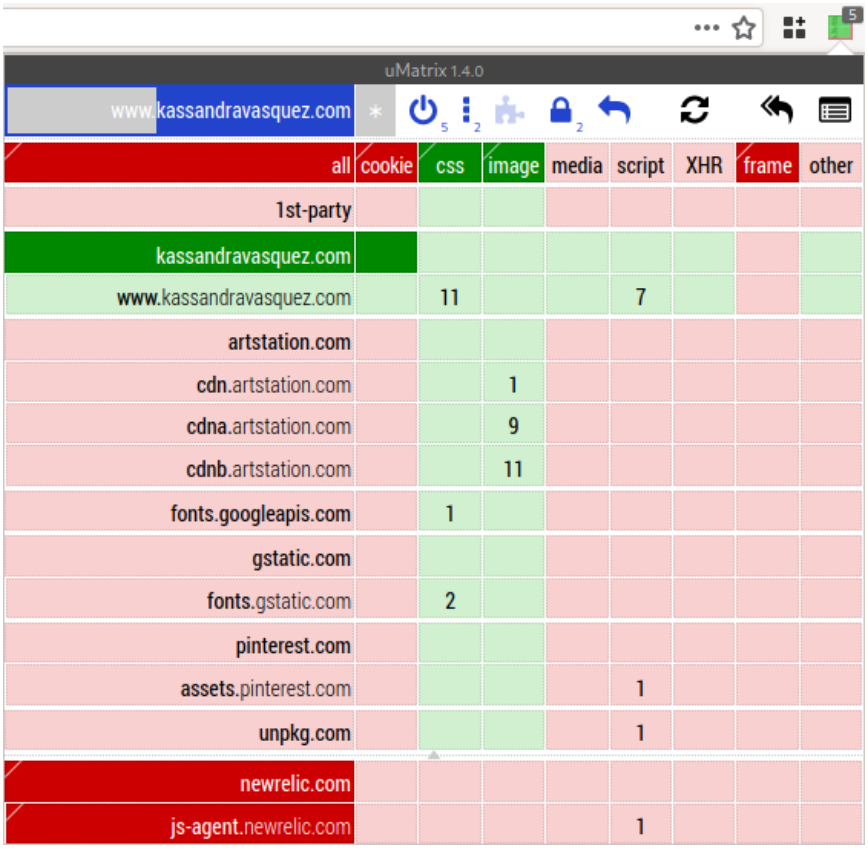

Figure 16: Enabling cookies for the domain

> Click reload again. When configuring rules for a website, you will click reload several times. For this reason, uMatrix provides a reload button in the uMatrix control panel.

Once you are happy with the site-specific rules, click the padlock icon to save them. In the above example, the generated rules are as follows.

```
kassandravasquez.com kassandravasquez.com * allow
kassandravasquez.com kassandravasquez.com cookie allow
```

Figure 17: My uMatrix on funtoo.org

Let's consider a more complex example.

Figure 17 is a configuration on a site to which I regularly log in. Therefore the domain needs cookies enabled. Also, there are some frames on the project's bugtracker, and there are a few embedded Youtube videos on the home page (more frames). Therefore I need to enable frames on a couple of domains. Notice how the rules are inherited. I.e. if I enable frames for `youtube.com`, then frames for `www.youtube.com` are also enabled.

