Understanding Regional Filter Lists: Efficacy and Impact

Christian Böttger
Institute for Internet Security &
Westphalian University of Applied Sciences

Nurullah Demir
Institute for Internet Security &
Westphalian University of Applied Sciences

Jan Hörnemann
AWARE7 GmbH

Bhupendra Acharya
CISPA

Thorsten Holz
CISPA

Norbert Pohlmann
Institute for Internet Security &
Westphalian University of Applied Sciences

Matteo Große-Kampmann
Rhine-Waal University of Applied Sciences & AWARE7 GmbH

Tobias Urban
Institute for Internet Security &
Westphalian University of Applied Sciences

Abstract

Filter lists are used by various users, tools, and researchers to identify tracking technologies on the Web. These lists are created and maintained by dedicated communities. Aside from popular blocking lists (e.g., EasyList), the communities create region-specific blocklists that account for trackers and ads that are only common in these regions. The lists aim to keep the size of a general blocklist minimal while protecting users against region-specific trackers.

In this paper, we perform a large-scale Web measurement study to understand how different region-specific filter lists (e.g., a blocklist specifically designed for French users) protect users when visiting websites. We define three privacy scenarios to understand *when* and *how* users benefit from these regional lists and what effect they have in practice. The results show that although the lists differ significantly, the number of rules they contain is unrelated to the number of blocked requests. We find that the lists' overall efficacy varies notably. Filter lists also do not meet the expectation that they increase user protection in the regions for which they were designed. Finally, we show that the majority of the rules on the lists were *not* used in our experiment and that only a fraction of the rules would provide comparable protection for users.

Keywords

user tracking, tracking protection, filter lists, web privacy, adblocker

1 Introduction

Ads are central to today's Web ecosystem, primarily serving as a revenue source for online businesses, content creators, services, and other entities. For many websites, especially those that offer free content, advertising is the primary means of monetization. This model leads to considerable security and privacy problems. One of the significant issues is the pervasive tracking that underpins targeted advertising: to maximize the effectiveness of advertising, advertisers and third-party networks often use tracking technologies such as cookies [25, 28, 54, 55] and fingerprinting [22, 33, 40].

Such security and privacy concerns lead to use of *ad blocking* and *tracking blocking* techniques. Such blockers use *filter lists* to identify and block requests to known ad servers and remove website advertising elements based on predefined rules. A filter list is a collection of rules and patterns designed to detect and block unwanted website content, such as advertising, tracking scripts, and other intrusive elements. When a user visits a website, the ad blocker compares the page's content with the filter list and dynamically blocks elements that match the specified patterns. Popular lists such as *EasyList* [19] are maintained by a community of volunteers who continually update them to keep up with evolving advertising techniques and new tracking methods. Ad blocker users can subscribe to these lists to ensure their browsing experience remains free from unwanted interruptions and privacy intrusions. *Regional filter lists* are specialized rules tailored to block ads and tracking scripts specific to certain geographic regions, languages, or cultural norms [20]. These lists address the unique advertising practices, ad networks, and tracking mechanisms prevalent worldwide that global filter lists may not comprehensively cover. Popular blockers like AdBlock [2] or uBlock [44] recommend using regional filter lists if you browse non-English websites. For instance, a regional filter list for Japan would contain rules for blocking advertising from Japanese advertising networks and content in Japan. Ad-blocking software can potentially provide more effective and localized ad blocking by including regional filter lists, ensuring that users in different regions enjoy a cleaner and more relevant browsing experience. These lists are often maintained by local communities or experts who know the regional advertising landscape.

In this paper, we analyze the effects associated with regional filter lists in different scenarios. To this end, we conduct a large-scale measurement study to understand how these lists affect users' browsing experience and privacy. More specifically, we analyze nine country-specific filter lists provided by the *EasyList* community and study three privacy scenarios to understand *when* and *how* users benefit from regional lists. Using our measurement framework, we visited over 1.8 million pages, collected over 207 million HTTP requests, and stored over 579 GB of data for analysis. We found that

Proceedings on Privacy Enhancing Technologies 2025(2), 309–325

https://doi.org/10.56553/popets-2025-0063

most of the used region-specific filter lists only block a minimal number of requests and that lists designed for other regions may outperform specialized lists even in a local setting. Furthermore, we find that lists do not meet the expectation that they perform well when users visit websites that belong to the region the list was designed for, questioning the need for localized blocking lists. Our results show that most rules (93%) in the filter lists are not used.

In summary, we make the following contributions:

- **Large-scale measurement to understand the impact of localized filter lists:** We collect regional filter lists ($n = 9$) that intersect with measurement locations worldwide. We provide a real-world measurement framework that allows researchers to analyze the impact of regional filter lists on security and privacy.
- **Impact of Regional Filter Lists on Privacy:** Our findings show that regional filter lists do not significantly enhance privacy in targeted regions and are often more effective in other regions. Combined with a standard filter list like EasyList, they provide benefits, suggesting that regional lists alone are insufficient.
- **Effectiveness of Filter List Rules:** Our analysis reveals that only 7% of the rules in filter lists are effective in identifying tracking requests, highlighting the potential for significant optimization. To aid this, we propose a master list for maintainers.

2 Background & Terminology

First, we want to introduce important terminology used throughout the paper and describe the fundamentals of filter lists.

Terminology. We use the term *site* to refer to the registrable segment of a specified domain, commonly known as "extended/effective Top Level Domain plus one" (eTLD+1) [8, 15, 37, 53]. The top-level domain (TLD) is the portion of a domain name after the last dot (e.g., example.edu, the TLD is edu). However, this structure does not always apply because many registrars allow organizations to register domains directly right under the TLD (e.g., example.ac.uk). The Public Suffix List [1] is a compilation of all suffixes under which organizations can directly register domain names, also known as extended/effective TLDs (eTLDs). The term eTLD+1 refers to an eTLD combined with the next part of the domain name. For instance, in the URL https://www.example.edu/, the eTLD+1 is example.edu (the `www.` part is *not* part of the eTLD+1), while in https://foobar.co.uk/, the eTLD+1 is foobar.co.uk. The term *page* (or *webpage*) refers to a distinct URL or, more precisely, the document (e.g., HTML or JavaScript) located at that URL.

Filter Lists. Filter lists are a standard tool to block the loading of advertisements or trackers on the Web. Over time, different lists emerged that aim to block various types of content (e.g., region-specific ads or ads that contain adult content). Tracking-protection tools (e.g., ad blockers) typically use these lists to identify trackers or other privacy-invasive objects. These tools are commonly implemented as browser extensions or, in some cases, directly embedded into the browser. Filter lists are text-based files that contain a set of rules. Usually, each line in a filter list contains a single rule and commonly used lists often contain tens of thousands of rules [19]. The syntax of the rules is very similar to standard regular expressions. These expressions are matched against a given URL and return a binary result that indicates whether the URL belongs to a tracker. For example, the rule `||example.com^` blocks all requests to the site example.com. The effectiveness of a filter list is based on the rules contained in the list. Finding and designing these rules is not straightforward and is mainly done by volunteers. It is possible that the blocking of requests that match a rule breaks a website because website content would have been delivered by the blocked request [49]. Popular lists are EasyList [19], EasyPrivacy [21], and Fanboy's Social Blocking List [26]. To improve these tools, users started to build localized filter lists that contain rules related to specific locations (e.g., refer to trackers that mainly operate in a specific country) [20]. In addition to lists that aim to protect users' privacy, other lists were created to block specific content on the Web (e.g., "nocomments" removes the comment section from websites) [6].

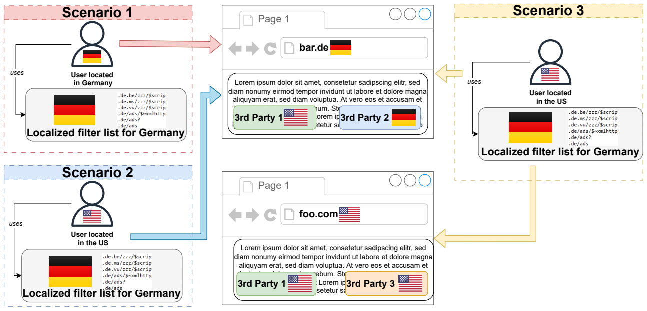
Scenario 1
User located in Germany
uses
Localized filter list for Germany
.de.be/zzz/$scrip
.de.ms/zzz/$scrip
.de.vu/zzz/$scrip
.de/ads/$~xmlhttp
.de/ads?
.de/ads
Scenario 2
User located in the US
uses
Localized filter list for Germany
Page 1
bar.de
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et
3rd Party 1
3rd Party 2
Page 1
foo.com
3rd Party 1
3rd Party 3
Scenario 3
User located in the US
uses
Localized filter list for Germany

Figure 1: Scenarios for the usage of localized filter lists. In scenario 1, a user located in, e.g., Germany uses a filter list for Germany and visits German URLs. In scenario 2, a user located in the US uses a German filter list and visits German URLs. In scenario 3, the user is in the US and uses a filter list localized for Germany but visits German and US URLs.

3 Selection and Comparison of Filter Lists

This section describes the privacy challenges using localized filter lists (Section 3.1), the selection of filter lists for our experiments (Section 3.2), and the similarities between localized lists (Section 3.3).

3.1 Privacy Challenges

In this work, we analyze the effect of different localized filter lists when browsing the Web. To assess their impact on users' privacy, we study several privacy challenges that we use throughout this paper. Our work is based on the following assumptions regarding the behavior of users and the setup they use:

- **Assumption 1**: The user uses a privacy-enhancing tool (e.g., an ad blocker) with a filter list. This assumption is reasonable, as such tools are widely used by users on the Web [5].
- **Assumption 2**: The used filter lists are region-specific. This assumption is justified, as the widely used *EasyList* [19] is primarily tailored to US users. Furthermore, different filter lists exist that target diverse user groups [6].
- **Assumption 3**: The region-specific filter lists extend the generally used EasyList for our analysis. This assumption also holds, as the non-standalone lists are intended to be combined with EasyList. To compare them reasonably, it makes sense to only look at rules on the region-specific filter lists because they are eventually combined with EasyList.

Based on these assumptions, we define three scenarios that negatively impact the users' privacy when using a localized filter list. Figure 1 provides an overview, and we describe them next:

- **Scenario 1**: In this scenario, the user visits websites whose visitors are commonly from a specific region (e.g., a user from India visits sites commonly visited by other Indian users). An example of this scenario would be a user browsing a local newspaper's website. The privacy challenge is that trackers on the page could be included in a region-specific filter list but not in a general list.
- **Scenario 2**: In this case, the user is (temporarily) in a different region (e.g., traveling) and visits websites she would usually visit. A privacy challenge is that the user gets served region-specific ads (e.g., for local goods) not covered by her filter list.
- **Scenario 3** (combination of 1 and 2): In this scenario, the user is temporarily in a different region and visits websites that are region-specific (e.g., being on vacation looking for local activities). The privacy issue is that the user might be served ads by both trackers based on the current location and trackers targeting a website's usual audience.

On the Web, targeted advertising is very common [53], and users get served personalized ads. It is essential to highlight that while ads might be user-specific, the included trackers or ad networks are not. For our work, users might see similar ads independent of the described scenarios, but the used tracker can differ.

3.2 Selection of Filter Lists

We want to analyze the privacy impact when users use different region-specific filter lists. Thus, we first collect different filter lists to assess their differences. For this analysis, we focus on 23 country-specific filter lists provided by the EasyList community [20]. Like the "original" list, the community updates these lists regularly, following a common standard. Some of the provided lists are standalone extensions of EasyList, meaning that they include most of the general EasyList rules and additionally include region-specific rules, while others only include region-specific rules. To allow comparison between the lists, we use the general EasyList (named `baseline` in the following) as the baseline and remove all rules present in the baseline from the region-specific lists. This approach also accounts for the fact that some of the analyzed filter lists are meant to be used as standalone lists (i.e., they include the baseline list), and some are meant to be used as an extension (i.e., they mainly include rules not present on the baseline list). Our approach of removing the rules from the baseline list may not completely reflect how users will use them, as they will most likely use them in combination with the baseline (i.e., standard blocklist). However, combining the baseline with each regional list will impact them equally. All lists are extended by the same rules, which means they will all block more trackers, but the ratio of blocked trackers between all lists will remain similar. Thus, in the remainder of the paper, we will only consider the revised lists that do not contain the rules from the baseline. However, to provide a picture of the effect of using the lists in combination, we provide an overview of the impact of using the baseline combined with a regional list in Section 4.4.

