Measuring the Privacy vs. Compatibility Trade-off in Preventing Third-Party Stateful Tracking

Jordan Jueckstock
North Carolina State University
jjuecks@ncsu.edu

Peter Snyder
Brave Software
pes@brave.com

Shaown Sarker
North Carolina State University
ssarker@ncsu.edu

Alexandros Kapravelos
North Carolina State University
akaprav@ncsu.edu

Benjamin Livshits
Brave Software &
Imperial College London
b.livshits@imperial.ac.uk

ABSTRACT

Despite much web privacy research on sophisticated tracking techniques (e.g., fingerprinting, cache collusion, bounce tracking), most tracking on the web is still done by transmitting stored identifiers across site boundaries. "Stateful" tracking is not a bug but a misfeature of classical browser storage policies: per-site storage is shared across all visits, from both first- and third-party (i.e., embedded in other sites) context, enabling the most pervasive forms of online tracking.

In response, some browser vendors have implemented alternate, privacy-preserving storage policies, especially for third-party site context. However, such changes risk breaking websites that presume the traditional model of non-partitioned third-party storage. Such breakage can itself harm web privacy: browsers that frustrate user expectations will be abandoned for more permissive, privacy-harming browsers, cementing rather than disrupting the *status quo*.

Our work improves the state of web privacy by measuring the privacy *vs.* compatibility trade-offs of representative third-party storage policies, with the end-goal of enabling design of browsers that are both compatible and privacy respecting. Our contributions include web-scale measurements of page behaviors under multiple third-party storage policies representative of those deployed in several production browsers. We define metrics for measuring aggregate effects on web privacy and compatibility, including a novel system for programmatically estimating aggregate website breakage under different policies. We find that making third-party storage partitioned by first-party, and lifetimes by site-session achieves the best privacy and compatibility trade-off. We provide complete datasets and implementations for our measurements and tools.

CCS CONCEPTS

• **Security and privacy → Privacy protections**; • **Information systems → Browsers**; *Web mining*.

Woodstock '18, June 03–05, 2018, Woodstock, NY
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06...$15.00
https://doi.org/10.1145/1122445.1122456

KEYWORDS

browsers, cookies, compatibility, breakage, tracking, privacy, measurement, crawling

ACM Reference Format:
Jordan Jueckstock, Peter Snyder, Shaown Sarker, Alexandros Kapravelos, and Benjamin Livshits. 2018. Measuring the Privacy vs. Compatibility Trade-off in Preventing Third-Party Stateful Tracking. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Web trackers use many techniques to track users and violate privacy on the web. Typical tracker practice combines stateful tracking (i.e., storing and transmitting unique identifiers in the browser) and stateless tracking, or fingerprinting (i.e., attempting to uniquely identify a browser based on distinctive browser, operating system, and hardware characteristics).

Though much recent privacy work has focused on stateless tracking (i.e., fingerprinting), there is cause to believe that the majority of tracking is still done using traditional stateful methods. Supporting evidence includes the adtech uproar over Google's recent announcement [1] to stop sending cookies (only one of many ways of storing identifiers) to third-parties in the future, prior research demonstrating the popularity of storage-based tracking [11, 13, 30, 31, 38, 41], and expert insight from browser developers.

While the privacy community has had some success in designing defenses to stateless tracking that protect users without breaking benign, user-serving page functionality[20, 26], researchers, industry, and activists have been less successful in designing practical, robust defenses against web-scale stateful third-party tracking. Blocking the transmission of cookies to third-parties for sub-resource requests is a welcome emerging development, but it does not provide protection against intentional stateful third-party tracking by JavaScript code with access to persistent cookies[1], `localStorage`[2], `indexDB`[3], or other JavaScript accessible storage methods (collectively, "DOM storage").

[1] For completeness, we note that this isn't completely true, and that `HttpOnly` cookies cannot be accessed from JavaScript. But since `HttpOnly` doesn't provide protection against intentional tracking (since such trackers could just omit the `HttpOnly` instruction), we don't consider `HttpOnly` further in this work, and omit it from further discussion for concision.

