Project repository:

In this repository you will find three folders:

1. cookiePilot – the browser extension I wrote that exposes stored browser cookies to the user.
2. websiteVisitor – another browser extension I wrote that will visit a stored list of websites automatically
3. DataCollectionAndResearch – this is where all the data I collected is stored as well as the code for the parsing of it.

Additionally, my project presentation link can be found here. Below is a write up for each part of my project.

CookiePilot:

CookiePilot is the name that I gave my browser extension. Its main function is to expose stored browser cookies to the user without having to go into developer tools to see them. Additionally, it also shows all cookies stored in the browser unlike the developer tools which show just the ones stored on the site. It has a couple of additional features including the ability to clear all cookies from the extension. This prevents users from having to go into the browser settings to do such. Unfortunately, it does not avoid removing useful cookies like login cookies so it must be used with caution. The extension also has a download cookies button. This button downloads the contents of the All Saved Cookies table to a csv which can then be used for analysis. For more detailed analysis, there is also a function "downloadAllCookies" that can be called from the developer's console. This will download all the cookie information, as opposed to the reduced cookie fields shown in the table, for more detailed analysis.

Website Visitor:

This is another browser extensions that I wrote that traverses through a list of websites stored in the extensions root directory. It sequentially visits each website in the list, waiting for the page to completely load before going to the next one. This extension works reliably about half of the time but there are some intermittent issues with it loading the next one without the previous site being entirely loaded.

Data Collection:

I decided to investigate cookies related to the most popular websites, hoping that this would closely emulate user behavior, to see what effects browser options had on cookies stored throughout web browsing. I initially endeavored to use a list of popular websites generated by CrUX, the chrome user experience survey. This was inspired by this research paper which suggested this was the most accurate list; however, the popularity of websites is only grouped by orders of magnitude down to the thousands. This means that there is only a list of the 1000^{th} most popular websites available but not in ranked order. Additionally, there was not a way to verify the security/safety of this extremely large amount of links. Therefore, a list of the 50 most popular websites in November 2023 was used from Similarweb. A downside of this ranking was that it only provides a list of top level domains. This means that only Amazon.com, for example, would be visited but not a specific product page. This doesn't accurately represent user behavior and likely decreased the number of cookies stored but it will

hopefully at least show trends across the different browser options. Additionally, it likely decreased the number of more privacy invasive cookies, like tracking cookies, that were stored. The list of the specific domains visited can be found in the domainsToVisit excel file. Of note, the domain “MicrosoftOnline” was omitted because it didn’t load and the domains from any adult content sites were removed from the list.

7 different data collections were performed on these domains using the Firefox browser. The first two runs used the two Firefox default options – custom and strict. The last five runs used the custom settings, each time with all options turned off and the cookies option set differently – shown in Fig. 1. These runs were run in the order of the options download. All cookie data was downloaded immediately after the run. All the runs were conducted using a separate Firefox browser profile, with all cookies and browser history cleared between each run. Ideally, more runs across different browser settings and browsers would have occurred but this was infeasible due to the time spent creating browser extensions.

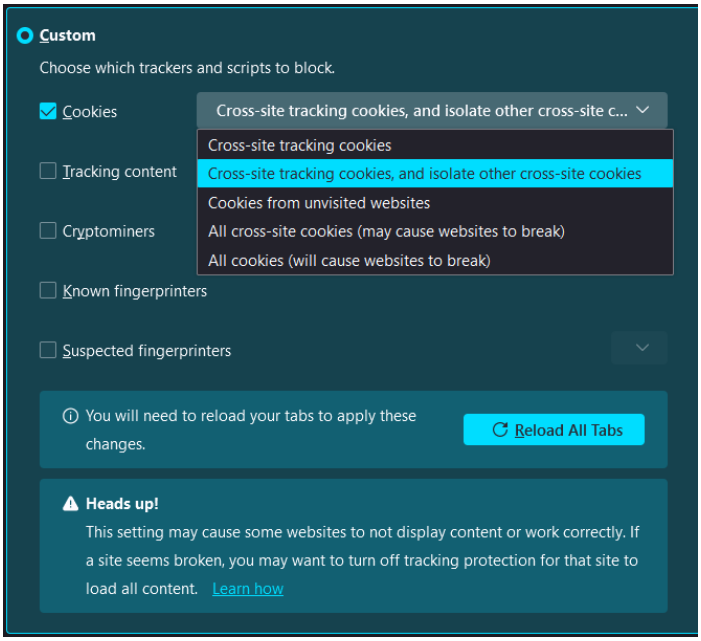

Fig. 1. Custom Firefox cookie options

Data Insights

In total, across the seven runs, 4096 cookies were collected. Each cookie had nine different fields that were downloaded from the cookiePilot browser extension: name, value, domain, hostOnly, path, secure, httpOnly, sameSite, and session. Session relates the browser session and was removed from analysis.

Of the remaining features, hostOnly, httpOnly, and secure are Boolean values. Therefore, the percentage of cookies with these flags set as true was recorded. Additionally, the percentage of domains classified as “First Party” – originating from the visited domain and the percentage of cookies with the sameSite feature set as "no_restriction" was recorded. The percentage of first party cookies was calculated by matching the domains of all the collected cookies with the list of visited domains. Three other features were calculated from the dataset: total number of cookies, total length, and the number

of unique domains. The total length of the cookies was calculated by summing the value flag of each cookie in every data set. The number of unique domains was calculated by the number of unique entries in the domain flag of each cookie. Table 1 shows all the calculated data for each of the 7 runs.

Table 1. Collected Cookie Data

Run	Number of cookies	Total Length	Number of Unique Domains	% first party	% secure	% hostOnly	% httpOnly	% sameSite
1	740	55479	102	100	44	26	17	84
2	481	40947	90	100	49	28	25	85
3	915	62865	254	65	61	19	20	92
4	688	53787	128	96	49	26	21	87
5	612	46284	97	100	46	25	19	85
6	580	46063	97	100	47	23	21	85
7	0	0	0	0	0	0	0	0

From a privacy perspective, httpOnly doesn't tell us much about the content of the cookies. The flag, according to the mdn docs, is only there to make it inaccessible to client-side scripts. This is likely why we don't see much variation in its value. Additionally, not much variation is seen in the hostOnly %.

There are a couple of different things that stick out to me from this table. First, Firefox's two default privacy options – standard and strict – do progressively limit the number of cookies. Interestingly though, at least according to the Firefox settings, the only difference between the two settings is the stronger protection against known and suspected fingerprints with the strict setting. So, unless there were nearly 300 cookies related to browser fingerprinting, a fact that I find hard to swallow, it is likely that some other things also changed with the strict setting. Further research using the custom settings and the same cookie setting as the strict could investigate the effects of limiting those types of cookies could help determine the difference between standard and strict more rigorously.

Another interesting thing is the difference between run 3 and run 1. When compared to run three, which is just blocking cross-site tracking – the same as the standard and strict setting – but not blocking tracking content, cryptominers, known fingerprints, or suspected fingerprinters, there is a difference of almost 200 cookies between run 1. However, the number of domains visited increases twofold. The number of cookies that are linked to these activities, according to Firefox, really surprised me. Unsurprisingly, however, the setting to block all cookies actually did block all cookies.

From Table 1, it seems like the most privacy preserving option is the Strict option. It allowed the storage of the fewest number of cookies in total acrost the fewest number of unique domains. While it's percent secure and percent sameSite are not class leading not much variation is seen in these options across the board; therefore, I don't think much information can be taken from these differences.

Ivan Pudwill

12/13/2023