Mitigating malicious websites

- Introduction -
- Mitigating requests -
- Mitigating Cloudflare -
- A better way? -

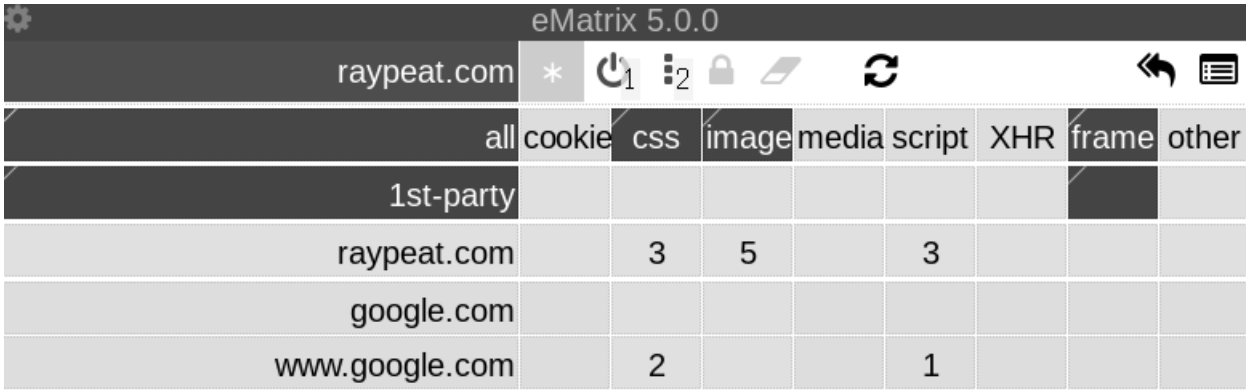
Introduction

The Web part of the Internet is broken, and **websites hate you**. Your average site loads several unneeded third party resources, and for mainstream sites, the figure is 15+. There are several reasons why this is undesirable, the chief one being that those resources can then spy on you in sometimes deep ways (such as collecting mouse movements or browsing history) and send that data wherever (such as to Facebook or Google). The unmitigated websites can also show you stuff you might not want to see (such as advertisements or porn) and the additional resources significantly increase the loading times and waste bandwidth. Then, 20+% of websites are hidden behind Cloudflare, an evil MitM (*Man in the Middle*, a type of attack where a third party intercepts a connection between you and the intended recipient) that collects everything about you, blocks you for arbitrary reasons and can serve modified sites. Even though this is an extremely big issue, almost everyone involved in these topics has either completely ignored it, or even made excuses for it (if a regular hacker MitMed a single big website, it would be front page news; CF does that for 20%? No problem). In this article we will learn how to mitigate

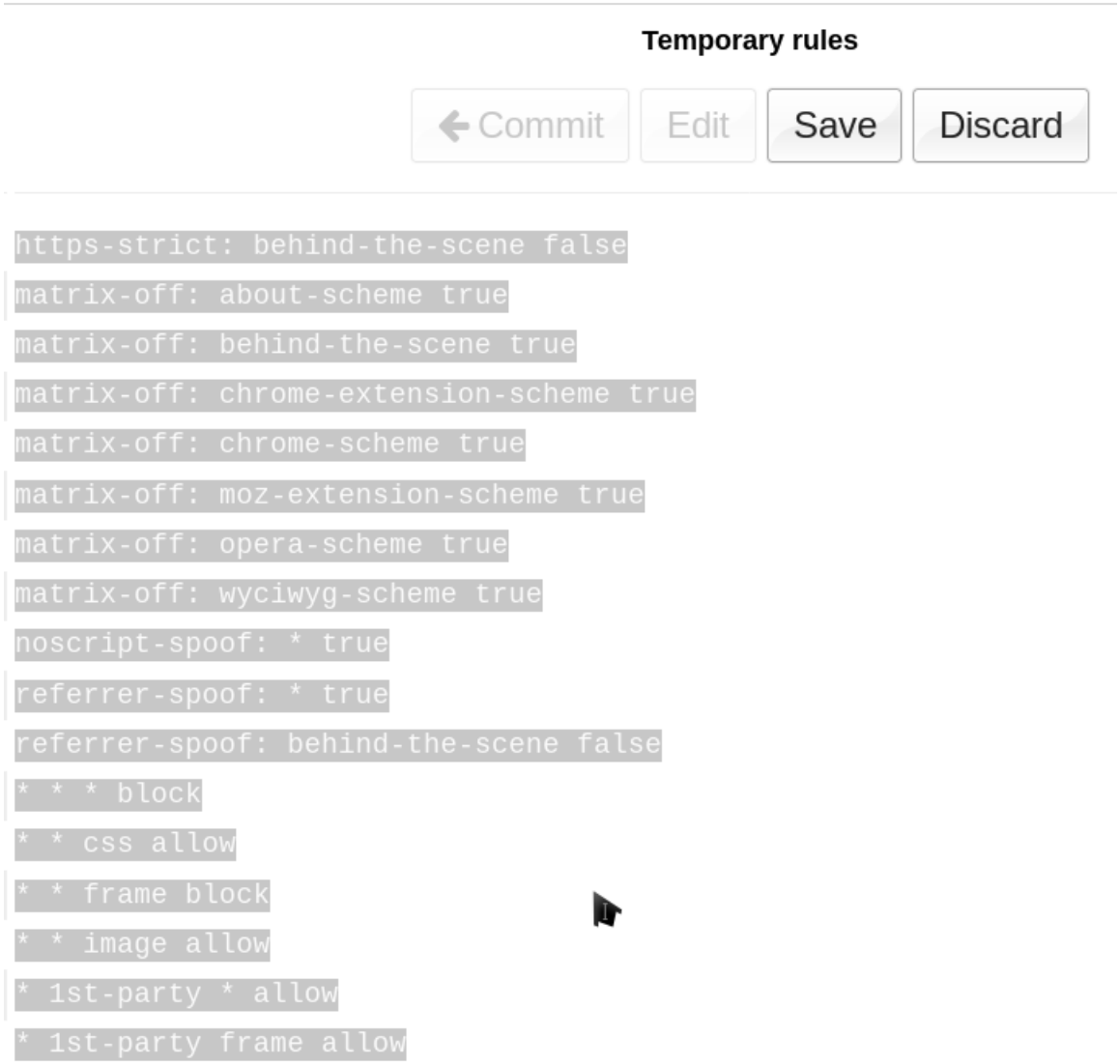
both, and also a long term plan for a better future of the Web.

