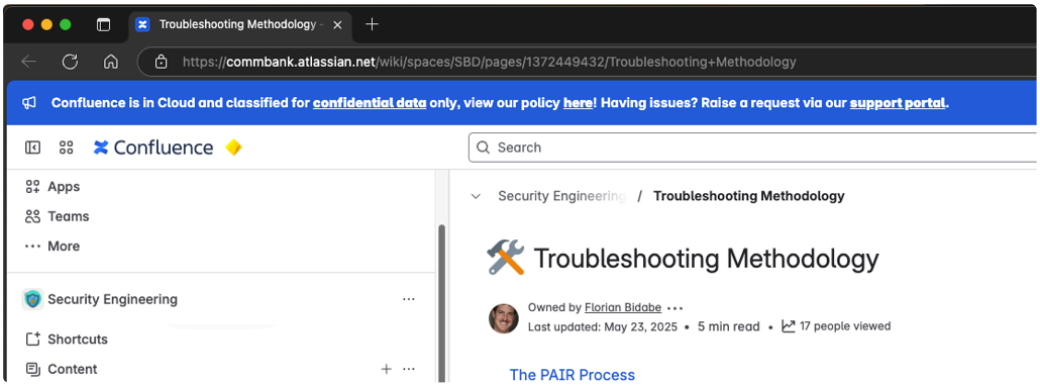
🙊 Block pesky HTML elements on a browser

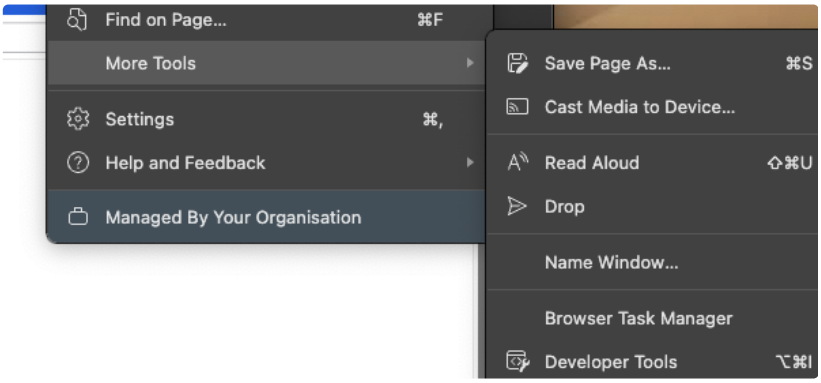
Say some elements are eating up some space on small display (e.g. laptop screen while on the go...)
and you really could make use of all that space for editing or reading content...
This guide shows you own



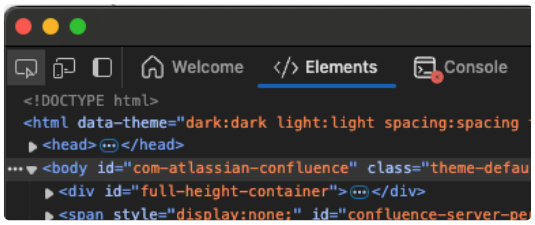
Like this blue banner which you wish could be acknowledged and closed... but unfortunately, you cannot close it!