Before our analysis, we tested if the presented lists were regularly updated by analyzing the GitHub repositories of the 23 filter lists. The Romanian filter list did not have a GitHub repository, so we did not analyze those commits and excluded the list from our analysis. Thus, in the following, we only analyze 22 lists. To analyze the repositories, we cloned the repository to a local device and inspected the commits with the `git log` method. Using this method, we can extract the SHA1 hash, date, author, and how many rules have been added or removed by a single commit. We provide the source code in the supplementary material of our work (see Section A). The filter lists used in our experiment are updated regularly, as shown by the number of commits in Figure 2. In the period from April 2023 until April 2024, the repositories of the 23 analyzed lists received on average 3,860 (min: 15, max: 38,505, SD: 7,922) commits. The most active filter list regarding the total number of commits is the standard EasyList (i.e., the baseline), with 38,505 commits during the period of our study. Compared to the US list, European lists are less active. For example, the German list has, 1,334 commits, and the Norwegian list has 5,787 commits.

The number of commits does not provide insights into the number of added, modified, or deleted rules. Thus, to provide deeper insights into the filter list ecosystem, we analyze the change in rules by each commit. Figure 2 provides an overview of added and deleted rules across the measurement period. Figure 15 in Section E provides a fine-grained overview of the added and removed rules for each filter list. Across the analyzed lists, there is a non-binding standard to label commits. There are three main types to update a filter list: `A` for added, `D` for deleted, and `M` for modified. If a commit updates a filter list, it should start with one of those letters followed by the affected domain. Note that filter lists for Czech and Slovak, Japan, and Israel do not follow this standard; therefore, they are not included in Figure 2. It should be noted that a modification commit is usually a combination of adding and deleting rules. We filtered the commit message of the analyzed repositories and identified 223,917 (avg: 11,195, min: 1, max: 103,782, SD: 23,369) commits that added rules, 1,883 (avg: 269, min: 1 ,max: 1,817, SD: 632) commits resulting in deletion of rules, and 67,060 (avg: 3,725, min: 1, max: 43,794, SD: 9,947) commits that modified rules on the list. Thus, it seems that most commits either add rules or change them (69%), while only a minority of commits (0.4%) remove rules. We could not identify 30.5% of commits of less usage of the non-binding standard. If we assess the number of rules added or deleted by the observed commits, we see a different picture. If we look at the number of added (avg: 31,072, min: 65, max: 458,690, SD: 54,088), or removed (avg: 26,771, min: 63, max: 432,462, SD: 50,105) rules in the commits, we notice that only slightly more rules were added than removed. However, if we look at the individual lists, we see most rules were added or removed in the Vietnamese filter list (5,082,891 rules in 9,925 commits). The least active list is the general *Anti Adblock Filter list* (4,997 rules in 1,154 commits). Across all localized lists (excluding the baseline), we see that, on average, 18 people (min: 2, max: 62, SD: 15), based on their GitHub names, are committing to the repositories. Regarding the size of the individual community, we see the largest community in Scandinavia (62 people with 16,323 commits) and the smallest in Bulgarian (two authors with 146 commits). The analysis shows that most localized lists are well-maintained by an active community that updates them regularly.

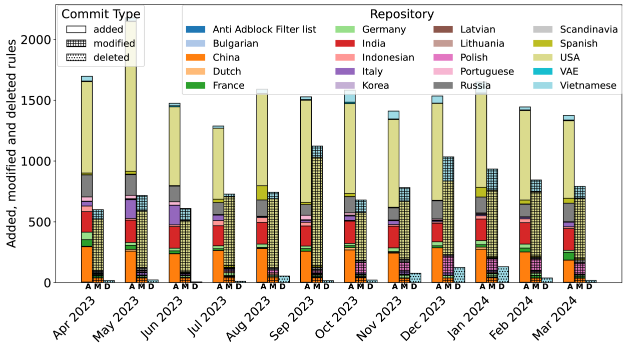

Figure 2: GitHub commits on each list's repository that added, modified, or removed rules between 04/2023 and 04/2024.

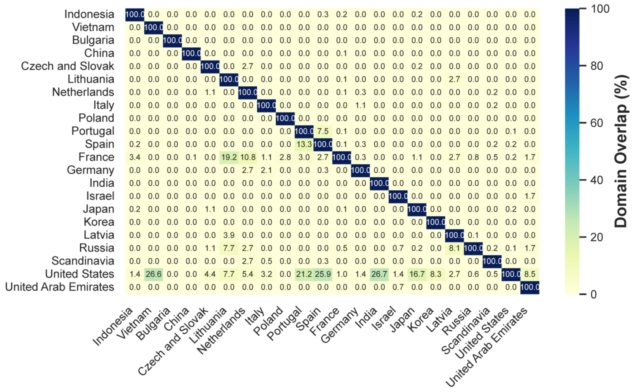

Figure 3: Overlap between the domains blocked by each list.

3.3 Differences in Localized Filter Lists

We start by analyzing the differences in localized filter lists to understand their distinct impacts on user privacy.

3.3.1 Comparing Different Filter Lists In this section, we assess to what extent the localized filter lists differ from each other and, if so, whether they protect users' privacy differently. This analysis examines whether the proposed privacy challenges (see Section 3.1) exist when a localized filter list is being used. Overall, there are 23 localized EasyLists available [20]. The lists are designed for users from 29 countries on three continents (i.e., some lists target users from different countries). We used the `General` EasyList as a baseline and removed all rules present on that list from the localized lists to compare them reasonably. On average, each localized list has 10,389 rules (min: 382, max: 74,564, SD: 17,230). In total, there are 198,989 distinct rules in all lists. Only 53 (0.04%) occur in two lists, and no rule in three or more lists. To better understand the similarity of the lists, we compared pairwise similarity between the rules. On average, two list share 1.9 rules (min: 0, max: 122, SD: 10.7), and the median of shared rules is 0. The most similar lists (Russia and France) share 122 rules. This suggests that the lists are sufficiently different to have an impact on users' privacy.

3.3.2 Understanding Filter List Similarity via Clustering Another way to understand the differences in the analyzed filter lists is to try to cluster them based on the rules present in each list. If we find clusters of filter lists, we can conclude that they are similar and, therefore, protect a user's privacy in a similar way. However, if the filter lists cannot be meaningfully clustered, this is another indicator that each list protects the users' privacy differently.

To perform the clustering, we utilize the *Jaccard distance* [36], which indicates the dissimilarity ("distance") between two given sets as a metric to measure differences between two filter lists. The metric is defined as follows: $JD(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$. By design, the index ranges from 0 to 1, where 0 denotes that the sets are equal and 1 that they have no element in common. Thus, in our case, one would mean that two lists have no element in common, and zero indicates that all rules are equal. Aside from two cases, the distance between two lists is one (i.e., no elements in common). Therefore, one can expect that clustering the lists will not be possible.

In our experiment, we use the standard *OPTICS* [3], *DBSCAN* [45], and *HDBSCAN* [9] clustering algorithms, which are all density-based, to search for a non-determined number of clusters in a given dataset. None of the used clustering algorithms could build meaningful clusters. The DBSCAN and HDBSCAN algorithms identified not a single cluster and labeled each list to be a "noise point" (i.e., outlier). In contrast, the OPTICS algorithm found a single cluster that contains all lists. These results show that the analyzed filter lists differ because they cannot be meaningfully combined or grouped based on underlying rules. This finding indicates that country or region-specific rules are not shared across lists.

3.3.3 Differences on Domain Level On the rule level, most lists differ extensively. However, it might be possible that the rules are designed to block the same domains but with (slightly) different regular expressions. To assess this, we utilize the *JustDomains* tool [38]. JustDomains is a tool that takes a filter list as input and provides a list of all domains (eTLD+1) that would be blocked by the rules on the filter list. Thus, the tool allows us to compare if the localized filter list aims to block different trackers or if they block similar trackers but with different rules. We computed the intersection between two lists created by JustDomains to understand the overlap in blocked domains. Fig. 3 provides an overview of the similarity between the lists. The results show nearly no similarity (i.e., overlaps) between different lists. On average, the overlap in blocked domains is 0.6% (min: 0%, max: 26.7%, SD: 2.9%), if we exclude self comparison. These results are in line with the low overlap in rules, indicating that each list aims to block a different set of trackers.

Lessons learned. The comparison of similar rules across the analyzed lists and the fact that it seems hardly possible to cluster them indicates that these lists are different. Thus, the lists aim to block different requests of specific domains for a given region. This observation shows that users who visit sites and pages not frequently visited by other users in their region may be more susceptible to tracking, as their used lists do not protect them from common trackers on such pages.

4 Measuring the Impact of Localized Filter Lists

This section describes the measurement framework that we use to analyze differences in the effectiveness of localized filter lists in the wild. We provide an overview of our dataset (Section 4.2), and discuss the impact of localized filter lists on user privacy (Section 4.3). Finally, we perform a runtime analysis of filter lists of different sizes (Section 4.6) and test how many rules are effectively used to block requests in our measurement (Section 4.7).

4.1 Experimental Design

4.1.1 Measurement Vantage Points We must measure user web traffic from different locations to measure the real-world impact of region-specific (or localized) filter lists. To select the locations, we identified intersections between the available instances of different cloud providers (i.e., Amazon Web Services (AWS), Google Cloud Platform, and Microsoft Azure) and the identified localized filter lists for the specific country (see Section 3.2). We identified five intersections for the Google Cloud Platform, seven for Microsoft Azure, and nine locations for AWS. For our analysis, we used Amazon Web Services (AWS) to simulate the different user locations because it offers the most locations for the analyzed filter lists, which is crucial for our experiment. We decided not to use multiple cloud providers to avoid biased results (e.g., due to blocking of specific cloud providers [37, 56]). This step also increases the reproducibility and comparability of our results because the technical setup is consistent in resources and technical latency when using only one cloud provider. Table 1 shows the analyzed filter lists and AWS instances we use in our experiment.

4.1.2 Websites to Analyzse Our experiment uses two measurement profiles that simulate the described scenarios. We selected the sites to analyze based on the Chrome UX Report (CrUX) [31] of January 2024 (version 202401), as it is the only toplist that provides extensive coverage of top sites for the analyzed regions. We selected the first 10,000 eTLD+1 sites from each country for our scope. To capture a representative behavior of each site, we randomly selected 15 pages from each domain for our analysis [4, 53]. To collect the subpages, we looked for first-party links on the landing page and used them for our analysis, a strategy commonly used [14, 16]. If we did not identify enough links on the landing page, we recursively looked at the identified subpages. From all identified links, we randomly sample 15 for our analysis. We captured these pages before we performed the measurement to visit the pages with our framework. We provide a list of the analyzed sites, pages, and retrieved categories in Section A. We use two profiles to simulate the privacy challenges described in Section 3.1 using the localized lists.

Profile 1:. In this profile, we simulated a user visiting the top pages of the user's location (see Scenario 1 and 3). To implement this, we ran measurements from different locations and visited the top 10,000 sites (eTLD+1) and pages on these sites for this location.

Profile 2. We also simulated that a user is visiting pages that are commonly visited by users from different locations (Scenario 2). Thus, we ran the measurements from each location and visited the top 1,000 sites and respective pages from all other locations. In total, we visited 8,000 sites in each location in this profile.