Mitigating requests

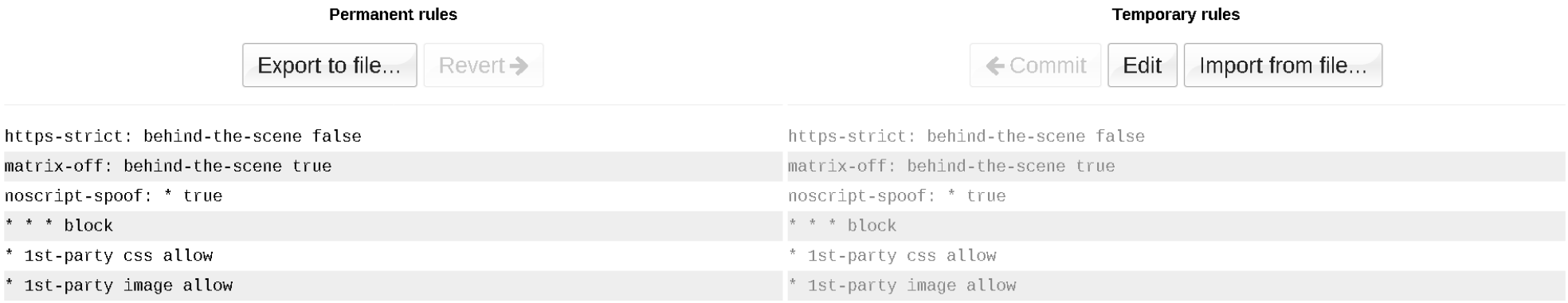
Install uMatrix (FF / Chrome) or ηMatrix (Pale Moon). An icon like this:  should appear in the top right corner of the browser. Go to, for example, https://raypeat.com (but any site works), then click the icon, and you'll be greeted with this view:



The grid tells you which requests your browser has loaded (the ones with a green background will be allowed, the red ones will be blocked). The default settings of uMatrix load all first-party stuff by default (including scripts, where most of the malice resides), as well as some third party. In this case, 2 CSS files from Google have been loaded, as well as 3 scripts from Ray Peat itself, that are not needed to read the site at all. To change the situation, click the extension title (e.g *"eMatrix 5.0.0"*). You will enter the settings menu. Click the *"My rules"* tab, then *"Edit"*. Select everything like this:



And with all the rules selected, press the *Delete* key on your keyboard. Click *"Save"*, then *"Import from file"* and import these rules. *"Save"* and *"Commit"*. The final view should look like this:



What you just did is globally deny sites the ability to load anything except first-party CSS and images, which are unlikely to negatively affect you. I call it the *mild mode*. However, by removing the last two rules *"* 1st-party css allow"* and *"* 1st-party image allow"* you can deny everything by default (*hardcore mode*). In this guide we will stick to the mild mode, though, as the hardcore mode will require you to enable images and CSS on almost every site. **NOTE**: the *"matrix-off: behind-the-scene true"* rule is essential in ηMatrix, or Pale Moon's downloading of images won't work.