Temporary Solution using DevTools

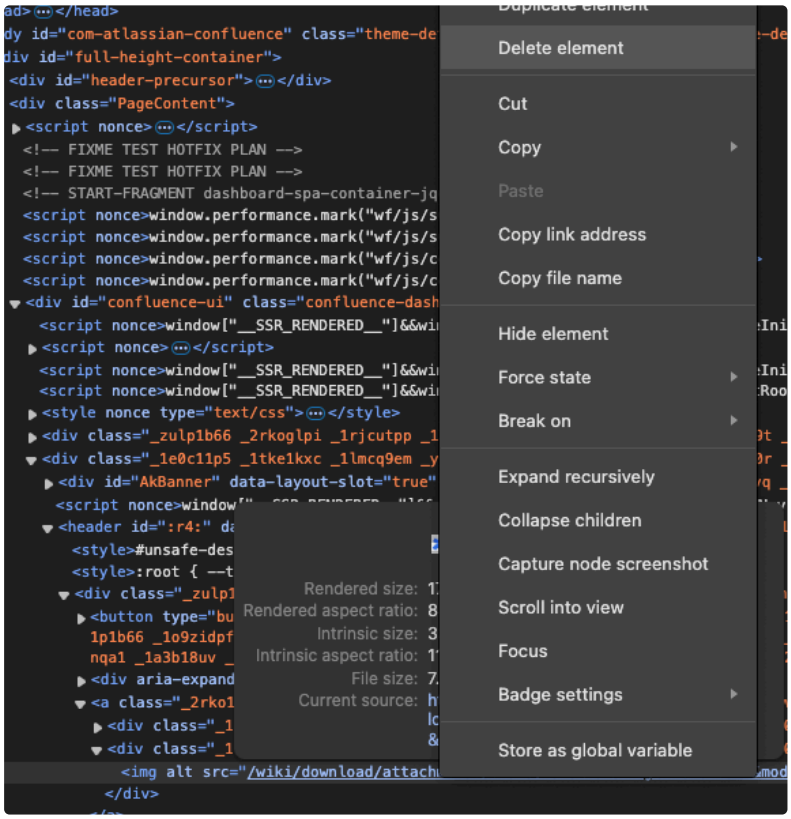
There is a way to optimise what's displayed on your page...
The first way is to hide or delete the element using the DevTools, for that you'd open the DevTools, Select the element and...