4.1.3 Measurement Framework We built our experimental setup using the measurement framework by Demir et al. [12, 14]. The framework consists of one master VM that orchestrates an experiment and manages multiple 'agent' VMs that conduct the measurements (one for each profile in a measurement run). Each VM runs a separate crawler with a distinct configuration. The profiles are described in Section 4.1.1. Each measurement uses OpenWPM [24] (v0.27.0), which uses the Firefox browser (Version 123.0), with the user agent (*Mozilla/5.0 (X11; Linux x86_64; rv:123.0) Gecko/20100101 Firefox/123.0*) and a screen resolution of 1920x1080 to visit the pages of interest. We configured OpenWPM to collect all HTTP(s) traffic, the content of the cookie jar and local storage data, DNS resolution, JavaScript (JavaScript cookies), and site visits. We performed a stateful crawl for each visited site, meaning we kept the browser's state when visiting all site pages and reset it before analyzing a new site. Based on the framework, we implemented two different ways to mimic user interaction: (1) simulating keystrokes and (2) pseudo-random mouse movements. Regarding the keystroke, we wait for the page to finish loading (or at most 30 seconds). We simulated three `Page Down` keystrokes followed by three `Tab` keystrokes, and finally, an `End` keystrokes with minimal periods of delay in between, ensuring the page is fully scrolled and rendered. We simulated the mouse movements via JavaScript to randomly move the cursor over the page. We store all collected data in a BigQuery [29] database.

Before measuring, we collected 15 pages from each site that we included in our analysis, which means that we analyzed up to 16 pages per site (i.e., the landing page and 15 subpages). Therefore, we configured the used framework to automatically visit each domain and collect the linked subpages (i.e., first-party links). To collect the pages, we used AWS EC2 instances located in specific countries to get the local IP address of each region. For each measurement profile in each region, we used an AWS EC2-instance (OS: Ubuntu 20.04, RAM: 32 Gib, CPU: 8 (3.0 GHz)/4 (peak of 3.1 GHz) depending on availability in the specific region) and installed our measurement framework on it. Each instance is supplied with the sites and pages to visit, and then successive visited. During the measurement, we did not use any filter list-based tool (e.g., a tracking blocker). After each measurement, we utilized the identified lists (see Section 3.2) to flag each request whether the specific list would have blocked it. To flag if an observed URL would have been blocked by a given filter list, we utilize the Rust library adblock-rust from Brave [41]. We slightly adjusted the library to not only return a boolean value if a URL had been blocked but also to return the rule that would have led to the request being blocked. Thus, for each URL, we recorded if a list would have blocked that URL and, if so, which rule or rules would have blocked the request to the URL. As we applied all filtering rules to each profile, we can compare the lists' effectiveness based on the same dataset. It must be noted that this approach assumes there is no dependency between different HTTP(S) requests, which is not entirely correct. For example, if a website includes an iframe from a third party that a block list would have blocked, all follow-up requests that load the content of this iFrame would also have been blocked. Our approach does not account for that. Thus, our approach will report an upper bound of trackers since a URL might also block the consequently loaded requests if it belongs to a tracker on a block list. The approach of identifying trackers after the measurement is common [7, 35, 43, 54]. Furthermore, it

		AWS Instance		Filter List					
No.	ID	Region	City	Target countries	∑ Rules	Δ EasyList	Link	Stand-alone	Version
1	US	US East	Ohio	EasyList (USA)✱	74,564	–	↗	✓	202405080949
2	CN	Asia Pacific	Hong Kong	China	93,561	18,997	↗	✓	202405081021
3	JP	Asia Pacific	Tokyo	Japan	5,911	5,633	↗	✗	2024/04/26 19:54
4	IN	Asia Pacific	Mumbai	India, Nepal, Bangladesh	83,452	8,888	↗	✓	202405081021
5	DE	Europe	Frankfurt	Germany	80,096	5,532	↗	✓	202405081021
6	NO	Europe	Stockholm	Norway, Denmark, Sweden	81,182	6,618	↗	✓	202405081021
7	FR	Europe	Paris	France	20,500	20,393	↗	✗	202405081021
8	IS	Israel	Tel Aviv	Israel	75,797	1,233	↗	✓	202405081021
9	AE	Near East	AE	United Arab Emirates	1,850	1,850	↗	✗	202405081021

Table 1: Used filter lists and corresponding AWS instances. ✱ The standard EasyList is our baseline. Δ indicates the number of rules not present on the baseline list. The target countries represent locations where the filter list is typically used.

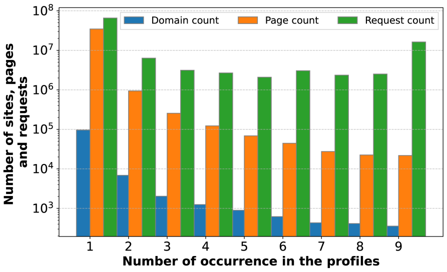

Figure 4: Occurrence of sites, pages, and requests across all profiles (logarithmic scale). The x-axis denotes in how many profiles a site, page, or request was observed.

would not be feasible to perform a large-scale measurement for each individual list due to scalability reasons. We make our setup and the collected data available (see Section A).

Statistical Analysis. Throughout the paper, we use the Kruskal-Wallis test [39] to evaluate whether disparities exist in the central tendency (median) of a continuous dependent variable across multiple groups. For all analyses, we maintain a 95% confidence interval ($\alpha = 0.5$) and use the η^2 test to gauge the effect size of a Kruskal-Wallis test. We define an effect as *small* if $\eta^2 \leq 0.06$, *moderate* if $0.06 < \eta^2 < 0.14$, and *large* if $\eta^2 \geq 0.14$ [10].

4.2 Measurement Dataset Overview

Using our measurement framework, we successfully visited 95% (min: 7,812, max: 9,421, SD: 680) of the identified sites, on average, and in total 1,828,493 (min: 90,906, max: 123,143, SD: 10,710) pages on these sites. On average, we visited 12 (min: 1, max: 16, SD: 6.5) pages per site. We visited 112,047 distinct sites and 1,024,676 pages. Figure 4 provides an overview of the occurrence of sites, pages, and requests in the different profiles. The figure shows the overlap of sites, pages, and domains across the profiles (e.g., how many pages were only visited in one profile). 88% of the analyzed unique domains only appear in one profile, meaning there is only a small overlap in popular domains across all regions. Furthermore, only 356 sites appear in all profiles. Finally, we found that 21,774 pages only appear in one profile. This observation shows that not only are the languages (i.e., domains) of a page unique, but also the subpages of these sites are not shared across regions. The distribution of requests among the profiles is more diverse, showing a notable overlap between them. Thus, while each visited page loads some content specific to the page, there are resources used by multiple pages and across profiles (e.g., images or trackers). Our observation suggests that regional filter lists might be needed as many pages and sites are only present in specific profiles. Thus, a single filter list would have to hold several rules that might only be used for pages that are only relevant to a fraction of users. In our experiment, we observed over 207 million HTTP requests and stored over 579 GB of data. Table 2 provides an overview of the measured data.

Identified Known Trackers. To understand the impact of the analyzed filter lists, we first compare the number of blocked requests by each list. Table 2 shows the number of identified trackers in each measurement run. Across all profiles, the baseline list (i.e., standard EasyList) identified 47,817,273 (23%) trackers from 6,967 distinct eTLD+1s across all measurement runs. We grouped the trackers by eTLD+1 to provide an indication of (1) how many different domains/organizations are responsible for the tracking requests and (2) to use the comparision eTLD+1 for further analysis. Regarding the localized lists, it is interesting that five of the lists that are designed for a specific location worked better in other locations (in terms of blocked requests). Especially the lists from the US (standard EasyList) and Japan performed well across all measurement profiles. These lists do not hold most rules (see Table 1). We found no statistically significant effect between the number of rules in a list and the number of blocked requests. On average, the lists blocked 79.78% less unique requests than the US list (avg: 79.78%, min: 11.99%, max: 99.5%, SD: 30.74%). Overall, 71.28% of the requests that would have been blocked by the standard EasyList would have also been blocked by at least one localized list, meaning that different rules lead to blocking a URL. There are severe differences in the effectiveness. For example, the German list, which contains 5,532 unique rules, blocks 41% of the requests that the US lists would block. The French list, which contains 20,393 rules, would block 7% of such requests. On average, a localized list blocks 12.33% of the requests that the EasyList would block (avg: 12.33, min: 0.02, max: 44.53, SD: 18.94). These results imply that the lists protect users to different extents based on their locations and the sites they visit.

		Analyzed		Cookies					Identified Trackers								
No.	Profile	Sites	Subpages	Unkno.	Necess.	Functio.	Perfor.	Ad/Targ.	AE	CN	DE	FR	IS	IN	JP	NO	US
0	AE.AE	9,009	96,877	3,758,577	314,762	166,367	854,944	1,231,417	70,792	72,712	1,591,599	322,609	11,365	159,048	2,271,223	113,518	2,916,015
1	AE.INT	7,862	94,070	3,504,266	284,870	163,365	717,760	1,169,901	69,509	110,495	1,602,255	348,995	13,275	125,439	2,869,242	131,951	3,496,173
2	CN.CN	9,163	95,698	2,184,326	175,071	72,795	398,088	555,357	42,385	215,896	537,197	171,787	1,569	72,363	1,072,443	70,803	1,235,006
3	CN.INT	7,898	95,106	3,572,406	306,500	171,400	746,143	1,102,507	71,353	93,919	1,738,538	384,788	13,792	132,193	3,149,406	151,129	3,776,849
4	DE.DE	9,423	123,143	1,789,670	292,747	111,132	360,194	162,496	109,874	89,489	245,575	229,326	15,064	90,964	891,945	289,542	973,819
5	DE.INT	7,940	92,831	1,984,507	257,131	105,002	472,688	272,833	63,000	109,957	435,669	291,015	12,855	72,906	1,566,296	122,997	1,424,820
6	FR.FR	9,345	116,426	1,726,714	287,401	114,736	427,350	185,151	64,728	68,443	245,379	251,974	9,846	67,085	948,217	132,306	907,988
7	FR.INT	7,937	93,913	2,023,274	279,026	116,024	481,233	275,707	68,891	111,358	401,145	301,771	12,746	76,424	1,546,377	142,244	1,410,360
8	IS.IS	9,177	111,529	3,521,347	264,026	146,972	1,049,727	875,432	186,252	80,439	1,144,677	518,415	38,262	194,891	2,377,522	111,810	2,596,534
9	IS.INT	7,935	93,795	3,300,797	291,299	156,956	638,305	988,116	58,199	119,535	1,376,454	357,826	4,851	107,768	2,769,964	145,127	3,240,390
10	IN.IN	9,267	98,727	2,596,974	270,506	120,949	698,332	739,039	61,046	47,447	1,530,389	285,447	6,710	145,545	2,081,332	89,549	2,736,181
11	IN.INT	7,902	94,900	3,344,793	287,209	167,151	696,363	1,000,235	66,203	116,165	1,275,258	330,597	13,456	110,469	2,726,502	132,277	3,000,770
12	JP.JP	9,161	112,257	4,694,217	805,957	198,707	1,067,546	1,448,750	81,813	174,850	1,708,237	500,945	7,273	197,656	4,509,725	84,790	3,696,261
13	JP.INT	7,916	93,192	2,993,598	237,785	146,113	632,548	774,804	65,844	106,032	1,233,537	320,424	14,103	103,877	2,325,961	142,060	2,862,989
14	NO.NO	9,173	120,479	1,849,223	318,695	122,411	497,193	198,548	72,546	62,132	280,798	268,844	20,328	91,965	1,086,358	192,213	936,292
15	NO.INT	7,813	90,906	1,936,953	268,004	108,990	466,995	264,463	65,319	88,882	383,436	288,211	11,922	70,556	1,471,478	121,786	1,390,004
16	US.US	9,273	109,581	7,698,833	722,356	403,457	1,434,344	2,843,014	146,464	96,109	3,003,330	594,850	15,131	170,892	4,941,606	250,583	6,644,538
17	US.INT	7,949	95,063	4,013,520	255,585	173,067	689,329	1,390,924	74,998	124,101	2,144,501	422,233	14,452	120,503	3,538,402	139,087	4,640,747

Table 2: Results from the measurement run from the different vantage points. The first part of the ID represents the country from which we visited the sites, and the second part is the country used to list(s) of the visited pages (e.g., US.INT indicates that we visited the top international pages from a server located in the US).