[2] https://html.spec.whatwg.org/multipage/webstorage.html#the-localstorage-attribute

[3] https://www.w3.org/TR/IndexedDB-2/

Historical attempts at comprehensive stateful tracking protections have struggled to balance privacy with compatibility. Filter list approaches which block third-party storage only for "known" trackers maintain good compatibility but cannot protect against new or stealthy trackers. Aggressive global blocking of third-party storage provides excellent privacy protections but breaks an unacceptably high amount of benign web content. Recent innovations in stateful tracking protection, exemplified by the latest iterations of Safari ITP and Brave Shields, suggest a possible emerging solution to the traditional privacy/compatibility dilemma: hybrid policies combining partitioned organization and ephemeral lifetime of third-party storage. However, evaluation of this approach's effectiveness hinges on web-scale measurement not only of privacy protections (fairly well understood by now) but also estimated web breakage/compatibility caused by storage policies (an open problem at scale).

This work directly addresses this evaluation question by implementing multiple simplified, representative third-party storage policies in a heavily-instrumented browser (Brave PageGraph [9, 18]), collecting comprehensive page-behavior metrics under each policy during a parallel crawl across top sites, and evaluating key privacy and novel compatibility/breakage indicators among compared policies. The policies tested include: **permissive** (third-party storage allowed globally, best compatibility); **blocking** (third-party storage blocked globally, worst compatibility), **site-keyed** (third-party storage partitioned by first-party but persistent), and **page-length** (third-party storage partitioned by and limited to the lifespan of the top-level page/document). Our privacy evaluation considers the comparative prevalence of potentially-identifying cookie values seen in storage and quantifies how many third-party domains had the ability to track our browsers across different first-party sites (cross-site trackability) and across repeat visits to the same first-party site (cross-time trackability). Our novel compatibility evaluation exploits the rich instrumentation of PageGraph to construct "behavior sets" and to compare their similarity between each alternate policy and the known-good baseline (no blocking). We complement these automated experiments with qualitative manual assessments of a random sample of visited pages to assess compatibility through human eyes.

Our privacy results confirm prior experience, and our compatibility results support the view that hybrid ephemeral third-party storage is emerging as a potential solution to the stateful tracking problem. All non-permissive third-party storage policies provided significant cross-site tracking protection, and page-length provided measurably better cross-time tracking protection than site-keyed. Page-length and site-keyed performed very similarly on our compatibility metrics, both showing much stronger behavioral similarity to the permissive baseline than did blocking, the "known worst-case" baseline. Our manual compatibility assessment showed generally low rates of user-perceived breakage across all tested policies, suggesting that the emergence of more effective stateful tracking protections is prompting a shift away from dependence on third-party storage for essential functionality.

This work makes the following concrete contributions:

(1) Open source, PageGraph-based, Puppeteer-drived instrumentation system allowing automated privacy and compatibility-estimate measurements across the web under multiple third-party storage policy implementations representative of both deployed and proposed storage systems.
(2) Design and implementation of metrics to programmatically evaluate the privacy and compatibility implications of privacy interventions, including a novel system for comparing, in aggregate, the compatibility effects of different privacy interventions.
(3) The results of a web-scale evaluation of how third-party storage policies inspired by those deployed in popular browsers compare in terms of privacy and compatibility benefit.
(4) A complementary manual, qualitative evaluation of the compatibility impacts of different privacy interventions.

2 BACKGROUND & MOTIVATION

2.1 Same-Origin Policy & Storage Basics

Browsers isolate storage (e.g., cookies, `localStorage`, `indexDB`) according to the Same-Origin Policy (SOP) [4]. Though the SOP is complex and inconsistent in practice [35], the SOP is relatively simple in regards to browser storage policies .The SOP says that scripts can access cookies and DOM storage (e.g., `localStorage`) only for their execution origin, and HTTP requests store and transmit cookies only for their destination origin.

When loading a website, the **first-party** is the "site" portion of the top level document. This is the eTLD+1 of the URL shown in the navigation bar of the browser. Any sub-resources or sub-documents included in the page are considered first-party if they're fetched from the same eTLD+1 as the top level document. **Third-parties** are any site not equal to the top-level document.

The storage values a script can access is determined by the "site" of the frame that script is executing in, *not* the site the script was fetched from. For example, if a page from origin A includes a script from origin B, the script is a third-party script, but has access to the first-parties (i.e., site A's) storage.

2.2 Online Tracking

We use the term "tracking" to refer to a third-party re-identifying a visitor across multiple site visits which are otherwise unrelated or associated. Such tracking can be *cross-site* (i.e., a third-party can link a visitor's activities across first-party sites) or *cross-time* (i.e., a third-party can identify the same visitor returning to the same first-party site across sessions). While many techniques for tracking have been studied, we focus exclusively on stateful tracking techniques, such as cookies. At root, these rely on a third-party site being able to access persistent state in different contexts, and using the persistently stored state to link (conceptually) unrelated behavior. As we will discuss at length later, approaches for preventing stateful tracking involve either preventing third-parties from storing values at all, providing third-parties with different storage context when embedded in different contexts, or combinations of the two.