The DevTools



notice, the top left icon, this is how you identify the element to hide / delete

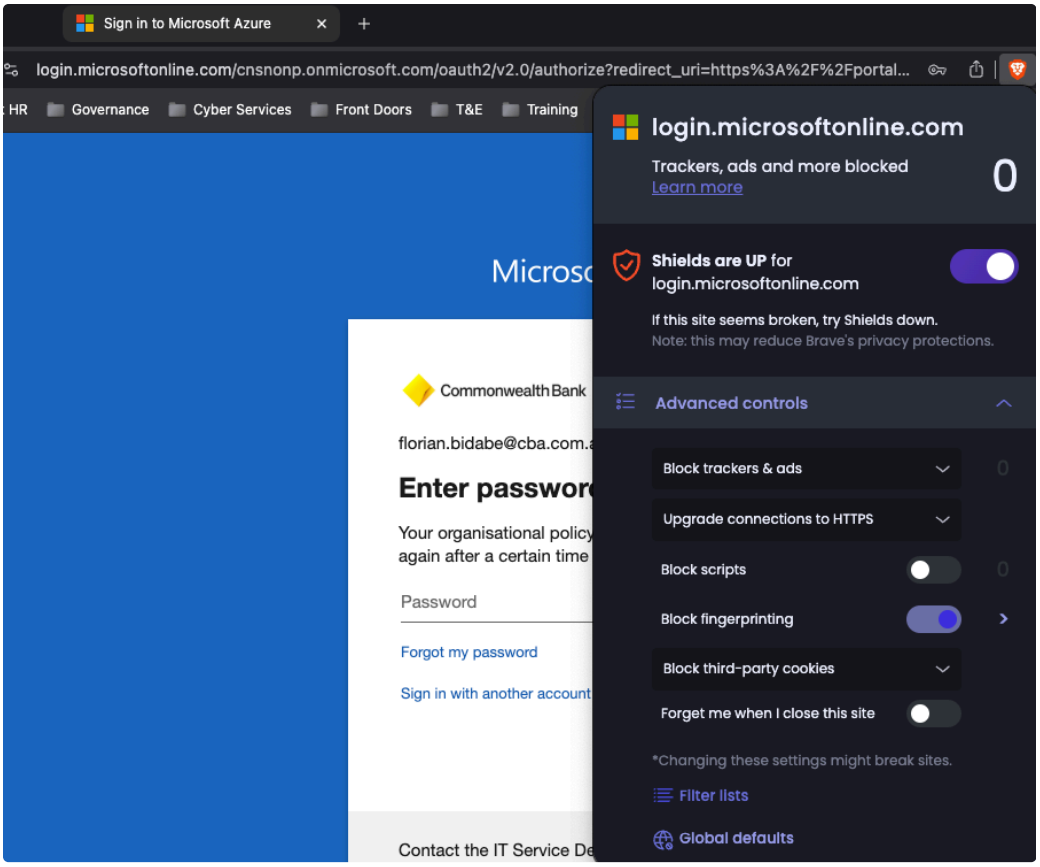


delete element

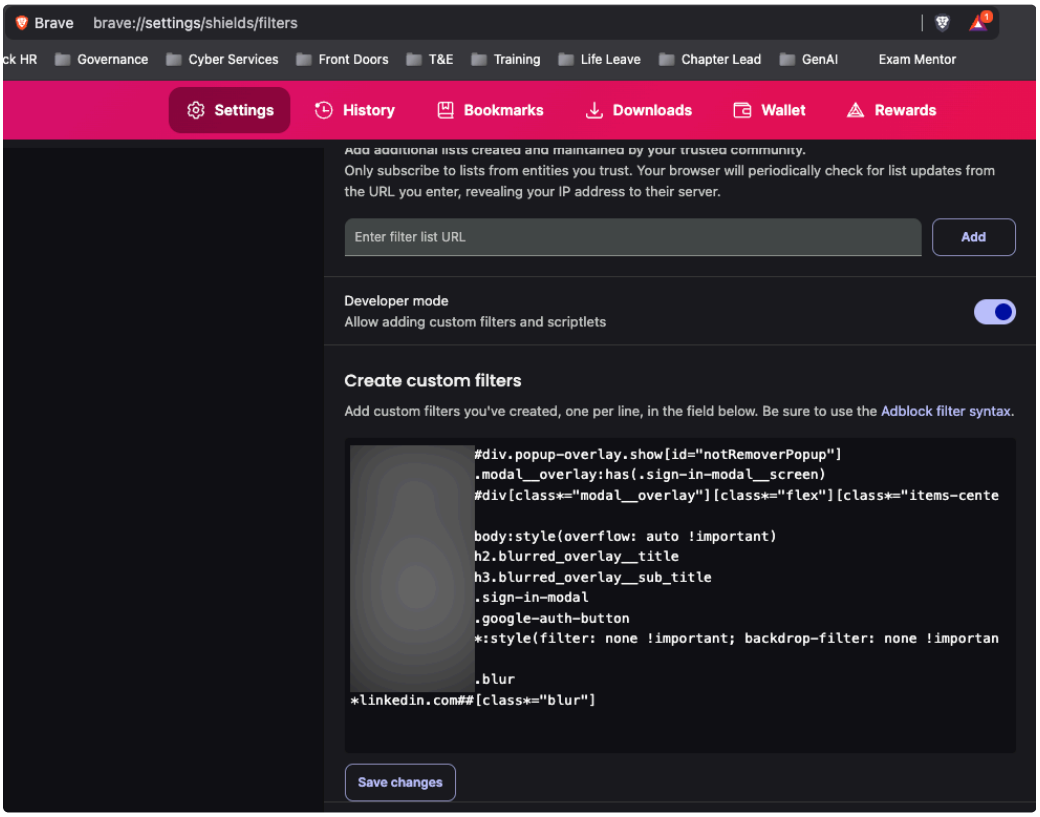
⚠️ << But the element reloads every time I reload the page 😵 ! >>
Yes, it does indeed. If we want this to be permanent, we need to use AdBlocker rule syntax to delete the element everytime it loads...

Permanent Solution using Brave

And that brings us:
Using a browser like Brave which allows to do that out of the box



Select Filter list or brave://settings/shields/filters



And block the pesky elements

⚠ << Hold on... I don't want to use a unmanaged browser. How about Edge, Chrome or Firefox... >>
See below...

Permanent Solution using a managed browser

DISCLAIMER: EDUCATIONAL PURPOSE ONLY

This article is provided strictly for educational purposes. Modifying system configurations or tampering with corporate security measures without explicit permission is prohibited and may violate organizational policies, employment agreements, and applicable laws.

This content demonstrates how corporate network defenses might be circumvented, specifically to educate security professionals about potential vulnerabilities. Organizations should implement a defense-in-depth approach with multiple layers of security to protect traffic sent and received from endpoints, including protection against threats from internal users (staff/insiders).

Always obtain proper authorization before implementing any techniques described in this article.

The challenge with managed browsers, is that we've disabled extensions... this is valid from a cyber standpoint since some extensions could leak sensitive data such as browser activity and maybe even browsing content / and POST requests (containing your bearer tokens and auth cookies)

So by default, you will not be able to install extensions unless, if you add **one** to the exclusion list...