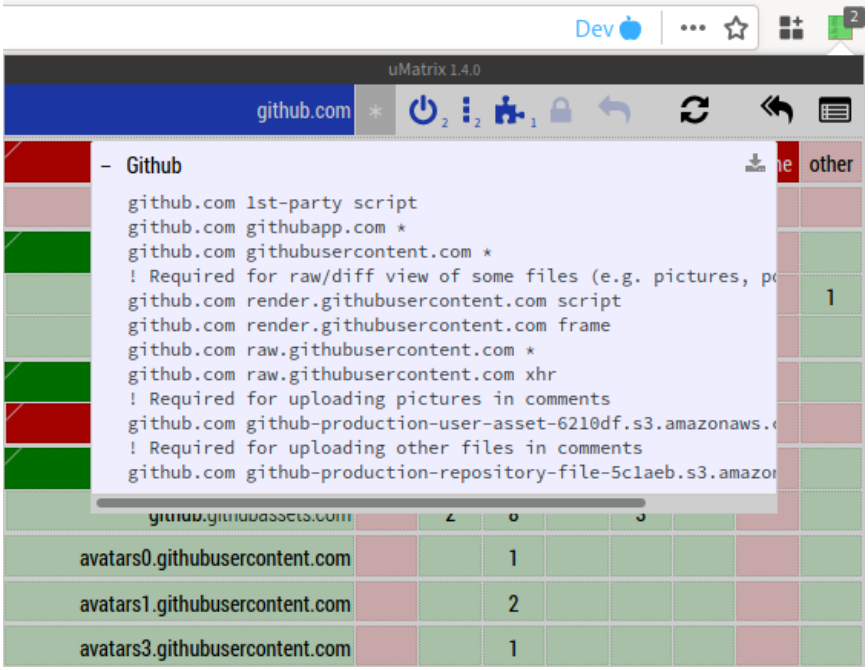

Figure 18: uMatrix suggestion for GitHub

Ruleset recipes [7] make the process of creating site-specific rules easier. (See Figure 18.) Ruleset recipes are templates of common site-specific configurations. When uMatrix has one or more ruleset recipe suggestions, the puzzle icon in the uMatrix control panel is enabled. To inspect the recipes, click on the puzzle icon. To add a recipe, click on the download icon next to the recipe name. Ruleset recipes are a lifesaver for the websites that use Google ReCaptcha.

After a few days of adding rules as you browse, your rule list will begin to stabilize. At this point, your workflow will change. You can whitelist sites

[7] `https://github.com/gorhill/uMatrix/wiki/Ruleset-recipes`

temporarily and then remove the temporary rules by clicking on the revert arrow.

10 Other Nifty Tools:

10.1 Cookie Quick Manager

Cookie Quick Manager [8] is a nice cookie manager. You can view cookies, edit them, and import/export them – much more than you used to be able to do with the builtin cookie manager during its heyday. Cookie Quick Manager is also container-aware, so it works well with the Multi-Account Containers extension.

> Be careful with exporting cookies that are in containers. Each container has a numeric ID, and the numeric IDs may not necessarily match on two machines. If you want to backup your cookie store, consider dumping all cookies into the default container temporarily. Then, when importing cookies, move each cookie into its correct container.

10.2 Absolute Enable Right Click & Copy

Some pages are intent on making life difficult and will try to disable right-click [9] and copy/paste. Fortunately, there is an extension to fix that: Absolute Enable Right Click & Copy [10] (Figure 19)

10.3 GhostText

GhostText [11] deserves its own article because its configuration is heavily-involved. GhostText allows you to open a text field in an external text editor. (See Figure 20.)

10.4 User-Agent Switcher and Manager

Sometimes, web sites discriminate against your web browser. User-Agent Switcher and Manager [12] allows you to spoof the User-Agent string of another

[8] `https://addons.mozilla.org/firefox/addon/cookie-quick-manager/`
[9] `http://turnoff.us/geek/welcome-to-hell/`
[10] `https://addons.mozilla.org/firefox/addon/absolute-enable-right-click/`
[11] `https://addons.mozilla.org/firefox/addon/ghosttext/`
[12] `https://addons.mozilla.org/en-US/firefox/addon/user-agent-string-switcher/`