Cookies. On average, we recorded 58,945 (min: 43,910, max: 105,040, SD: 13,902) distinct cookies per measurement run. In total, we identified 20,526,074 distinct cookies (22% of all cookies) . We identified them by name, path, and domain. Furthermore, we found a statistically significant effect (p-value < 0.0001) of the measurement profile on the number of set cookies with a moderate effect size (i.e., in each profile a different number of cookies is set). Using *Cookiepedia* [11], we could classify the purpose of 41%. This seemingly low share is comparable to other studies that used Cookiepedia [14, 53]. The API classified 3,035,132 (15%) of the cookies for "Targeting/Advertising". Most cookies classified as "Targeting/Advertising" appear in the US-based profiles. Overall, most cookies are used for technical and not for tracking purposes. On each page (by eTLD+1) and on average, all lists combined would have blocked 759 (min: 1; max: 2,510,324; SD: 23,531) requests that would have resulted in a response that sets a cookie. On one visited site, we observed one request where the response would set a cookie. On the upper bound, one site sent 2,500,00 requests which would result in a cookie being set.

To get a first indication of how different block lists might affect the number of cookies set by a page, we cross-compare the cookies set in each measurement. We use the Jaccard index to compute the cookies' similarity in the different profiles. The average similarity between the profiles is 0.16 (min: 0.05, max: 0.42, SD: 0.11), meaning that the sets of cookies between all profiles are rather different. Consequently, filter lists that block a different set of requests might impact the cookies set. The highest similarity is between the German and French profiles. Furthermore, all lists would have blocked 74,178,594 requests that would have set 31,729,456 cookies.

Lessons learned. The high-level analysis of the impact of the localized lists and the measurement shows that different trackers are present in the different measurement runs and that the analyzed lists block a different proportion of them. These findings indicate that based on the privacy threat model described in Section 3, users must select the lists they use carefully.

4.3 Understanding the Impact of Localized Lists

We now discuss the impact of localized filter lists and analyze the effects of different lists based on locations and categories.

4.3.1 Effects of Different Filter Lists First, we test the effect of local filter lists on identifying tracking requests. Figure 5 shows the average fraction of identified tracking requests by local filter lists for different profiles at the page level. At a high level, we can see that filter lists show varying performances in identifying such requests. Overall, the baseline profile (US) identifies, on average, 13% of all requests as tracking requests per page. The local filter lists for China, India, Israel, and the United Arab Emirates show a low detection rate, averaging under 1.3% tracking requests, corresponding to less than 10% of the tracking requests identified by the baseline profile. The French and German filter lists show moderate identification rates of 2.6% and 5.1% tracking requests, respectively. Most notable is the performance of the Japanese filter list, with an average identification rate of 14.3% tracking requests per page. The number of rules in the Japanese list corresponds to 8% of the rules in the US list, but the list outperforms the baseline in the Japanese measurements, showing an optimized local effect (see Table 1).

In the following, we test the statistical differences of the local filter lists on the page level and how their performances vary. We run our test with *Repeated Measures ANOVA*, which allows us to test statistically significant differences in multiple dependent samples (tracking requests) across independent variables (local filter lists). The assumptions for this test, such as homogeneity, were evaluated and satisfied. However, due to a violation of sphericity, we applied the *Greenhouse-Geisser correction* to adjust the results accordingly. We found a statistically significant different effect (p-value < 0.001) on the number of identified tracking requests across different filter lists. The results indicate significant variability in the effectiveness of rules in the local filter lists, with some regions, such as China, India, Israel, and the United Arab Emirates, performing worse than the other filter lists. In Section 4.7, we discuss the efficiency of each rule for identifying tracking requests.

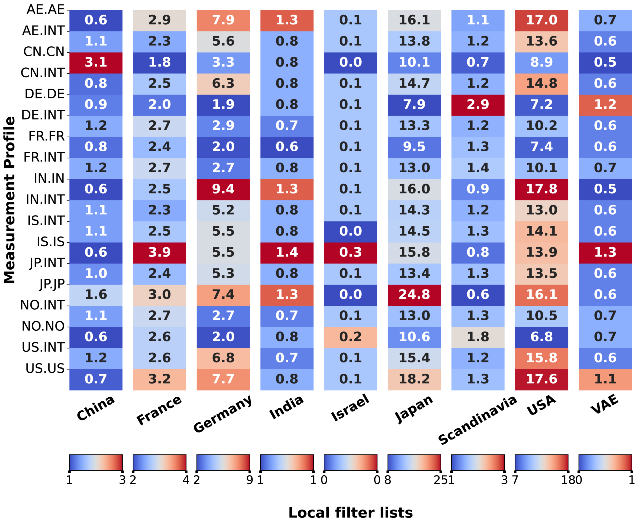

Figure 5: Fraction of HTTP traffic identified as tracking requests by measurement profile and filter list at the page level.

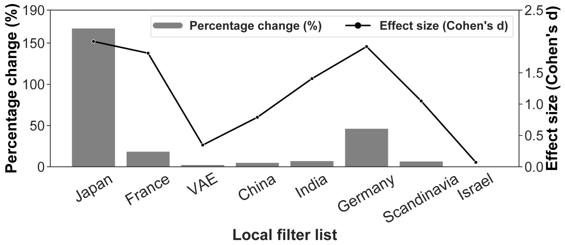

Figure 6: Change in the number of identified tracking requests and effect size per filter list compared to the baseline.

Next, we test the impact of combining local filter lists with the baseline filter list. More precisely, we evaluate how using local filter lists alongside the baseline affects the identification of tracking requests at the page level, using percentage change and *Cohen's d* effect size as metrics. We chose Cohen's d to compare the means and quantify the effect size between conditions. In this analysis, we add the number of tracking requests per page identified by the baseline to those identified by the local filter lists to determine the total identifiable tracking requests. The results, demonstrated in Figure 6, show that the effects of local filter lists vary significantly. Most notably, the Japan filter list had the most substantial impact, increasing the number of identified tracking requests by 168% with an effect size of 2.0. Germany and France followed, with increases of 47% and 19%, respectively, and effect sizes of 1.92 and 1.81. In contrast, filter lists such as Israel and AE had minimal impacts, with percentage changes of less than 3% and small effect sizes. These findings highlight the varying efficacy of local filter lists in enhancing tracking request identification, with some lists showing substantial improvements and others having negligible effects. The significant variability in the impact of local filter lists suggests that regions like Japan, Germany, and France have more optimized and effective rules that complement the baseline list well. When used alongside the baseline list, these rules substantially increase identified tracking requests. The minimal impact observed in regions like Israel and AE may be due to their rules' lower specificity or outdated nature, indicating a need for further refinement and updates.

4.3.2 Differences in User Locations We now analyze the effect of localized filter lists on users' privacy depending on the user's location. Along the defined privacy challenges (see Section 3.1), we study how effective the different lists block requests on top pages users of a location commonly visit as well as other pages. We begin the analysis by assessing the effect when users visit the top pages of the user's location (Scenario 1 in Section 3.1). Only 3 (38%) lists worked best in the region they were designed for in terms of the number of blocked requests. Thus, while the localized lists increase the protection of users' privacy, they do not show a specialization for websites or trackers that are common in the region for which the list was designed. The baseline and Japanese filter lists block most requests. We found a statistical effect for the used filter lists on the number of blocked trackers (p-value < 0.0001). On average, the country-specific filter lists in scenario 1 block 1,368,280 (12.99%; min: 38,262, max: 6,644,538, SD: 2,446,242) requests. This is partly due to rules that do not block a specific domain but block URLs based on a specific pattern (e.g., `/graphics/ads/*`).

However, we see a different picture if we look at the requests that are *only* blocked by the list for the specific region. For example, the German (DE) list shows a low coverage of non-German trackers. On average, the list exclusively blocks only 478 (min: 65, max: 1123, SD: 371) requests in each of *other* profiles. In contrast, all other lists would exclusively block 153,961 (min: 48, max: 1,607,728, SD: 310,658) requests, on average. Thus, the German list seems to include rather general rules that apply to various domains, trackers, and regions (see Table 2) that are also covered by other lists. On the other hand, the non-German lists seem to have more specific rules that are not covered by other lists, but these rules are not specific to one region. Section B provides an overview of the number of rules blocked by the different lists for our Scenario 1.

In the following, we analyze the effects when users visit the top pages of regions the user is *not* located in (Scenario 2 in Section 3.1). In our measurement, we found that only one (14%) of all lists worked better when visiting the 'international' pages rather than visiting the 'local' pages. On average, the lists blocked more 178% (min: 0.9%, max: 3,655%, SD: 444%) requests than for the local setting. Percentages over 100% indicate that the list worked better in the local setting. These observations show that while the used lists are optimized for the users' country, they still offer additional protection for other locations. Finally, we look at the effectiveness of these lists when a user browses on websites they commonly visit which currently are in a different geographical location (Scenario 3). In our measurement, five (56%) of all analyzed lists blocked most requests in this scenario. Similar to the findings in Section 4.3.1, this observation is unexpected, as the used lists, except the baseline, were not designed for this purpose. Overall, for this scenario, we found 38,544,174 requests that were only blocked by one list not designed for this purpose (i.e., the list was neither designed for the location of the user nor the location of the visited page). Figure 7 provides an overview of the number of requests each list exclusively blocked in the different scenarios. Exclusively means that a request

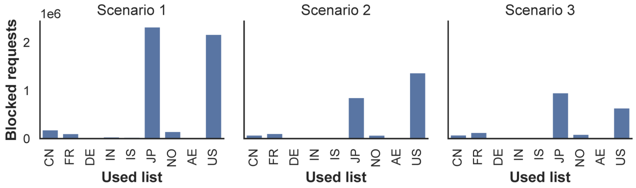

Figure 7: Number of requests *exclusively* blocked by the analyzed filter lists in the different scenarios.

is *not* blocked by any other list. In all scenarios, the baseline list (i.e., US) and Japanese list (i.e., JP) outperform all other lists. This observation is expected for the baseline due to its size and common usage in many locations. Regarding the Japanese list, the results show that while the list only holds 5,633 rules (Table 2), it still outperforms many other lists, indicating that a careful selection of rules can vastly increase the quality of a list. The combination of the regional filter list and one other filter list blocked on average 1,862,990 (22.96%) (min: 63,050, max: 11,586,144, SD: 2,296,942) more requests than just the region-specific filter list.

4.3.3 Effects of Categories Previous work has shown that some site categories (e.g., News pages) contain more trackers [51, 53] than other site categories. Thus, in this section, we analyze if the lists (or some of them) perform better or worse on specific sites and pages. We utilize the Google Topics API [30] to classify the category of the different sites. We could classify 48,106 (42%) of the analyzed sites into 24 different categories. The remaining sites (58%) could not be classified by the API (i.e., the category is "Unknown"). This is the case for sites that serve adult content (e.g., gambling websites) and sites the API cannot classify. 17,634,612 (8.5%) sites were classified as "News" and 6.7% sites as "Arts & Entertainment". The lists blocked most requests in the category "Internet & Telecom" (5.5%). The category from a website has a statistically significant impact on blocked requests (p-value < 0.0001).

Figure 8 provides an overview of the requests blocked by the filter lists in the different categories. For the categories that hold the most pages in our experiment ("News" and "Arts & Entertainment"), the Japanese list identified most tracking requests(51% and 56% of all identified requests). The lists blocked most requests in the category "Internet & Telecom". The baseline list performed best within this category by blocking 38% on all identified tracking requests. We assume that sites in this category (e.g., "Jobs & Education" and "Shopping") mostly target users from across the globe; the EasyList is optimized for this use case. Overall, the Chinese filter list performs best in most categories (30% of all categories), followed by the filter lists from Japan (26%). The categories where the Chinese list is most effective have fewer requests and sites, which explains the overall low performance of the list (e.g., "People & Society", which presents 0.5% of blocked requests). The baseline list performs best in only three categories (i.e., "Internet & Telecom", "Shopping", and "Jobs & Education"). Based on the number of blocked requests, the list from China has a low effect on user privacy but still blocks most requests in some categories.