The whole point of uMatrix is to deny resources by default and allow them as needed to enable the website functionality you want. With the current setup, you deny almost everything that's possibly undesirable. And yet, most sites will still work perfectly fine. What if you find one that doesn't? Consider Euractiv. Though it loads 15 first-party CSS files, it still looks clearly broken:

Experts: US delays F-16 fighter jets over trust issues with Bulgaria

- Editions
- New
- Ever
- Mark
- Futu
- Search
- Sea

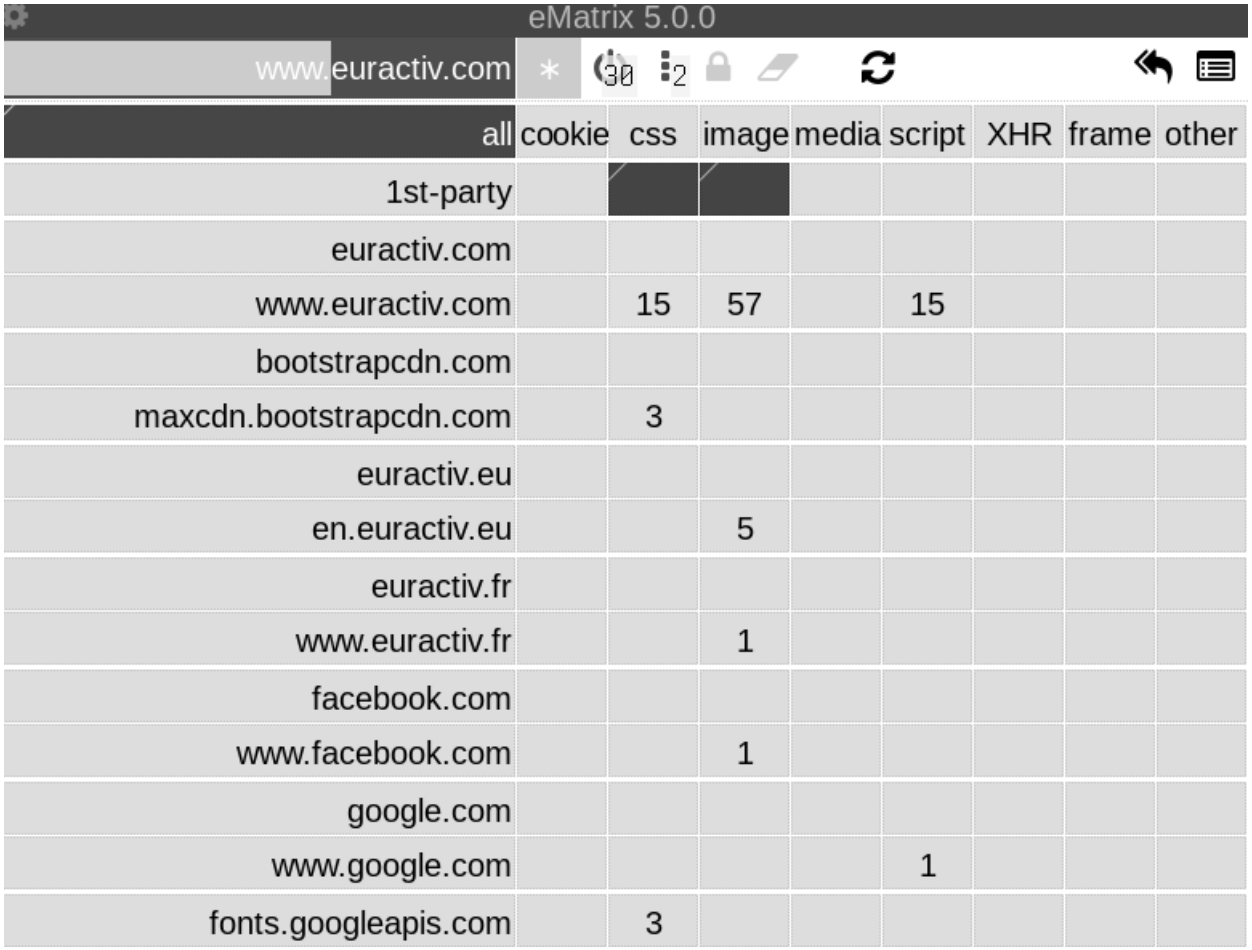
Europe
Deutschland
France
España
Italia
Polska
България
Česká republika
Ελλάς
Hrvatska
România
Србија
Slovensko