2.3 Threat Model

Here we present a simple threat model defining the scope boundaries for our proposed storage policy improvements. It provides

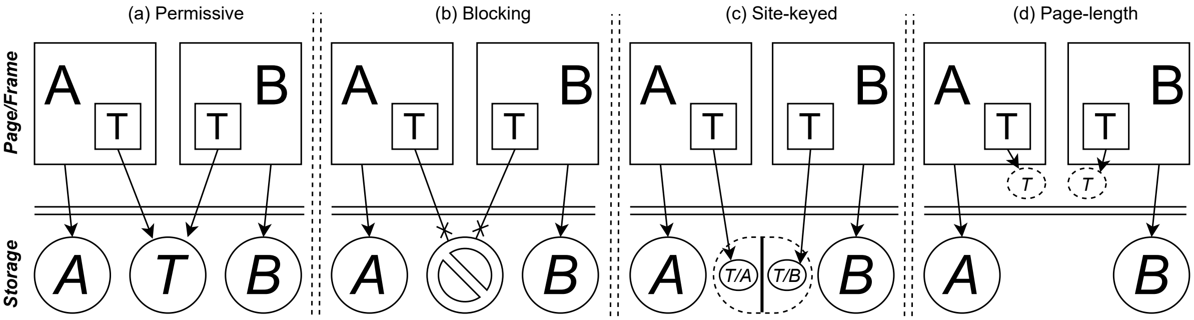

Figure 1: Third-party storage (a) fully allowed, (b) fully blocked, (c) partitioned by first-party context, and (d) scoped to hosting page life time (our proposal). ***A, B, & T are distinct domains; T is embedded as a third-party within A & B.***

useful criteria for evaluating both deployed and experimental policy alternatives.

Actors. We exclusively consider threats originating from third-party content providers engaged in user tracking. While we do consider the possibility of first-party errors or carelessness amplifying the threat posed by third-party actors, we consider *active collusion* between third-parties and first-parties (e.g., the disturbing new tactic of cloaking third-party content behind first-party CNAME DNS records) to be out of scope.

Mechanisms. Our focus is on *stateful* tracking, though we consider instances where stateless tracking mechanisms may be used to bridge or synchronize stateful sessions. We consider the threat of fully stateless tracking (i.e., a universal, per-user fingerprint needing no state transfer) to be out of scope. This choice is deliberate: we believe stateful tracking is where browsers are most lacking practical, robust, compatible defenses. While significant research has gone into building web-compatible defenses against stateless tracking (e.g., [20, 26]), the existing techniques for preventing stateful third-party tracking are either incomplete (i.e., they still allow significant privacy harm to occur) or incompatible (i.e., they break a significant number of websites).

Threats. The primary threat considered is classic cross-site user tracking as enabled by traditional unified, persistent third-party storage. We do not believe it is controversial to consider such tracking, which amounts to disclosure of a user's browsing history, to be an undesirable breach of personal privacy. However, there exist additional subtle *cross-time* tracking concerns raised by persistent third-party storage even when it is partitioned by first-party context (a relatively common proposed defense mechanism; see Figure 1 and Section 2.4). Such third-party tracking of return-visit activity within a single first-party context can enable or amplify attacks like *session linking* or *cookie syncing*.

By *cookie syncing* we refer not to cross-vendor syncing [30] but to the possibility of cross-site syncing enabled by browser implementation flaws. E.g., consider a scenario in which a first-party site embeds a third-party frame. One week later, that frame gains the ability to cookie sync (e.g., a new browser feature adds enough entropy to fingerprint). But the next day, that ability is lost (e.g., a high-priority browser update removes the privacy leak). Effectively, this disaster scenario temporarily neuters any attempt to partition third-party stored state by first-party context. The impact on privacy is determined by how much longitudinal data is available in third-party storage to by synced across first-party boundaries. In the example scenario, it is *one week* of browsing data with stable third-party storage and *one day* of data with only ephemeral storage. What we are considering a threat, then, is not the possibility of cookie-syncing itself, but rather the scale of damage it could cause. Our concern is defense in depth, just as cryptographers implementing *perfect forward secrecy* do so not because they expect frequent key exposure but because they wish to mitigate the impact of its hopefully unlikely occurrence.