The results show differences to the findings of related work [51, 53], where the most blocked requests appear on the "News" sites.

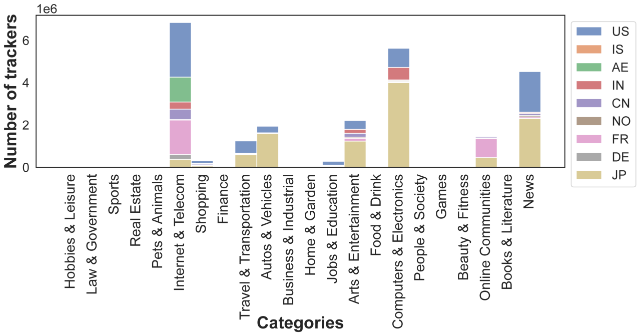

Figure 8: Number of blocked trackers per filter list of each category. Some categories have fewer blocked requests than others and some dominate with a total number of trackers.

It has to be noted that EasyList performed better in the news category than all other categories (except "Internet & Telecom"), which could explain this shift in the findings compared to previous works that only use EasyList. Based on our privacy challenges, a user in Scenario 3 benefits from multiple filter lists while browsing local and international websites. The user in Scenario 2 can run into several privacy issues while using an insufficient filter list.

Lessons learned. Based on the number of blocked requests, we see that only some lists (i.e., Japan and Germany) blocked a relevant number of requests in our measurement. Our experiments have shown that the localized filter lists often do not meet the expectation that they protect users from localized trackers or perform notably better in specific regions. Overall, these observations raise the question of whether the effort that volunteers into contributing to these lists is conducive or whether the community should focus on optimizing the standard EasyList.

4.4 Analyzing Combined Lists

Previously, we have shown the effects of the rules present in the localized lists. In the following, we analyze the effects of combining a localized list with the baseline list. Therefore, we are adding the same rules (i.e., the ones on EasyList) to all regional lists and comparing the effects of the resulting lists.

4.4.1 Combining Local Filter Lists and the Baseline This section assesses the effects of combining the baseline filter list (i.e., EasyList) with a local filter list. For that purpose, we analyzed the number of blocked requests using (1) the local filter list and (2) one local filter list and the baseline, and (3) only the baseline in the country-specific profiles (i.e., US.US, CN.CN). Figure 9 provides an overview of the blocked requests by these lists (blue bars) in contrast to the request, only blocked by the regional list (orange bars), and the baseline list (gray bars). On average, 88.46% (min: 75.43%, max: 98.55%) of the requests that are blocked using the combined list would not be blocked by the local filter list. Thus, our approach of removing the rules of the baseline list from each regional list is reasonable. For locations like the United Arab Emirates and France, where the baseline list is not part of the fabric list, the combination increases

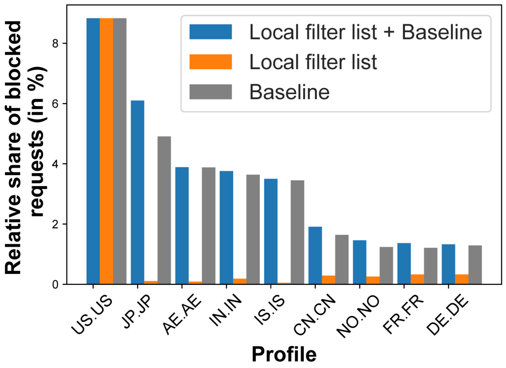

Figure 9: Combination of filter lists and the baseline list. The chart shows the difference between the standalone local filter lists and the combination of baseline and local filter lists.

the number of blocked requests by 97.58% (AE) and 75.54% (FR). The visual inspection of Figure 9 shows that in some locations, the ratio between the blocked requests of the local lists and the baseline list is smaller (e.g., DE, FR, NO, or CN) than in other regions (e.g., JP, AE or IL). This observation propagates to the combination of lists where the combination often blocks lists and blocks fewer requests. Thus, it seems that some regional-specific lists are more suited to the designated regions than others, but at the same time, the baseline lists seems less effective in these regions. The results show that using a combined list will significantly increase the number of blocked requests. Furthermore, it strengthens the usage of the baseline filter list. Some local filter lists already contain some of the rules from the baseline, but not all of them.

4.4.2 Effects of Combining Lists Previously, we have shown that the lists do not work as expected in the defined scenarios because they do not provide the best protection in the regions they were developed for. Therefore, it is interesting to analyze the effects when users combine multiple or all lists to maximize their protection. As already described, the lists with the most blocked requests are the baseline, Japanese, and German. When combining the localized filter and baseline lists, each profile blocked on average,3,794,718 (min: 1,128,505, max: 13,289,076, SD: 3,296,845) requests. The base list increases the number of blocked requests by 5,306% (min: 81.9%, max: 66,797%, SD: 15,479%) on average. With the combination of at least two lists (e.g., DE and Japan) on average 1,547,361 (22%) (min: 43,954, max: 11,586,144, SD: 1,676,152), requests can be blocked. Combining three lists increases the value to an average blocked request of 1,848,490 (+15.46%). Fig. 10 provides an overview of blocked trackers from each list in the different experiments. Next, we analyze the protection level if users would use a list that combines *all* analyzed lists. Based on Figure 10 and the different privacy scenarios (see Section 3.1), we identified three different groups of filter lists. The filter lists in the first group block more requests in the local profile (i.e., Scenario 1) than in the international profile. The second group of filter lists blocks notably more requests in the international setting (i.e., Scenario 2) than in the regional profile, and the third group blocks a similar number of requests in both profiles. In group one, two (23%) lists (US and JP) blocked more requests on the local profile than the international one. It is feasible that tracking in those locations is higher than on international websites. In the second group, we identified seven (77%) (CN, DE, FR, NO, IN, IS, AE) lists with more blocked requests on the international profile than in the local profile. Overall, those locations block 1,781,388 (48.4%) more requests on average in the international profile. The biggest difference is location CN with 178% more blocked requests than in the international profile. European countries (DE, FR, NO) are subject to the GDPR, which may impact the number of tracking activities in those countries: the number of international trackers is higher than in the local profiles. Users of locations in the second group face fewer trackers while surfing on their local sites than on the international ones. For the third group, we define a threshold of 15% for the similarity of blocked requests on local and international profiles. We could only identify one location (IN) that blocks a similar number (including the threshold) on their local and international profiles. We choose 15% as the threshold because we assume a fluctuation of rules between 5% and 15%. Accordingly, the protection of both profiles is only similar at one location, which means that one profile dominates at other locations. A user in scenario 3 is more likely to have problems with their privacy.In summary, we found different behaviors in the three privacy scenarios. For some locations (e.g., Germany) we observed fewer trackers on the local profile than on the international one. This could be because of fewer tracking activities compared to other locations (e.g., US) or less detection through the filter lists. Users face more tracking activities in some countries (e.g., US and JP). Possible reasons are a greater market for targets or advertising.

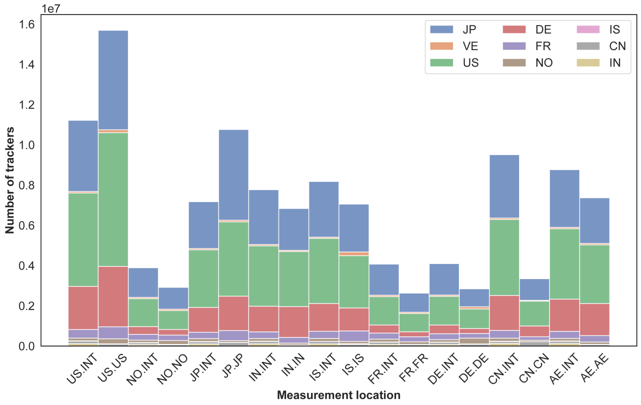

Figure 10: Number of blocked trackers per filter list for each location. For each measurement location, both profiles (local and international) are shown.

4.5 Impact of Commits and Rule Additions

Previously (Section 4.4), we have shown that the effectiveness of the different localized filter lists varies. In the following, we analyze if the size and activity of the community that maintains the lists impact its effectiveness. To do so, we perform a statistical analysis of the number of tracking requests and the activities within a filter list repository. Therefore, we use the *Spearman Correlation* [57] between blocked requests and commits from a filter list. First, we want to analyze the impact of the number of commits on the number of blocked requests. Further, we found no statistically significant correlation between commits and blocked requests (p-value < 0.7). Also, it is easier to assume other suggestions with further data or factors. We found no correlation between the number of added (p-value > 0.7), modified (p-value > 0.7), or removed (p-value > 0.4) rules. Finally, we tested the effect of the size of a community that maintains filter lists and the list's effectiveness. We found no correlation on the blocked requests size of the community (p-value > 0.4. The results show that the size of the community may not affect the number of blocked requests. The activity in commits of the individual lists can influence the number of trackers identified. Thus, the results indicate that the effectiveness of a filter list is not dependent on the number of changes and potential timely reactions to new trackers. This observation could indicate that carefully crafting a few rules is more important than adding or modifying several.

4.6 Runtime Analysis for Filter Lists

Previously, we have shown that the effectiveness of the different lists varies and that each list seems to block different trackers in all analyzed regions. Therefore, combining multiple lists into one large list might be an option to maximize user protection. We study the effects of a filter list's size on its runtime, required resources, and effectiveness in blocking trackers.

We implemented the test framework using the Rust library *adblock-rust* from Brave [41]. We combined all filter lists to perform runtime analysis and took their distinct rules (143,654 rules). We simulated our experiment with 100,000 URLs, which we randomly selected from our corpus. We generated five different rule sets for this analysis and tested each of them individually. For each set, we reduced the size of the set by 20.000 rules and randomly removed them. We used a virtual machine (VM) for this analysis that provides resources similar to a typical end-user device. The used VM has one CPU (2,095MHz) and 16 GB RAM and runs Ubuntu as the operating system. During the runtime analysis, no other process (aside from the standard processes of the OS) was running on the host system. This setup is reasonable since an extension (e.g., ad blocker) in a Web browser will never use all resources of a system (e.g., only a single CPU might be used). To collect a reliable data set, we tested the runtime and memory usage of each list in 300 rounds and used the average value for further analysis. We measured a mean runtime of 14.94s using the full set of rules, resulting in an average of 0.000104s per rule. During that time, the host system used a total of 9.615 GB RAM. We have to subtract about 1.5,GB of memory usage for other processes (1.2,GB) and the storage of the URLs used in the memory (0.3,GB). Therefore, we identified an average memory usage of 8.115 GB. In Section D we provide a graphical overview of the runtime analysis.

In our experiment, adding further rules increased the total number of blocked URLs, but the ratio between blocked URLs and the rules used decreased. Accordingly, creating a master filter list would increase the runtime and memory consumption but the efficiency of adding further rules decreases. The most interesting results are in blocked URLs: our experiment showed that while more filter lists block more URLs, the efficiency of additional rules decreases. Larger rule sets offer diminishing returns, increasing runtime, and resource usage without proportionate benefits in blocked URLs.

Lessons learned. Our runtime analysis has shown that more URLs are blocked when the number of rules used increases. However, integrating new rules increases the runtime and the resources used linearly. For this reason, a new filter list should not be completely integrated, but the rules used within the filter lists should be considered, as we show in Section Section 4.7.

4.7 Usage of Rules

In this section, we test the effect of each rule on all filter lists to re-visit and replicate the results presented by Snyder et al. [50], who found that many rules in a filter list are not used in the field. Using our method of identifying tracking requests described in Section 4.1.3, we can determine which rules lead to the identification of tracking requests. Our dataset has a total of 122,548 rules. Note that for this analysis, we only test rules for identifying tracking requests and not for hiding elements in the DOM (e.g., hiding consent banners). Only 8,163 (6.6%) distinct rules identify at least one tracking request. Based on these high-level results, we see that most rules (93.3%) do not identify *any* tracking requests and are never used, at least in our experiment. In the supplementary material of this paper (see Appendix A), we provide an overview of all rules that identified at least one tracking request as a master filter list.