Sections EURACTIV

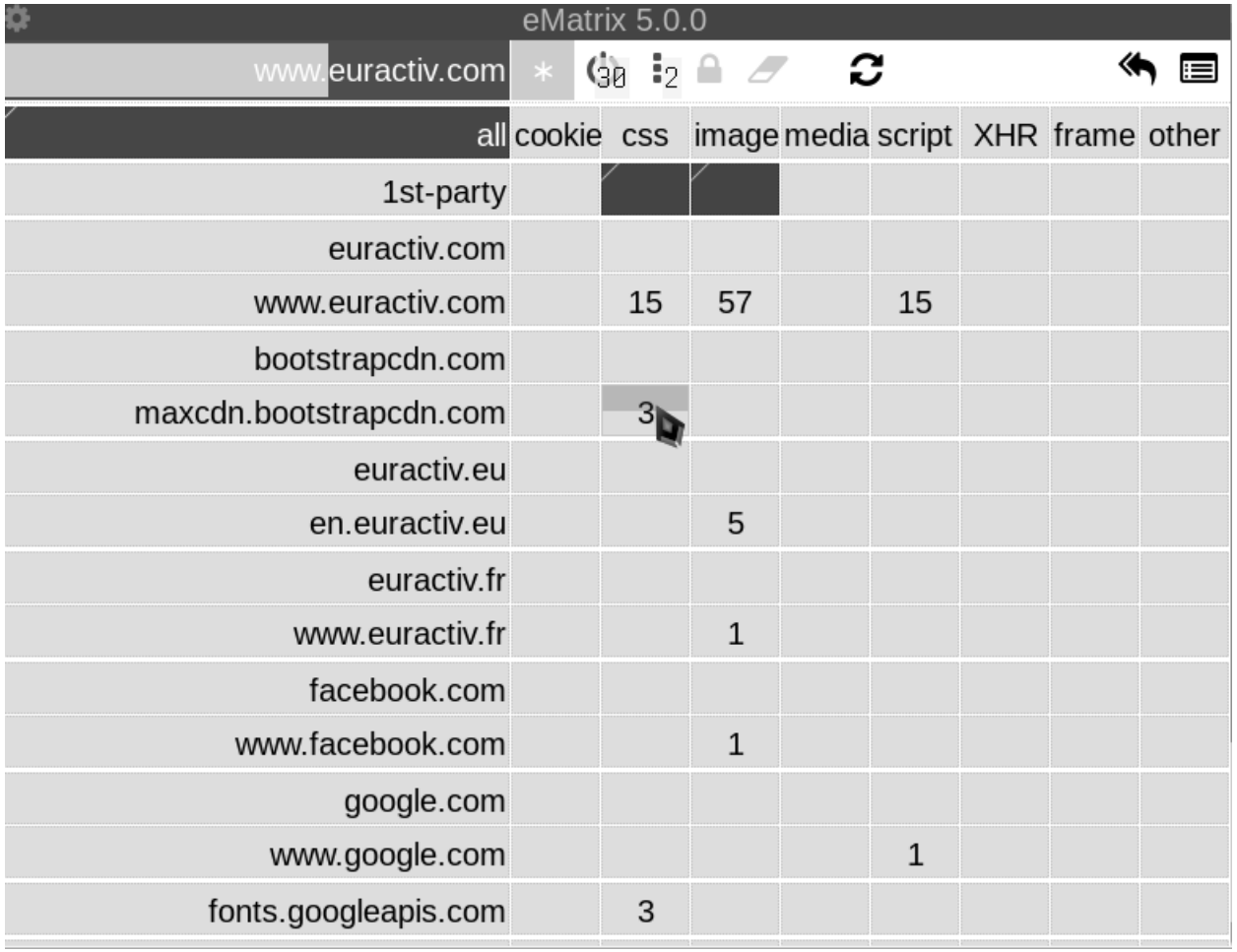
- The Capitals
- The Capitals
- The Brief