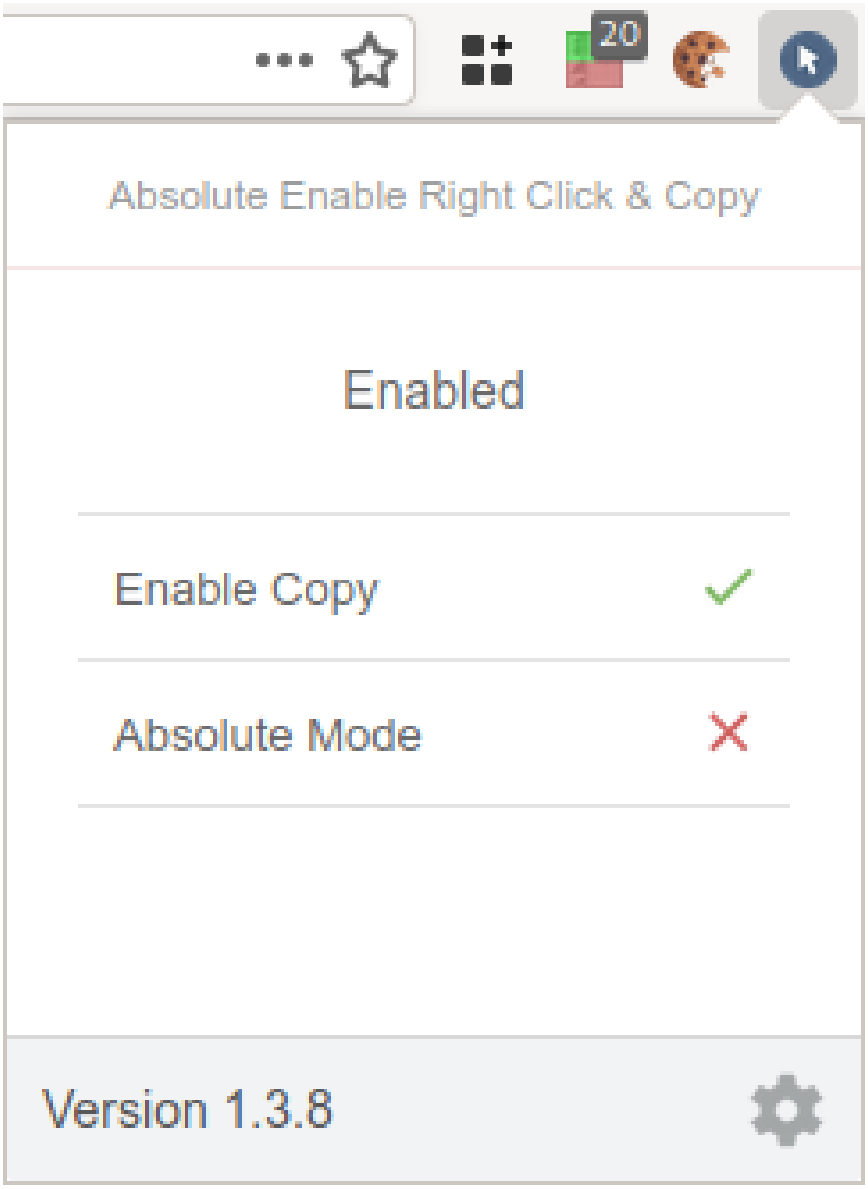

Figure 19: Down with obnoxious sites that disable the right-click menu!