Figure 12 shows the number of rules used and the number of identified tracking requests per filter list. The results indicate that most rules are unused, including those in the baseline profile (standard EasyList). While the Japan, German, and French filter lists have more rules that lead to identifying requests, supporting our results in Section 4.3.1, the other local filter lists have fewer rules that identified a tracker in our experiment. Figure 11 provides an overview of the proportion of rules needed to block all identified requests in our experiment. The main plot zooms into the upper arc of the ECDF plot to highlight the increase in the proportional share, while the subplot shows the entire ECDF plot. Only a few rules are enough to block most of the identified tracking requests. The results indicate that the top 100 rules cover 80% of identified tracking requests, the top 500 rules cover 97%, and the top 1,000 rules cover 99%. Thus, a fraction of the rules currently present in all filter lists would offer similar protection as the combination of all rules, while also considerably decreasing the runtime. Further, we analyzed how many different eTLDs+1 a single rule blocks. Therefore, we count each eTLD+1 as a rule block in the measurement profile. We provide an overview of how many different eTLD+1s a single rule blocks Figure 13 in Section C. Most rules block a single or very few domains; however, outliers exist. On average, a rule blocks 12 (min: 0 , max: 17,275 , SD: 260) eTLD+1. The rule with the most blocked eTLD+1s (17,275) is in the Chinese filter list. The rule (`.gif|$domain=7mshipin.org|viidii.info`) is designed to block an image (gif) loaded by two domains (7mshipin.org and viidii.info). The problem with the rule is that it did not only block requests from the two named domains but all domains (e.g., https://www.samsung.com/ajax-loader.gif) that load a gif. Thus, it seems to be a mistake in the committed rule. Another example of a malformed rule that blocks too many eTLDs is the rule

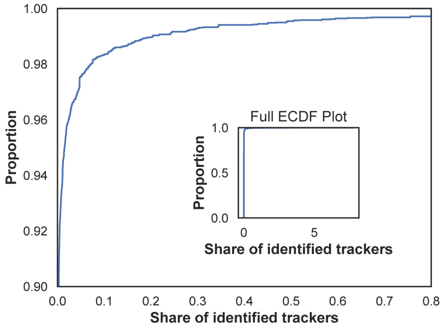

Figure 11: Each rule's share in blocking requests.

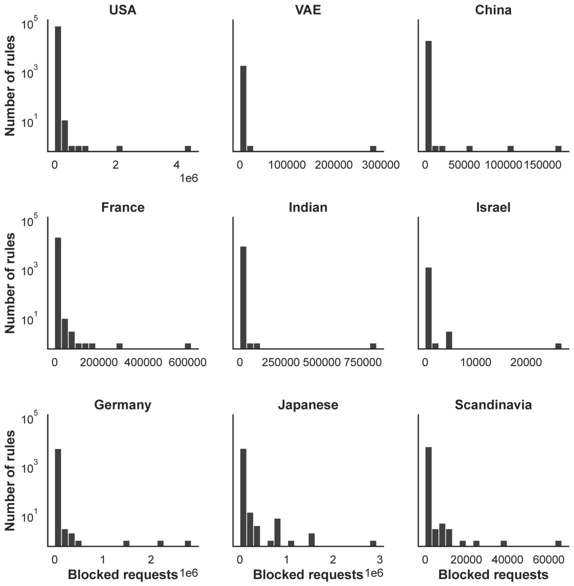

Figure 12: Overview of how often a rule was used in each list. The more right on the x-axis a bar is, the less often the rules used. Higher bars represent a large number of rules.

'`/banner/*$domain=arthparkash.com|firlive.com|jogsanjog-times.com|kanvkanv.com|keralafinance.com|khabor.com|laksam.com|nbs24.org|tamil.com|thereport24.com`' from the Indian filter list. This rule blocked 3,113 distinct eTLDs, all URLs that contained the phrase `/banner/`. These rules are candidates for a rule that could break different sites by being too coarse-grained, while it does not look like it at first sight.

In the following, we study the usage of each rule in our dataset. The results show that the percentage of used rules per local filter list varies significantly. Overall, 6% of the baseline rules identified at least one tracking request. The highest utilizations are seen for the Japanese (13.6%), French (6.6%), and German (4.8%) lists, indicating more targeted and efficient rule applications in these filter lists. Conversely, the lowest utilizations are observed for the Indian (0.7%), Israeli (2.4%), and Chinese (2.8%) lists, suggesting more underutilization or inefficiency in their filter rules. These results highlight overall the high inefficiency of the rules in the local filter lists. Based on these findings, we built a "master list" from those 8,163 rules and compared it to others. In contrast to the baseline list, the master list identifies 9,650,332 (83%) more trackers. One important aspect of the master filter list is that each privacy scenario would profit from this list. All local tracking requests can be identified in scenario one, just like in scenario two, which is all international tracking requests. Also, as shown in the previous section, the resource costs of using filter rules decrease enormously.

4.8 Potential Site Breakage

Understanding whether a blocking rule might break a website or other websites it was not directly built for is a non-trivial task [23]. Providing an in-depth overview of potential site breaks is not the goal of this study and was done before [47]. However, we want to provide some high-level figures on whether some of the nine analyzed lists might lead to more site breakage than others. Therefore, we perform a two-fold analysis: (1) We analyze if the rules lead to the blocking of first-party requests, and (2) we test if a rule that was designed for blocking a specific domain blocked a wrong domain. Blocking of first-party requests might not always lead to site break as first-party more and more also act as tracker [13]. The results show that across the entire measurement, only 72 (0.000037%) first-party requests were blocked by all lists. These requests were distributed across 43 (0.43%) visited sites. Next, we analyzed whether a rule that was designed for a specific domain might have blocked a wrong request. For instance, if a rule was designed to block the tracking URL tracker.biz but blocked a request to school.edu?q=tracker.biz. We could identify 31 (0.31%) of the domains that might block a wrong request. From the 43 blocked first-party URLs, all eTLD+1 are included in one filter list.

Lessons learned. Our analysis shows that a significant proportion of the rules in the local filter lists were not used in our measurement (only 6% of the baseline rules) and that a rather small list would provide similar protection rates. These findings highlight that in the future, maintainers of the filter lists should evaluate which rules in a list are no longer needed. These efforts also improve the performance and usability of the lists regarding their memory usage and runtime.

5 Limitations & Ethics

A data set with 100,000 URLs was used for the runtime analysis. Since we used an analysis device for the runtime analysis, the real usage of the working memory and CPU resource might differ on a user's device. Accordingly, our conclusions and recommendations for legitimate use have limitations. One major limitation of our approach is that we can only measure which requests a filter list would have blocked. However, we cannot measure whether this blocking would have resulted in page breakage, a common problem with filter list-based blocking approaches also observed by other works [49]. Another limitation is that a site can detect that we use Amazon Web Services (AWS) for visiting their pages and may deliver no content due to this configuration [32]. Further, our crawler creates traffic on the visited site. Also, we might see ads that incur provider costs from the site. Since our crawler visits each page only once, we assume these issues are minor and can be accepted. Furthermore, we identify tracking requests after running the measurement and not during the measurement, which could vary in the number of blocked requests. This approach is essential to maintain a consistent dataset for comparability across filter lists and to prevent the risk of page breakage and disruptions.

6 Related Work

The general idea of using a filter list is widely prevalent in computer science, especially with Web technologies. Approaches to blocking unwanted behavior can be found in other areas as well [17, 46, 52].

Filter List Generation. Feal et al. [27] analyzed the transparency and behavior of open-source blocklists. They looked at the overlap between specific providers in the open-source blocklists and found differences between open-source and commercial blocklist ecosystems. To reach this conclusion, they compared their results with the work of Li et al. [42]. Sjosten et al. addressed the problem of poorly maintained filter lists in underserved regions [48]. The author proposes a two-step filter list generation pipeline that combines deep browser instrumentation and an ad classifier. The pipeline is applied to three regions with poorly maintained filter lists (Sri Lanka, Hungary, and Albania) and generates new ones that complement existing ones. Smith et al. identified the problem that the continuous development of filter lists poses a problem for the authors, as the degree between "blocked" and "broken" is very small. For this reason, Smith et al. have developed a tool in their work [49] that automatically determines whether a rule is suitable for a filter list or is likely to break different websites. Synder et al. [50] analyzed the growth and health of filter lists. They find that 90.16% of the resource blocking rules on EasyList provide no benefit to users in the common browsing scenarios. They also find that checking a URL synchronously with a reduced list while asynchronously checking with a complementary list improves performance significantly while maintaining the coverage of the complete EasyList.

Filter List-based Ad Blocking. One study by Iqbal et al. introduces ADGRAPH, a graph-based approach to ad and tracker blocking [35]. The authors highlight that filter list maintainers can analyze disagreements between ADGRAPH and filter lists to identify and fix potential inaccuracies in filter lists. Additionally, ADGRAPH can support the generation of filter lists targeting under-served languages or regions on the Web. Another study by Iqbal et al. introduces a retrospective measurement and analysis of anti-Adblock filter lists [34]. The authors target the problem, where online publishers deploy anti-adblock scripts to detect adblockers from the users. Fouad et al. analyzed EasyList and EasyPrivacy, as well as the Disconnect Filter lists to find gaps in tracking mechanisms and the detection of these mechanisms [7]. Merzdovnik et al. [43] studied the effectiveness of popular tracking blocking extensions. They evaluated how many third-party requests each extension blocked.

7 Discussion and Recommendations

Our results show that localized trackers exist that are *not* identified by the general EasyList (i.e., our baseline). Analyzing the regional filter list ecosystem, we found that the only usage scenario for which three of the nine regional filter lists seem to work as intended is when a local user browses a local website (scenario 1). For traveling users who visit their regular local websites (scenario 2) and for temporary traveling users who visit region-specific websites (scenario 3), we identified a significant drop in the effectiveness of the regional filter lists when they were used without the baseline list. Our results show that the approach to combining the baseline list and localized list enhances user privacy for scenario one as well as for scenarios 2 and 3. Thus, we need to find ways to improve the regional lists further to boost their effectiveness and enhance the privacy protection they offer.

One way to do that could be to find more volunteers to work on these lists. Yet, our results show that the community size, the number of commits, and the number of rules added or modified by each commit do not impact the number of requests they block. Thus, the quality of the list is not directly determined by the size of the community maintaining it but by the rules created by the community. Thus, the localized (but also the standard) list maintainers and volunteers must consider these rules when creating them. However, creating these rules is not straightforward and requires some experience. Consequences (e.g., page breakage) may not always be foreseeable. Therefore, the research community should look for ways to support these communities and research ways that enable the volunteers to build effective rules more easily (e.g., [18]). We encourage future research to understand better why specific filter lists perform better in blocking efficiency and understanding the intricacies of the community-maintained filter lists.

Another idea could be to combine the efforts of all communities and create one large block list that should work for all regions in all of the defined scenarios, which would mean eliminating localized lists and merging them into the baseline list. Putting aside the fact that a simple merge could lead to unforeseeable incidents of site breakage, our runtime analysis has shown that further increasing the size of the used filter list could lead to significant overhead in using them in a real-world scenario as they would add too much latency. A mechanism that clusters and combines rules that block the same resources would be needed to support this approach. Such a method could also help sanitize lists from no longer needed rules.

8 Conclusion

In this paper, we analyzed the effectiveness of different localized filter lists and compared them against each other. We used the standard EasyList as a baseline and performed a large-scale measurement incorporating nine vantage points and 18 measurement profiles. Our experiments have shown that only two localized lists (i.e., German and Japanese) notably improve the protection of users and that most lists only identify very few trackers. We made this observation in all three defined scenarios. These results question the usefulness and need for these localized lists and show that most trackers are not specific to a given region. Our results reproduce, replicate, and support the findings of Snyder et al. [50] and show that most (93% compared to 90% in Snyder) of the rules present in all lists did not block any URLs in our large-scale measurement. Therefore, filter list maintainers should reconsider the mechanisms for removing redundant rules. Such steps could increase the lists' performance in terms of memory utilization and classification time.