x

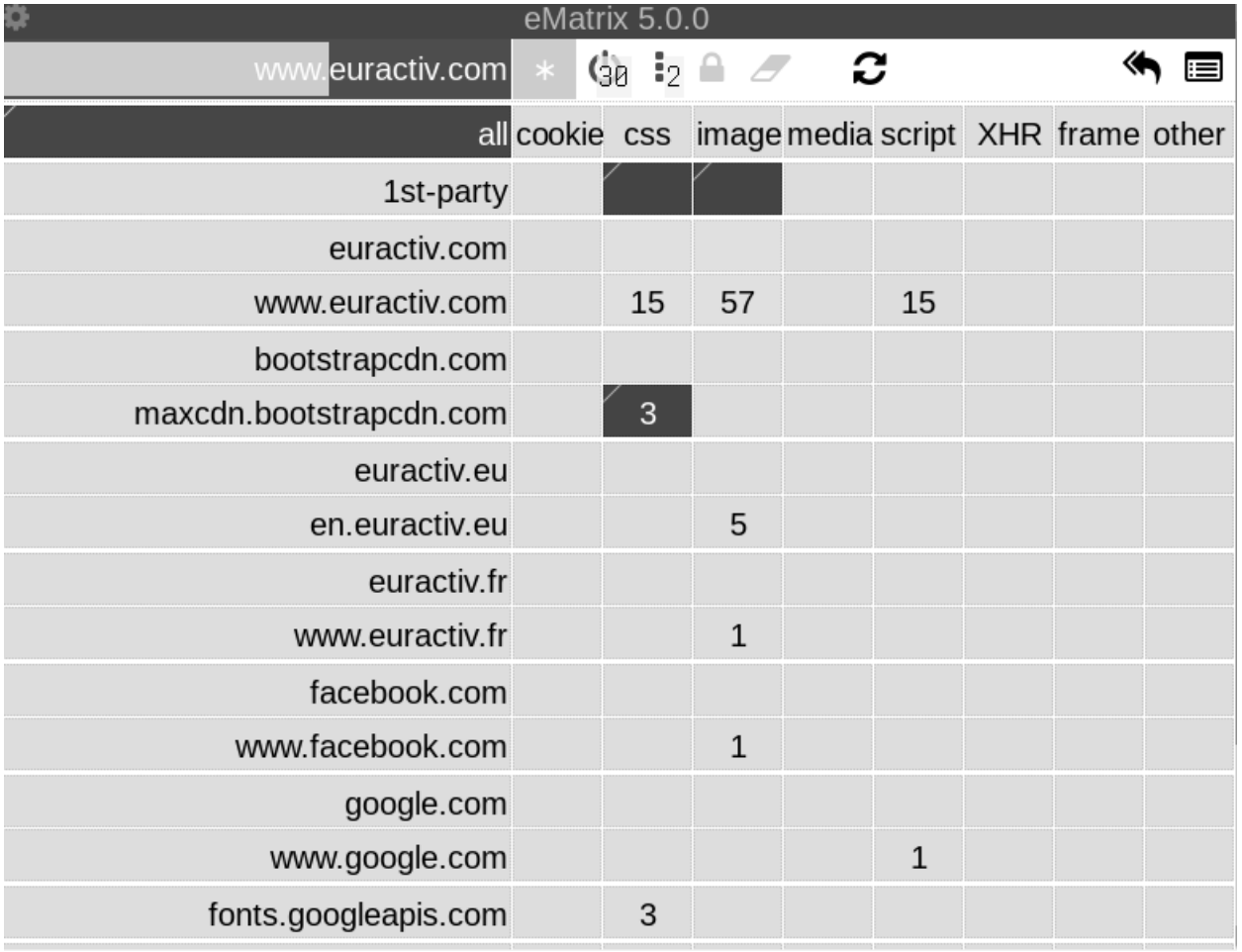
The uMatrix grid of Euractiv looks like this:



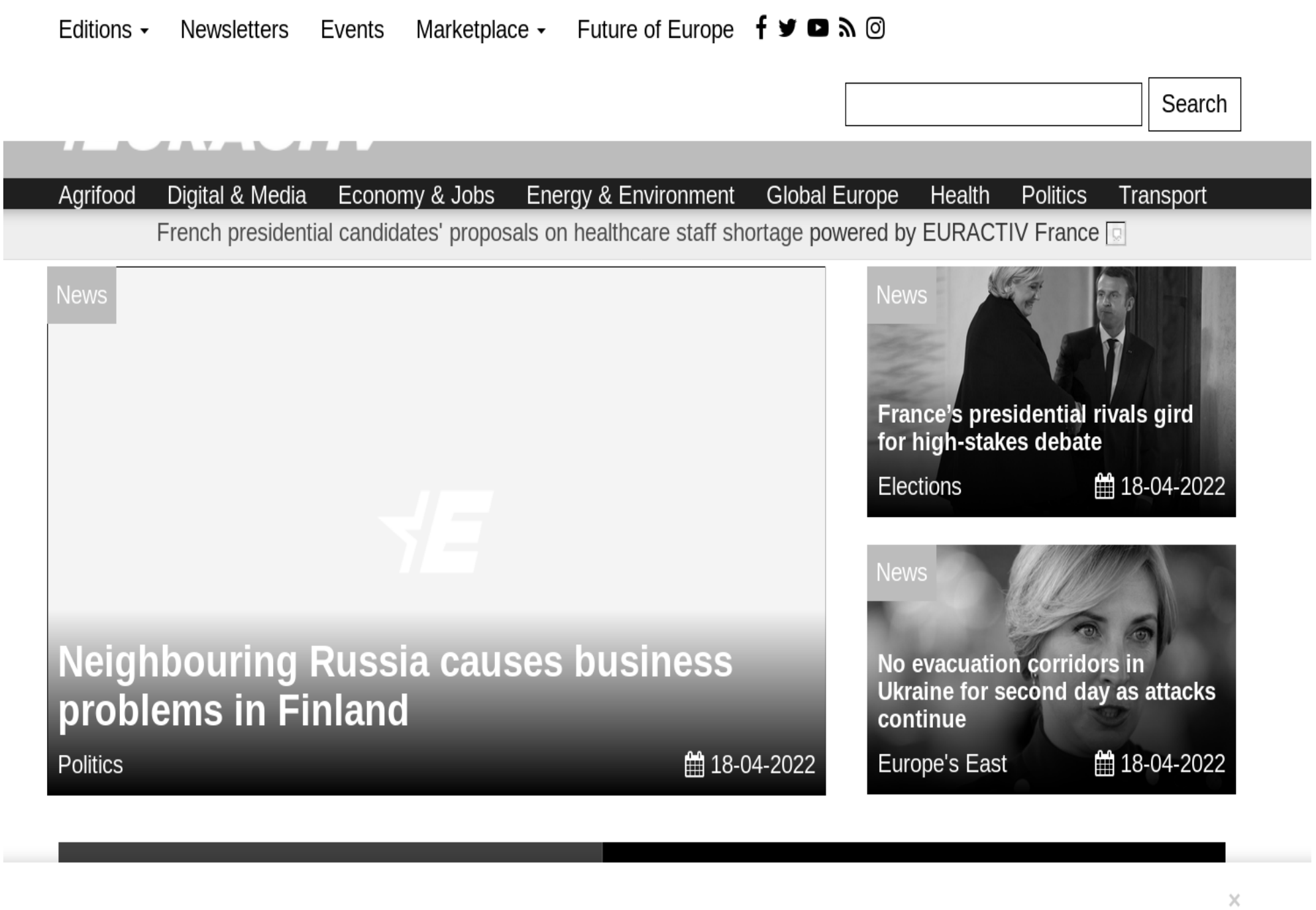
There are more requests down there, too (and you won't see all of them until you allow scripts, since scripts can load their own requests - making the real situation much worse than seen in the above picture). But we are only interested in the *"maxcdn.bootstrapcdn.com"* CSS one (actually 3). Click the top half of the tile with the 3 CSS files, like this:



This is how the Euractiv grid should look like after:



Remember - upper half is enabling, bottom half is disabling. Red is disabled, green is enabled. **Only the requests with green tiles get loaded**. Oh, and you might be wondering why the bootstrap CSS you've just allowed is a darker shade of green. The darker shade appears for the tiles you have specifically allowed (such as the bootstrap CSS). The 15 first-party CSS files and the 57 images are all allowed because of inherited global rules - which is why they are light. The same applies to the red shades. **Dark is local, light is global**. Functionally, it doesn't matter - **all green is allowed, all red is banned**. The only difference is in the information provided to the person sitting in front of the screen. Anyway, here is how the fixed site looks like:



Much better ^_^. If you are satisfied with the results, click the padlock to make them permanent

for the site you're on. And now, you've learned how to unbreak sites in uMatrix. Wasn't that hard, was it? With only one domain enabled over the default settings, you were able to make a site look properly and still block all undesirable stuff. What would a "minimal" browser be able to do (or any without uMatrix installed)? Either allow everything - including the 15 useless scripts (it is really more, since as I've stated above, they don't all show up unless the others - that are now blocked by uMatrix - are allowed to load them) - or deal with a broken website. Anyway, there is no magic rule that determines which requests need to be allowed to fix a site. That knowledge comes from experience. It just so happens that the bootstrap CSS is a pretty common third party resource that is also necessary. But sometimes you will need to allow scripts, and even refresh several times so that new requests get loaded, that you will then also need to allow.

However, **it is mythology that uMatrix is a chore to use**. Again, most sites are readable out of the box, so you don't need to do anything at all - but you still enjoy the protection from all the undesirable stuff. If a site displays badly, you most often only need to allow one or two domains with CSS, images, or a cookie to keep logins. This will become second nature while you become experienced with uMatrix. Sometimes more tinkering is necessary (especially to make interactive stuff like searches work), but this is rare - and you can permanently save rules, anyway. So, the next time you come back to a site, it will work the way it did when you left it. **Using uMatrix gets more effortless the longer you do it** - so much that, after you "cover" your most common sites, you almost don't notice it. All in all, uMatrix is **the** way to gain almost full control of what requests your browser is sending to the websites you visit - with only a little effort (and much less than writing your own adblocker lists would take). By the way, you **do not want to use an adblocker**, since they cannot do even close to what is necessary to mitigate the Web. Why that is, I have explained more deeply here.