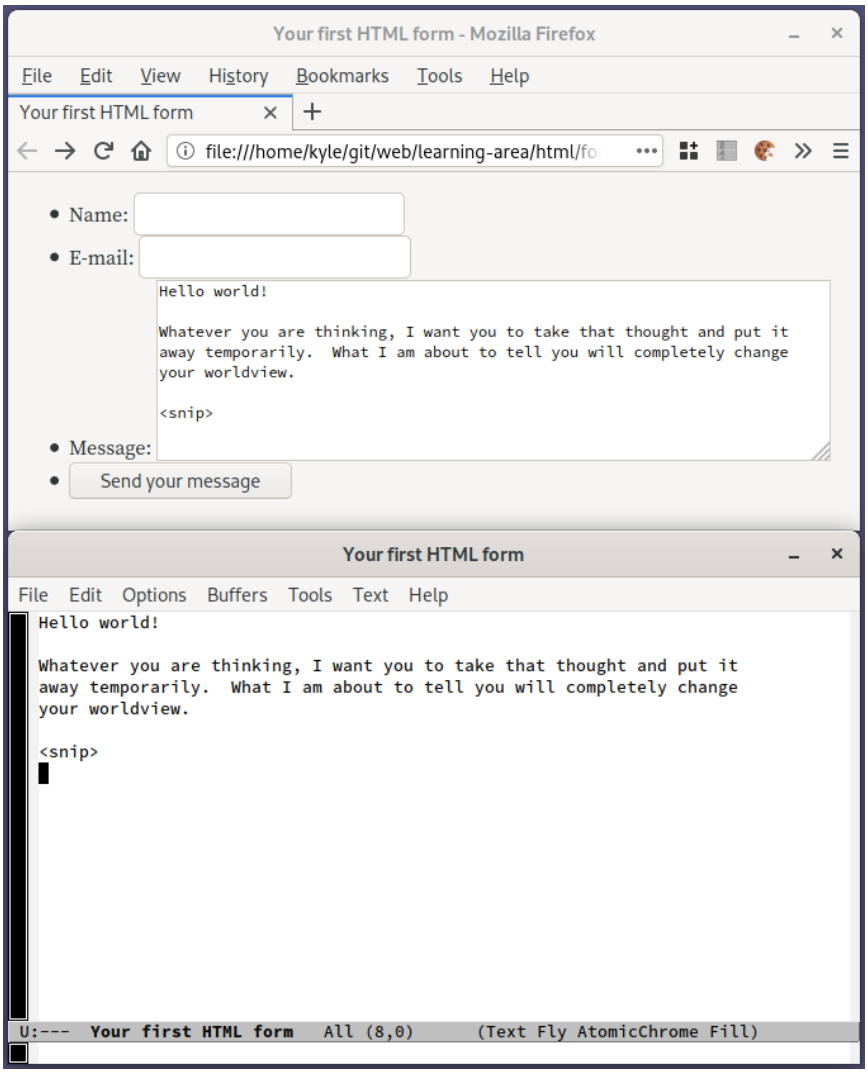

Figure 20: GhostText in action: editing a <textarea> with GNU Emacs.

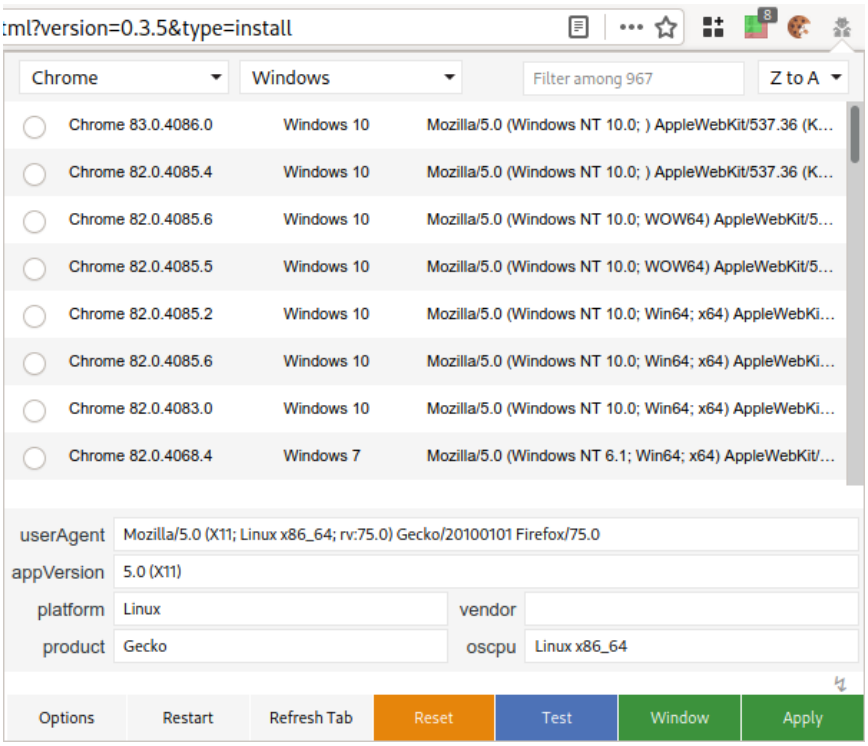

Figure 21: User Agent Switcher and Manager

browser. This is a useful tool about which to know in case you ever need it. (See Figure 21.)

11 Weaknesses and Pitfalls

Problem: some sites are very complicated, and it is easy to reach the point where you spend more time racking your head than being productive.

Solution: don't panic. Load the site in a private browser window. No addon runs in a private browser window, so no uMatrix rules apply. When you are done, close the private browsing window, and all data accumulated in it will be deleted.

Problem: it is easy to accidentally navigate in the wrong container to a site that has cookies whitelisted.