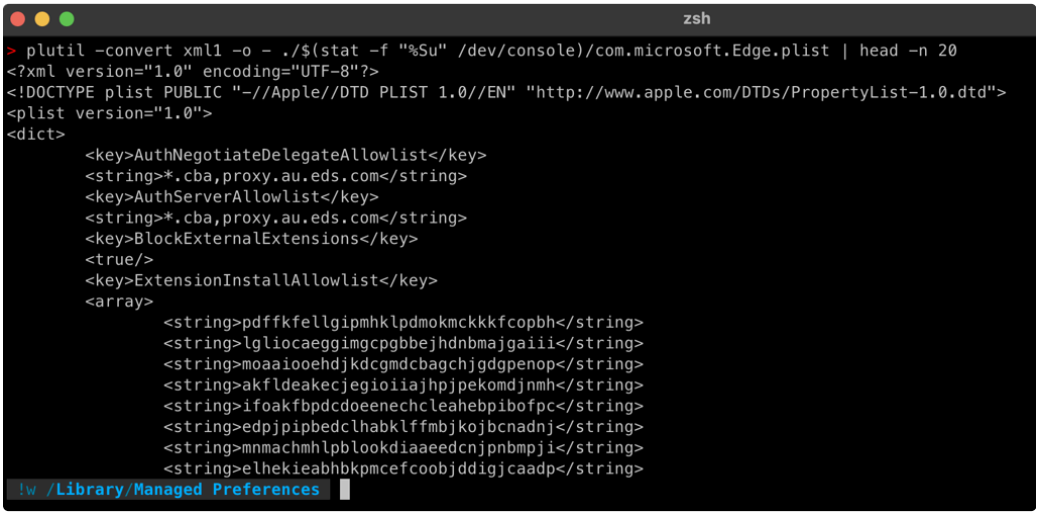
And that is AdGuard for instance (which I have tested and inspected traffic from using Charles, and it does not leak data).

ℹ Note another really handy extension could be FoxyProxy if you;re troubleshooting proxy issues, where you set a HTTP proxy which is not inherited from the system! (Just like in Firefox really)

We'll use Edge as an example:

⚠ Again, this is for educational purpose and I would re-iterate you should not perform these changes, without due approval from your management and cyber.

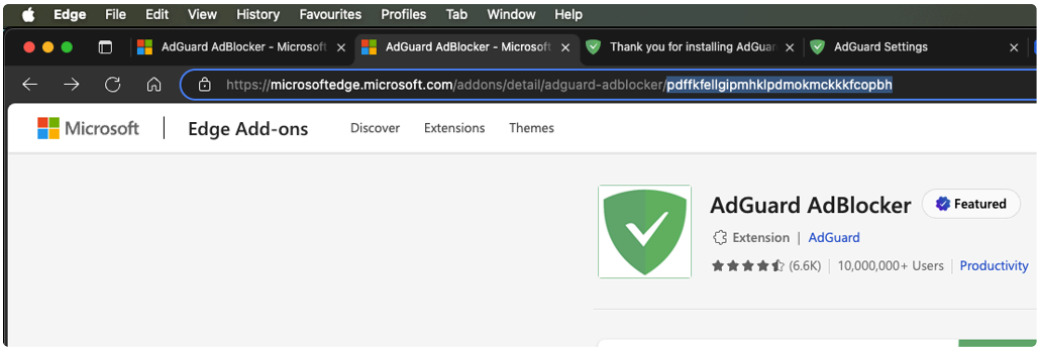
In the same fashion than https://commbank.atlassian.net/wiki/spaces/SBD/pages/1362577886/Local+MITM+capture+all+traffic+from+the+SOE?atlOrigin=eyJpIjoiZTYzYzQ2NWRlZDMwNGIyMTk3ZDQ5Njg3MzFlZGYzMmUiLCJwIjoiYyJ9 Can't find link , you'd need to edit `com.microsoft.Edge.plist`



As you can see, we have a list of allowed extensions...