Let's explore the other settings in the menu. You can *"Collapse placeholder of blocked elements"* - this will prevent showing a big ugly square for e.g a youtube video that didn't load. Probably should *"Spoof <noscript> tags when 1st-party scripts are blocked"*, since otherwise, every website that uses those tags will assume you're running scripts and fail to display properly. *"Spoof HTTP referrer string of third-party requests."* can be turned off and replaced by your browser's settings (e.g *"Spoof referer to target URL"* in Pale Moon's *"Advanced Preferences"* - this needs *Pale Moon Commander* addon), which will also work for first-party referers, not just third party (these can also track you). Turn on *"Block all hyperlink auditing attempts."*, since the only point of those is privacy invasion. Also make sure to enable *"Resolve CNAME records"*, since some trackers are now pretending to be first-party to avoid extensions like uMatrix. By default, uMatrix **lets websites store cookies on your disk** - even if they're blocked. This still prevents the cookie-based tracking, because they're **not sent to the website** (if blocked); but - if you don't want them sticking around - you can enable the *"Delete blocked cookies"* option. You can go to the *"Hosts files"* tab and disable them all. With the grid set up according to this guide, they are unnecessary and just bring additional load. In the *"My rules"* tab, you can export your uMatrix settings in order to import them to another device later. This prevents having to redo your website fixes.

You might be wondering why I even wrote this guide. The addon is old and surely it's been covered over and over, right? Well, as usual, the other guides do not satisfy my standards. This one (archive), for example, focuses on some other stuff instead of the beautiful grid. It takes them until the end of the page to mention the stuff that actually matters. And this is a very old version of the addon, which does not even allow entering the crucial global mode. Another guide (archive) just talks and talks, and also tries to pull people away from the essential global mode. By doing so, they glorify allowing all the trash to load by default - they pretty much

admit it later *“The good news is that, as a beginner, you can ignore all the settings positioned to the right of "all.””*. **It is pointless to use uMatrix in anything other than mild or hardcore mode** - such as allowing all scripts or images by default. What makes uMatrix so powerful is being able to situationally decide which categories of requests get loaded, and when. Allowing whole categories by default (like the other guide recommends) destroys this advantage, and makes uMatrix work more like a worse uBlock Origin. That site also requires JS and XHR just to see images, ironically making it a good boot camp for uMatrix usage. It is also Cloudflared.

Mitigating Cloudflare

Just block the Cloudflare IP addresses (archive) on the firewall, router, etc. Here is a way to do it with iptables:

```
sudo iptables --table raw --append OUTPUT --destination 173.245.48.0/20 --jump DROP
sudo iptables --table raw --append OUTPUT --destination 103.21.244.0/22 --jump DROP
sudo iptables --table raw --append OUTPUT --destination 103.22.200.0/22 --jump DROP
sudo iptables --table raw --append OUTPUT --destination 103.31.4.0/22 --jump DROP
sudo iptables --table raw --append OUTPUT --destination 141.101.64.0/18 --jump DROP
sudo iptables --table raw --append OUTPUT --destination 108.162.192.0/18 --jump DROP
sudo iptables --table raw --append OUTPUT --destination 190.93.240.0/20 --jump DROP
sudo iptables --table raw --append OUTPUT --destination 188.114.96.0/20 --jump DROP
sudo iptables --table raw --append OUTPUT --destination 197.234.240.0/22 --jump DROP
sudo iptables --table raw --append OUTPUT --destination 198.41.128.0/17 --jump DROP
```

sudo iptables --table raw --append OUTPUT --destination 162.158.0.0/15 --jump DROP
sudo iptables --table raw --append OUTPUT --destination 104.16.0.0/13 --jump DROP
sudo iptables --table raw --append OUTPUT --destination 104.24.0.0/14 --jump DROP
sudo iptables --table raw --append OUTPUT --destination 131.0.72.0/22 --jump DROP
sudo iptables --table raw --append OUTPUT --destination 104.16.0.0/12 --jump DROP
sudo iptables --table raw --append OUTPUT --destination 172.64.0.0/13 --jump DROP

And for IPv6:

sudo ip6tables --table raw --append OUTPUT --destination 2400:cb00::/32 --jump DROP
sudo ip6tables --table raw --append OUTPUT --destination 2606:4700::/32 --jump DROP
sudo ip6tables --table raw --append OUTPUT --destination 2405:b500::/32 --jump DROP
sudo ip6tables --table raw --append OUTPUT --destination 2405:8100::/32 --jump DROP
sudo ip6tables --table raw --append OUTPUT --destination 2a06:98c0::/29 --jump DROP
sudo ip6tables --table raw --append OUTPUT --destination 2c0f:f248::/32 --jump DROP