Acknowledgments

The authors gratefully acknowledge funding from the *Federal Ministry of Education and Research* (grants 16KIS1629, 16KIS1628K "Ubi-Trans", 16KIS1648 "DigiFit"), the *Federal Office for Information Security of Germany* (01MO23033B "5Guide"). We also thank Google for the generous funding (EDU Credit 330580204) that supported this research. The authors used ChatGPT4 and Grammarly to revise the text in all sections to correct typos, grammatical errors, and awkward phrasing.

References

[1] Mozilla Foundation. 2024. Public Suffix List. https://publicsuffix.org/list/public_suffix_list.dat.

[2] AdBlock. 2024. Introduction to Filter Lists - Filter Lists For Non-English Languages. https://web.archive.org/web/20240327002504/https://helpcenter.getadblock.com/hc/en-us/articles/9738523403027-Introduction-to-Filter-Lists.

[3] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: Ordering Points to Identify the Clustering Structure. In *International Conference on Management of Data (SIGMOD)*. Association for Computing Machinery, New York, NY, USA, 49–60. https://doi.org/10.1145/304182.304187

[4] Waqar Aqeel, Balakrishnan Chandrasekaran, Anja Feldmann, and Bruce M. Maggs. 2020. On Landing and Internal Web Pages: The Strange Case of Jekyll and Hyde in Web Performance Measurement. In *ACM SIGCOMM Internet Measurement Conference (IMC)*. Association for Computing Machinery, New York, NY, USA, 680–695. https://doi.org/10.1145/3419394.3423626

[5] AudienceProject. 2024. Attitude towards advertising and use of ad blocking. https://web.archive.org/web/20240430001441/https://audienceproject.com/resources/insight-studies/attitude-towards-advertising-and-use-of-ad-blocking-2/.

[6] Collin M. Barrett. 2023. FilterLists. https://web.archive.org/web/20240528114401/https://filterlists.com//.

[7] Nataliia Bielova, Arnaud Legout, Natasa Sarafijanovic-Djukic, et al. 2020. Missed by Filter Lists: Detecting Unknown Third-Party Trackers with Invisible Pixels. *Proceedings on Privacy Enhancing Technologies* 2020, 2 (2020).

[8] Stefano Calzavara, Tobias Urban, Dennis Tatang, Marius Steffens, and Ben Stock. 2021. Reining in the Web's Inconsistencies with Site Policy. In *Symposium on Network and Distributed System Security (NDSS)*. https://doi.org/10.14722/ndss.2021.23091

[9] Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. 2013. Density-Based Clustering Based on Hierarchical Density Estimates. In *Advances in Knowledge Discovery and Data Mining*. Springer Berlin Heidelberg, Berlin, Heidelberg, 160–172. https://doi.org/10.1007/978-3-642-37456-2_14

[10] Jacob Cohen. 1988. *Statistical power analysis for the behavioral sciences* (2 ed.). Routledge Member of the Taylor and Francis Group, New York, NY.

[11] Cookiepedia by OneTrust. 2024. Largest Database of Pre-Categorized Cookies. https://web.archive.org/web/20240524223252/https://cookiepedia.co.uk/.

[12] Nurullah Demir, Matteo Große-Kampmann, Tobias Urban, Christian Wressnegger, Thorsten Holz, and Norbert Pohlmann. 2022. Reproducibility and Replicability of Web Measurement Studies. In *Proceedings of the ACM Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. Association for Computing Machinery, New York, NY, USA, 533–544. https://doi.org/10.1145/3485447.3512214

[13] Nurullah Demir, Daniel Theis, Tobias Urban, and Norbert Pohlmann. 2022. Towards Understanding First-Party Cookie Tracking in the Field. *CoRR* abs/2202.01498 (2022). arXiv:2202.01498 https://arxiv.org/abs/2202.01498

[14] Nurullah Demir, Tobias Urban, Norbert Pohlmann, and Christian Wressnegger. 2024. A Large-Scale Study of Cookie Banner Interaction Tools and their Impact on Users' Privacy. *Proceedings on Privacy Enhancing Technologies* 2024 (2024).

[15] Nurullah Demir, Tobias Urban, Kevin Wittek, and Norbert Pohlmann. 2021. Our (in)Secure Web: Understanding Update Behavior of Websites and Its Impact on Security. In *Conference on Passive and Active Measurement (PAM)*. Springer-Verlag, Berlin, Heidelberg, 76–92. https://doi.org/10.1007/978-3-030-72582-2_5

[16] Demir, Nurullah. 2023. MultiCrawl. https://github.com/nrllh/multicrawl. Used version: MultiCrawl (v0.1.2).

[17] Christian J Dietrich and Christian Rossow. 2009. Empirical Research of IP Blacklists. In *ISSE 2008 Securing Electronic Business Processes: Highlights of the Information Security Solutions Europe 2008 Conference*. Springer, Vieweg+Teubner, Wiesbaden, 163–171.

[18] Zainul Abi Din, Panagiotis Tigas, Samuel T. King, and Benjamin Livshits. 2020. PERCIVAL: Making In-Browser Perceptual Ad Blocking Practical with Deep Learning. In *2020 USENIX Annual Technical Conference (USENIX ATC)*. USENIX Association, USA, 387–400.

[19] EasyList. 2024. EasyList. https://web.archive.org/web/20240521020759/https://easylist.to/easylist/easylist.txt.

[20] EasyList. 2024. Other Supplementary Filter Lists and EasyList Variants. https://web.archive.org/web/20240521020545/https://easylist.to/pages/other-supplementary-filter-lists-and-easylist-variants.html.

[21] EasyPrivacy. 2024. EasyPrivacy list. https://web.archive.org/web/20240521020908/https://easylist.to/easylist/easyprivacy.txt.

[22] Peter Eckersley. 2010. How unique is your web browser?. In *Proceedings on Privacy Enhancing Technologies*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1–18.

[23] Saiid El Hajj Chehade, Sandra Siby, and Carmela Troncoso. 2024. SINBAD: Saliency-informed detection of breakage caused by ad blocking. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, Los Alamitos, CA, USA, 211–211. https://doi.org/10.1109/SP54263.2024.00199

[24] Steven Englehardt and Arvind Narayanan. 2016. Online tracking: A 1-million-site measurement and analysis. In *Proceedings of ACM CCS 2016*. Association for Computing Machinery, New York, NY, USA, 1388–1401.

[25] Steven Englehardt, Dillon Reisman, Christian Eubank, Peter Zimmerman, Jonathan Mayer, Arvind Narayanan, and Edward W Felten. 2015. Cookies that give you away: The surveillance implications of web tracking. In *International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 289–299.

[26] Fanboy. 2024. Fanboy's Social Blocking List. https://web.archive.org/web/20240521020623/https://easylist.to/easylist/fanboy-social.txt.

[27] Álvaro Feal, Pelayo Vallina, Julien Gamba, Sergio Pastrana, Antonio Nappa, Oliver Hohlfeld, Narseo Vallina-Rodriguez, and Juan Tapiador. 2021. Blocklist babel: On the transparency and dynamics of open source blocklisting. *IEEE Transactions on Network and Service Management* 18, 2 (2021), 1334–1349.

[28] Roberto Gonzalez, Lili Jiang, Mohamed Ahmed, Miriam Marciel, Ruben Cuevas, Hassan Metwalley, and Saverio Niccolini. 2017. The Cookie Recipe: Untangling the Use of Cookies in the Wild. In *2017 Network Traffic Measurement and Analysis Conference (TMA)*. 1–9. https://doi.org/10.23919/TMA.2017.8002896

[29] Google Inc. 2023. BigQuery: Cloud Data Warehouse. https://web.archive.org/web/20240529092514/https://cloud.google.com/bigquery/.

[30] Google Inc. 2023. Topics API Internals: Classifier. chrome://topics-internals.

[31] Google Inc. 2024. Chrome UX Report. https://web.archive.org/web/20240521020848/https://developer.chrome.com/docs/crux.

[32] Luca Invernizzi, Kurt Thomas, Alexandros Kapravelos, Oxana Comanescu, Jean-Michel Picod, and Elie Bursztein. 2016. Cloak of Visibility: Detecting When Machines Browse a Different Web. In *2016 IEEE Symposium on Security and Privacy (SP)*. 743–758. https://doi.org/10.1109/SP.2016.50

[33] Umar Iqbal, Steven Englehardt, and Zubair Shafiq. 2021. Fingerprinting the Fingerprinters: Learning to Detect Browser Fingerprinting Behaviors. In *2021 IEEE Symposium on Security and Privacy (SP)*. 1143–1161. https://doi.org/10.1109/SP40001.2021.00017

[34] Umar Iqbal, Zubair Shafiq, and Zhiyun Qian. 2017. The ad wars: retrospective measurement and analysis of anti-adblock filter lists. In *Proceedings of the 2017 Internet Measurement Conference* (London, United Kingdom) *(IMC '17)*. Association for Computing Machinery, New York, NY, USA, 171–183. https://doi.org/10.1145/3131365.3131387

[35] Umar Iqbal, Peter Snyder, Shitong Zhu, Benjamin Livshits, Zhiyun Qian, and Zubair Shafiq. 2020. Adgraph: A graph-based approach to ad and tracker blocking. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 763–776.

[36] Paul Jaccard. 1912. THE DISTRIBUTION OF THE FLORA IN THE ALPINE ZONE.1. *New Phytologist* 11, 2 (Feb. 1912), 37–50. https://doi.org/10.1111/j.1469-8137.1912.tb05611.x

[37] Jordan Jueckstock, Shaown Sarker, Peter Snyder, Aidan Beggs, Panagiotis Papadopoulos, Matteo Varvello, Ben Livshits, and Alexandros Kapravelos. 2021. Towards Realistic and Reproducible Web Crawl Measurements. In *International Conference on World Wide Web (TheWebConf)*. Association for Computing Machinery, New York, NY, USA, 80–91. https://doi.org/10.1145/3442381.3450050

[38] justdomains. 2024. justdomains/ci - Domain Blocklists. https://github.com/justdomains/ci/tree/master Accessed: 2024-11-02.

[39] William H. Kruskal and W. Allen Wallis. 1952. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association* 47, 260 (1952), 583–621.

[40] Pierre Laperdrix, Nataliia Bielova, Benoit Baudry, and Gildas Avoine. 2020. Browser Fingerprinting: A Survey. *ACM Trans. Web* 14, 2, Article 8 (apr 2020), 33 pages. https://doi.org/10.1145/3386040

[41] Lazarev, Anton. 2024. adblock-rust. https://web.archive.org/web/20240327153321/https://github.com/brave/adblock-rust.

[42] Vector Guo Li, Matthew Dunn, Paul Pearce, Damon McCoy, Geoffrey M Voelker, and Stefan Savage. 2019. Reading the tea leaves: A comparative analysis of threat intelligence. In *28th USENIX security symposium (USENIX Security 19)*. 851–867.

[43] Georg Merzdovnik, Markus Huber, Damjan Buhov, Nick Nikiforakis, Sebastian Neuner, Martin Schmiedecker, and Edgar Weippl. 2017. Block Me If You Can: A Large-Scale Study of Tracker-Blocking Tools. In *Proceedings of the 2nd IEEE European Symposium on Security and Privacy (Euro S&P)*.

[44] r/uBlockOrigin. 2024. r/uBlockOrigin/Wiki. https://web.archive.org/web/20240514202126/https://www.reddit.com/r/uBlockOrigin/wiki/index/?rdt=45919.

[45] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. 2017. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM Trans. Database Syst.* 42, 3, Article 19 (2017), 21 pages. https://doi.org/10.1145/3068335

[46] Steve Sheng, Brad Wardman, Gary Warner, Lorrie Cranor, Jason Hong, and Chengshan Zhang. 2009. An Empirical Analysis of Phishing Blacklists. (2009).