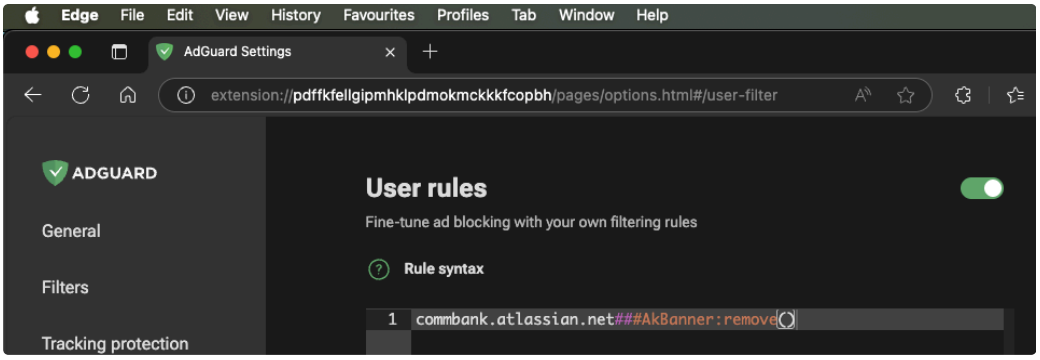
By adding the extension ID, you can now install it...

Altho you'd need to reload the managed profiles by rebooting or by `sudo killall cfprefsd` killing the process that reads then, so it can reload...



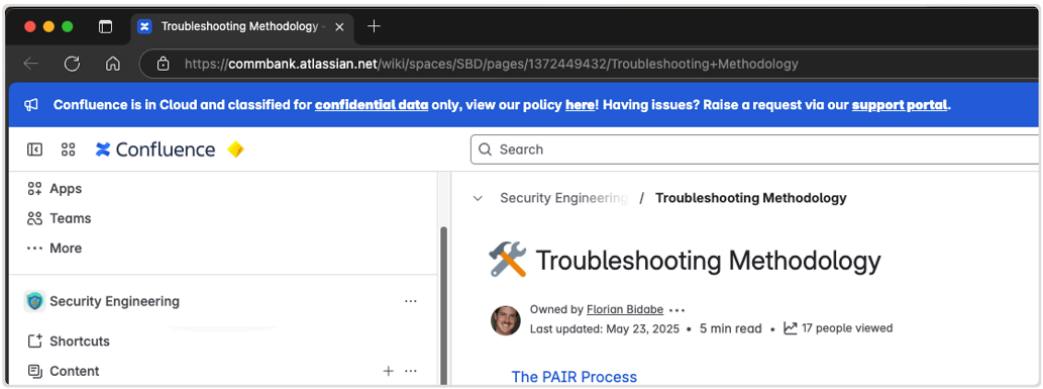
AdGuard is now installed

That's it... no more pesky Ads, and no more banner eating up your tiny display space without working on your public commute !

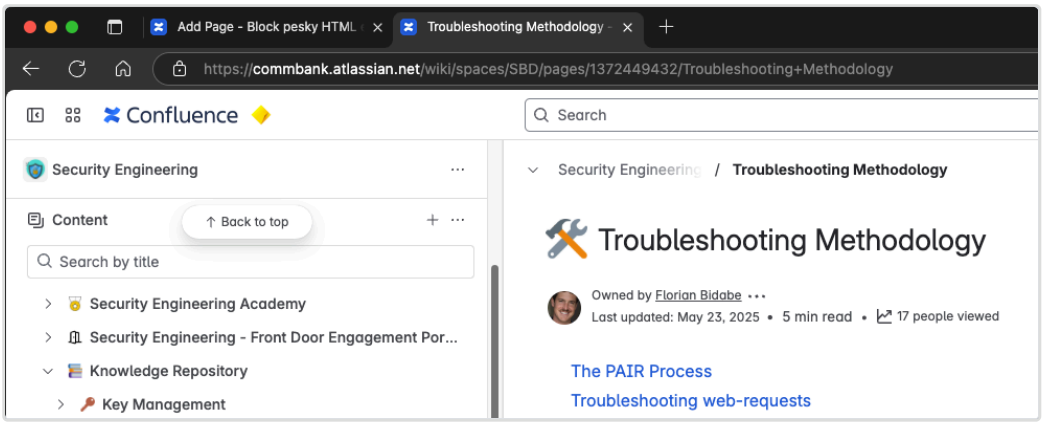


The Blocking rule... do not ommit `:remove()` or the rule will simply hide the element and replace it by a white space, so it wouldn't free up display space...

Outcome



BEFORE



AFTER