Solution: The Multi-Account Container extension allows you to configure a website to always open in a certain container. Simply check the box 'Always open in <container>'. (See Figure 22.)

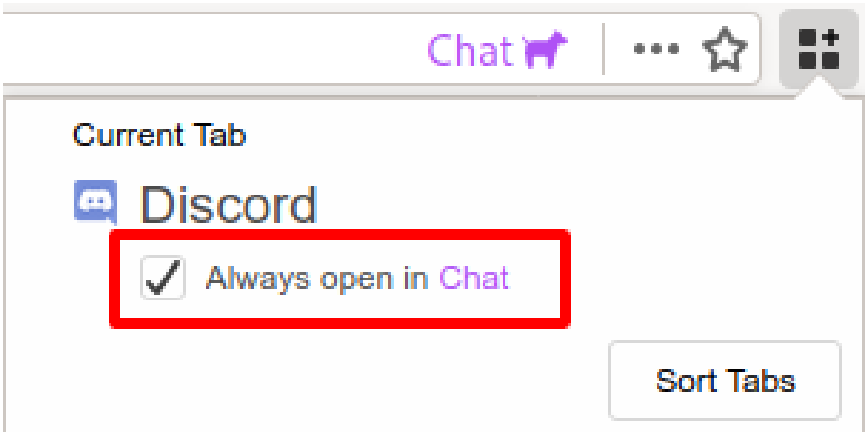

Figure 22: 'Always open in <container>' (available in the Multi-Account Container pulldown menu)

In case you need to cleanup after a cookie spill, Cookie Quick Manager has a couple of buttons to quickly clear cookies in the current context. (See Figure 23.)

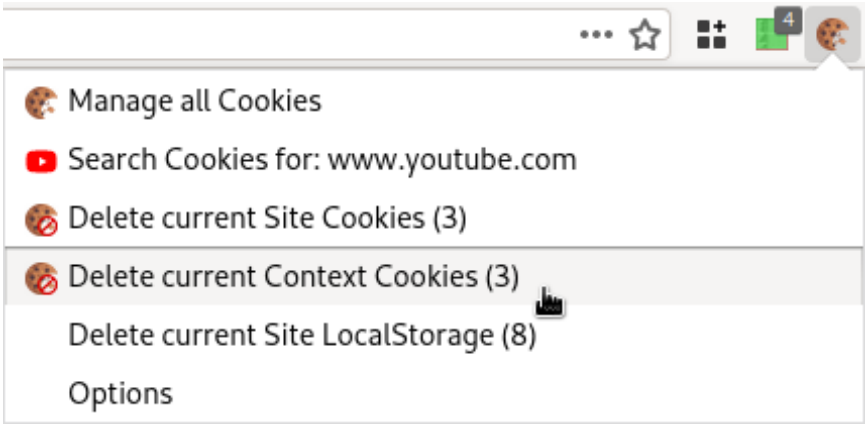

Figure 23: "Delete current Context Cookies" (available in the Cookie Quick Manager pulldown menu)

12 Other Guides

Dedoimedo's exposition on WebExtensions – Two years later [13] convinced me to take a serious look at Firefox. Most of the tools here are reviewed there also. Thank you, Dedoimedo!

Also, Dedoimedo gives a good argument for why you should use Firefox [14]. Bottom line: the only formidable opponent preventing Google Chrome from attaining a mindshare monopoly is Mozilla Firefox.

13 Conclusion

On the web, your enemy is the advertiser who follows you across domains. Every day there is more babble online about security problems on the Internet, but despite all the hand-wringing no productive solutions are formulated. I give you one in this article. If you are tired of advertisers becoming increasingly creepy, then take some initiative. Embrace your inner geek, learn a little about how your Web browser works, and learn how to use uMatrix and Multi-Account Containers to protect yourself.

Let's work toward a saner Web that takes security and privacy seriously.

[13] `https://www.dedoimedo.com/computers/firefox-webextensions-value-two-years-later.html`

[14] `https://www.dedoimedo.com/computers/firefox-why-you-should-use.html`

14 Contact Me

Questions? Comments? What does your web security stack look like? Does it involve w3m or mothra? I would love to hear from you. Contact me [15].

[15] `https://klipkyle.gitlab.io/about.html`