[47] He Shuang, Lianying Zhao, and David Lie. 2024. Dumviri: Detecting Trackers and Mixed Trackers with a Breakage Detector. arXiv:2402.08031 [cs.CR] https://arxiv.org/abs/2402.08031

[48] Alexander Sjösten, Peter Snyder, Antonio Pastor, Panagiotis Papadopoulos, and Benjamin Livshits. 2020. Filter list generation for underserved regions. In *Proceedings of The Web Conference 2020*. Association for Computing Machinery, New York, NY, USA, 1682–1692.

[49] Michael Smith, Peter Snyder, Moritz Haller, Benjamin Livshits, Deian Stefan, and Hamed Haddadi. 2022. Blocked or Broken? Automatically Detecting When Privacy Interventions Break Websites. *Proceedings on Privacy Enhancing Technologies* 2022, 4 (2022). https://doi.org/10.56553/popets-2022-0096

[50] Peter Snyder, Antoine Vastel, and Ben Livshits. 2020. Who filters the filters: Understanding the growth, usefulness and efficiency of crowdsourced ad blocking. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 4, 2 (2020), 1–24.

[51] Jannick Kirk Sørensen and Sokol Kosta. 2019. Before and After GDPR: The Changes in Third Party Presence at Public and Private European Websites. In *Proceedings of the 2019 International Conference on World Wide Web (WWW)*. Association for Computing Machinery, New York, New York, USA, 11 pages.

[52] Kurt Thomas, Chris Grier, Justin Ma, Vern Paxson, and Dawn Song. 2011. Design and Evaluation of a Real-Time URL Spam Filtering Service. In *2011 IEEE symposium on security and privacy*. IEEE, 447–462.

[53] Tobias Urban, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. 2020. Beyond the Front Page: Measuring Third Party Dynamics in the Field. In *International Conference on World Wide Web (TheWebConf)*. Association for Computing Machinery, New York, NY, USA, 1275–1286. https://doi.org/10.1145/3366423.3380203

[54] Tobias Urban, Dennis Tatang, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. 2018. The Unwanted Sharing Economy: An Analysis of Cookie Syncing and User Transparency under GDPR. *CoRR* abs/1811.08660 (2018). arXiv:1811.08660 http://arxiv.org/abs/1811.08660

[55] Tobias Urban, Dennis Tatang, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. 2020. Measuring the Impact of the GDPR on Data Sharing in Ad Networks. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security* (Taipei, Taiwan) *(ASIA CCS)*. Association for Computing Machinery, New York, NY, USA, 222–235. https://doi.org/10.1145/3320269.3372194

[56] Gerry Wan, Liz Izhikevich, David Adrian, Katsunari Yoshioka, Ralph Holz, Christian Rossow, and Zakir Durumeric. 2020. On the origin of scanning: The impact of location on internet-wide scans. In *Proceedings of the ACM Internet Measurement Conference*. Association for Computing Machinery, New York, NY, USA, 662–679.

[57] Jerrold H. Zar. 2014. *Spearman Rank Correlation: Overview.* John Wiley & Sons, Ltd, Weinheim, Berlin, Chapter Encyclopedia of Biostatistics. https://doi.org/10.1002/9781118445112.stat05964 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118445112.stat05964

A Availability of Data & Code

To foster future research, we release our code, queries for the entire data processing pipeline and evaluation, and other supplementary information openly online at: https://github.com/internet-sicherheit/Understanding-Regional-Filter-Lists-Efficacy-and-Impact-

B Number of Requests Exclusively Blocked by Different Filter Lists

Table 3 provides an overview of the number of requests blocked exclusively by one list in Scenario 1 (see Section 3.1). Exclusively means that no other list would have blocked the request of interest.

C Blocked eTLDs by Rule

In Fig. 13, we provide an overview of the number of blocked eTLDs by a distinct rule for each filter list. The more right on the x-axis a bar is the more eTLDs a specific rule blocks. Rules that block many eTLDs are candidates for malformed rules, as it is likely that they block too many requests. Most rules block only one eTLD.

	ID	AE	CN	DE	FR	IS	IN	JP	NO	US
0	AE.AE	7,046	56,338	390	126,609	11,172	34,889	1,021,418	59,526	751,295
1	CN.CN	2,449	183,623	94	60,876	1,526	20,913	534,296	38,208	395,284
2	DE.DE	40,314	68,471	13,672	77,459	14,796	25,743	520,955	234,836	392,240
3	FR.FR	10,021	53,625	1,123	105,255	9,438	25,358	586,992	84,719	342,908
4	IS.IS	3,104	61,164	310	197,590	30,446	53,339	1,169,997	64,373	607,204
5	IN.IN	48	33,278	307	100,599	6,597	36,376	942,548	34,885	703,249
6	JP.JP	3,883	144,733	65	172,795	7,249	51,490	2,324,275	34,380	682,159
7	NO.NO	5,148	48,783	764	110,694	20,105	33,862	693,689	148,852	334,142
8	US.US	19,348	63,508	769	214,554	14,259	26,249	1,607,728	188,786	2,169,025

Table 3: Number of requests exclusively blocked in Scenario 1 by each analyzed list.

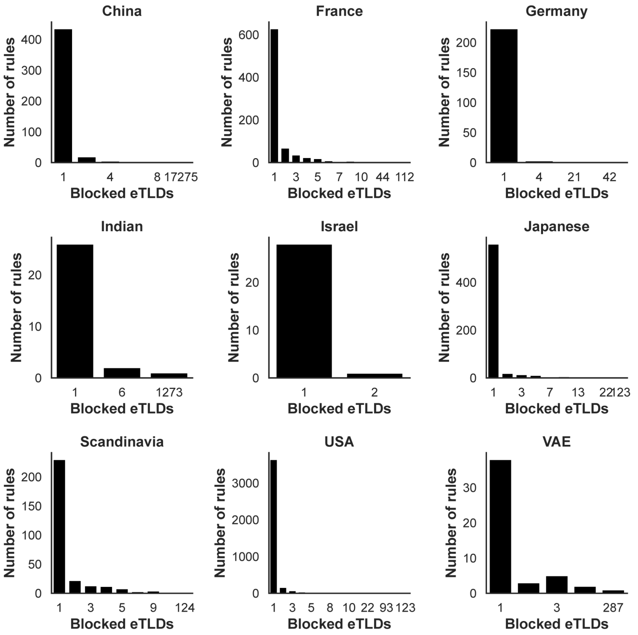

Figure 13: Overview of how many eTLDs were blocked by one rule.

D Runtime analysis

Figure 14 provides an overview of the performance measured from the generated lists. The figure shows that the number of rules used directly impacts runtime and memory consumption. It can be seen that both the memory consumption and the runtime increase almost linearly with the number of rules used. However, the efficiency of the individual rules used decreases linearly. This means that adding more rules above a certain database size does not have a major effect on the number of blocked URLs. Still, the negative effects (longer runtime and higher memory consumption) continue to increase linearly.

E Commitments to the Filter Lists

Figure 15 shows the number of added and removed rules for 23 open-source filter lists over a 12-month period. The results highlight

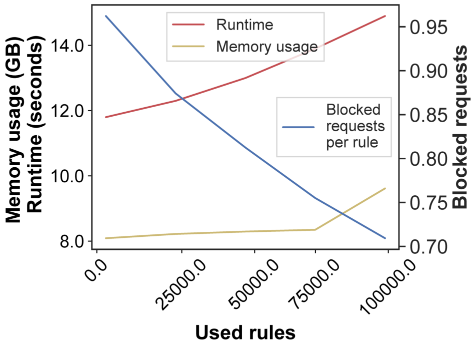

Figure 14: Runtime and memory analyses of different numbers of used rules

significant differences in update frequency between repositories. While some repositories, such as Bulgarian, exhibit minimal activity (e.g., 5 out of 12 months with fewer than 100 entries), others, like USA, consistently show over 5,000 entries per month. Additionally, filter lists that block more tracking requests demonstrate sustained activity levels (see Section 3.2). For nearly all repositories and months, the number of added rules exceeds the removed rules, suggesting a continuous growth in the size of these filter lists.

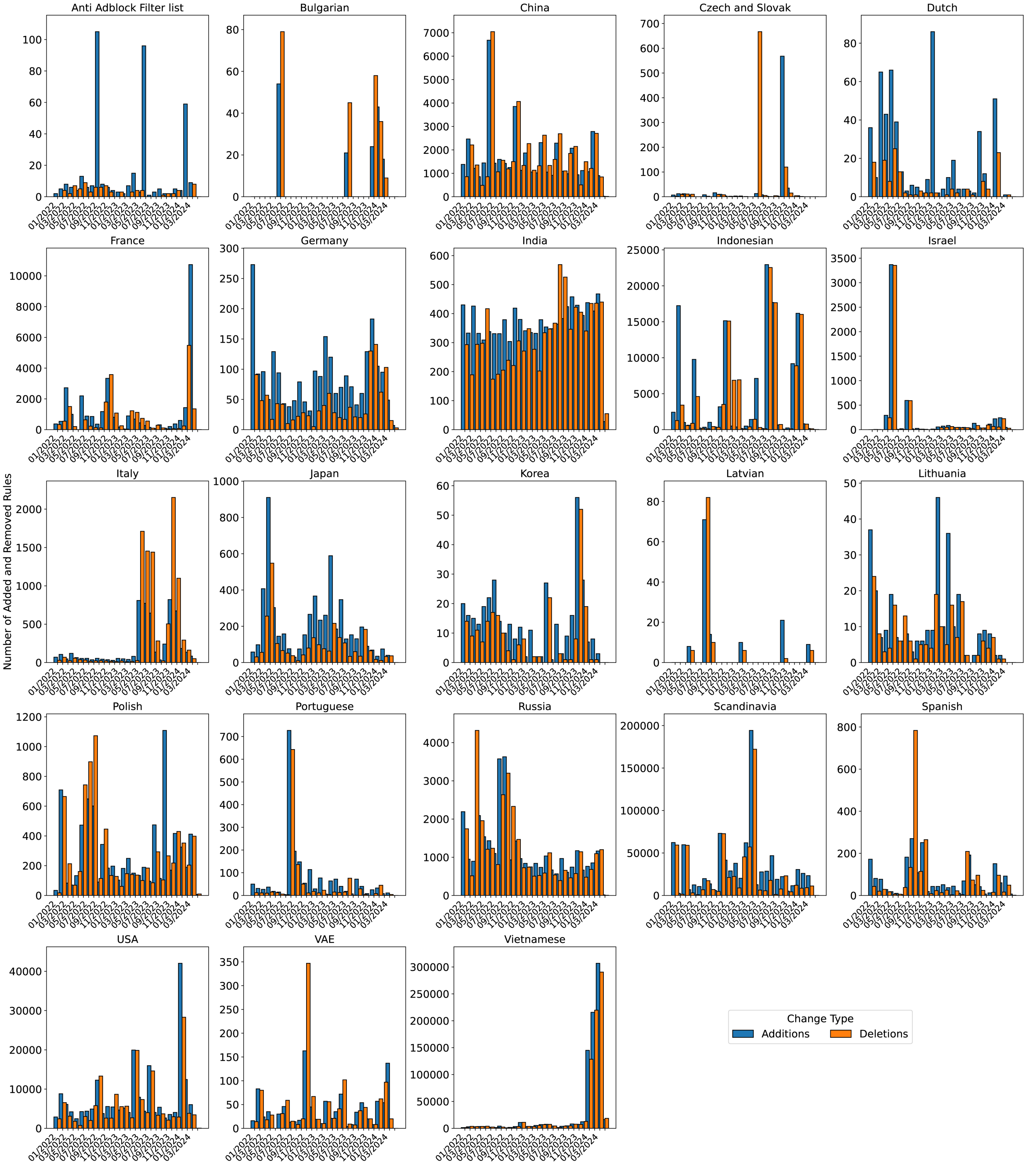

Figure 15: Number of added and removed rules for 23 regional filter lists.