After that, save with *“sudo iptables-save > /etc/iptables/block-cloudflare.conf”* and *“sudo ip6tables-save > /etc/iptables/block-cloudflare-ipv6.conf”*. And load the config files on startup by putting *“iptables-restore /etc/iptables/block-cloudflare.conf&”* and *“ip6tables-restore /etc/iptables/block-cloudflare-ipv6.conf&”* in */etc/rc.d/rc.local*. After applying the changes, you will receive a "connection timed out" message when trying to visit a CF site. Another way to block CF is to distrust their certificates, but **that method does not work for many CF sites**, so I won't go into the details. The only possible problem with the iptables method is that CF might be hiding some IPs, but I doubt that really happens (however, they might be adding new ones every so often, so keep checking for their current lists). Effectively, **it is the perfect way to block CF**.

To check whether a site is behind Cloudflare, type the command *“dig [website_address.com]”* into the terminal (as in *“dig naturalnews.com”*). A part of the reply will look like this:

; ANSWER SECTION: naturalnews.com. 300 IN A 104.16.135.70

Now type the command *“whois 104.16.135.70”*. If the result contains things such as *“Cloudflare, Inc.”* anywhere - then the site is behind it. If it doesn't, it's not.

If you really want to visit Cloudflared sites, a few ways exist to make it somewhat safer:

- The Wayback Machine. This depends on the website having been archived in the first place. If it hasn't, then you can try to do it on your own, but many site admins disable this option. And, interactivity (sending forms or other server-side stuff) won't be possible with this method, either way.
- Ghost Archive. A simpler version of the above. **UPDATE June 2024**: now Cloudflared itself. Too bad.
- Morty Proxy. This visits the website as it exists right now, so it doesn't require additional user effort, and sites can't be blocked from being accessed, unless the instance IP is blocked. However, JavaScript isn't loaded so much website functionality won't be possible.
- Third Party Request Blocker extension can automatically redirect websites to their Web Archive versions, if they exist. What *Block Cloudflare MitM Attack* used to do. The downside is that it only works in Firefox.
- Using a separate browser - such as TOR Browser or TORified Chromium for Cloudflared sites could also be considered a mitigation, in that it creates a new identity to absorb the tracking. This is the only way that allows full interactivity with the target

> site, but it comes at a cost. Namely that you're the party that's being MitMed, instead of Web Archive or someone else. And you have to obey the UA and other requirements of CF.

All those ways suffer from the same critical flaw though. Someone out there still has to submit to the evil, and take the MitM up the ass as well as endure the browser and other restrictions CF encumbers upon their victims. Dealing with CF is like dealing with an abusive husband that you're dependent on financially. You might send an intermediary (as with all the mitigations except the last one) or try to hide your activities somehow, but in the end, you're still submitting to him. So, stop submitting and block CF system-wide :D.

A better way?

The current situation cannot go on forever. Sites will be getting more complicated and more malicious - and eventually, it will be impossible to mitigate them. This happens mainly because of two diseases: **soydevism** and **capitalism** (though many other minor reasons exist). Soydevism happens when a webdev didn't learn the basics (HTML, CSS, etc) properly and instead relied on frameworks to create their website. Some programmers have said that these days, a junior dev doesn't even learn the basics, but only the high-level stuff. Soydevs are also known for making their websites dumping grounds for trash like reCaptcha, third party fonts, social media buttons, etc (probably because they don't know any better). Capitalism happens when people stop making sites for enjoyment or passion and instead focus primarily on money (this will necessarily happen in a world where money decides your level of power and is also required just to live). That is when people figure out they can throw in a bunch of ads, tracking scripts, sponsorships etc. to make their site profitable. It also explains filling their websites with 20

clickbait articles per day (more impressions for ads and data collected by scripts). Ars Technica is a great example of a site that combines soydevism and capitalism. Of course, big corporations have contributed to the problem, by creating increasingly complicated standards that only care about working in Chrome (and then people - like the Mullvad team - stop supporting any other browsers). The more these behaviors are normalized, the more of a junkyard the Web ends up being.

The better way is to stop supporting the bad practices. Move away from visiting places that hate you towards those that respect you. This means websites with no ads, exploitative captchas, analytics, or CDNs. Sites that work in all browsers, including 20 years old ones - instead of only the big corpo abominations. Sites that - even if they decide to include modern functionality - practice graceful degradation when doing so, so that they still work properly in older or mitigated browsers. Sites that put their hearts and souls into everything that's on them, instead of dumping it there just to have it. Then, if you have your own site, **do link** to the other good ones. Though uMatrix, *I don't care about cookies*, *URL Rewriter*, BCMA, etc. are great tools, I really wish we could realize that they are **mostly bandages on a Web that is broken at the fundamental level** - then we can decide to just give them up. Let's create an internet full of love by default :D.

Back to the front page