By *session linking* we mean third-parties exploiting any flaw that allows inference of first-party login state to link two or more login identities that the user intended to keep disassociated. Robust SOP enforcement should prevent such inference, but loopholes (e.g., *Referer* leaks, *postMessage* mishandling) have been and probably will continue to be found and exploited in the wild. If such a vulnerability is ever found in a sensitive first-party site (e.g., a web mail or personal finance portal), the persistence of third-party state across first-party session boundaries opens up the possibility of a session linking attack by any third-party content embedded in that site.

Finally, we consider breakage of essential web content to be a threat, too. *Availability* has always been a critical component of information security. If a storage policy prevents all cross-site and cross-time third-party tracking *perfectly* but breaks any significant amount of the web in the process, users will not tolerate the breakage and will revert to policies that are vulnerable to one or more of the other threats described above.

2.4 Deployed Stateful Tracking Defenses

With the exception of Google Chrome, all of today's major web browsers implement proactive user tracking defenses. These defenses illustrate a range of possible trade-offs between privacy and compatibility. All of them provide some degree of **cross-site** protection, preventing third-parties from using stored identifiers to link browsing behavior across first-party sites. More aggressive defenses also attempt **cross-time** protection, preventing third-parties from using stored identifiers to link browsing behavior across visits to the same first-party site. Note that these summaries do not cover tracking defenses unrelated to third-party storage (e.g., third-party content blocking, first-party storage lifetime restrictions, "bounce" tracking defenses, fingerprinting defenses, etc.).

User tracking defenses can be decomposed into two independent aspects: mechanism (i.e., how storage access is affected) and policy (i.e., for what actors, under what conditions). Mechanisms include altering the lifetime of third-party storage, partitioning it by first-party site context, or even blocking it entirely. Such defense mechanisms can be applied to all third-party storage or to a restricted subset of storage mechanisms (e.g., cookies *vs.* local storage, HTTP Cookie headers *vs.* JS code). Defense policies may be global, for all third-party domains; or selective, for third-party domains classified as trackers based on *a priori* filter lists or dynamic behavior analysis and scoring.

Microsoft Edge and classic Mozilla Firefox defenses have selectively blocked third-party storage using Disconnect [3] to identify known trackers, resulting in cross-site and -time protection only as complete as the filter lists used. Firefox has since introduced[4] a strict "Total Cookie Protection" opt-in mode that partitions third-party storage by first-party site context globally, providing comprehensive cross-site protection.

Brave traditionally blocked all third-party storage globally, providing excellent cross-site and -time protection at the cost of reduced compatibility. Typical incompatibility issues for classic Brave Shields were failures of stateful third-party widgets (e.g., a stock history graph, or an interactive programming language interpreter window) to load properly without third-party session cookies or similar being accessible. Recently, Brave has moved to an ephemeral-storage mechanism in which third-party storage is partitioned by and prevented from outliving individual browsing sessions; this approach retains most of the cross-site and -time protections of blocking with improved compatibility.

Apple Safari's "Intelligent Tracking Prevention" (ITP) defenses have evolved significantly over time, shifting from selective enforcement policy guided by local machine-learning of tracker identities to global enforcement of a hybrid blocking/partitioning/lifespan-shortening mechanism. While cross-site protection with good compatibility appears to be Apple's principle goal, ITP's most recent iterations (e.g., flushing what little partitioned third-party storage is allowed every browser restart) provides a good measure of cross-time protection as well.

3 METHODOLOGY

We evaluate the privacy *vs* compatibility trade-offs illustrated by four real or representative third-party storage policies by comparing their tracking and compatibility performance during automated, stateful crawls of popular web sites.

3.1 Stateful Crawl Methodology

3.1.1 Target URLs. We generated a seed list of URLs to visit in parallel using a stateless *pilot crawl* of the Tranco 1k sites [32]. To achieve depth and representative sampling of web content, we must explore more than just the "landing page" of each site. But each of our 8 parallel crawls must visit the same sequence of page URLs to produce comparable results. Coordinating the link spidering and selection process across parallel crawls introduces needless engineering complexity. Our solution was to perform a stateless pilot crawl using stock Chromium to visit the Tranco 1k sites' landing pages and spider three links deep into the site structure. This approach, using the 2020-08-13 Tranco list snapshot, produced 3,419 total deduplicated page URLs to visit.

3.1.2 Policy Variants. We collect data using four distinct policy variants.

Permissive: Allows all forms of third-party storage, as per Figure 1a. Stock Chrome behavior. Presumed to cause no breakage.

Blocking: Blocks all forms of third-party storage, as per Figure 1b. Treats access as no-op. Known to cause some site breakage in the wild; e.g., when third-party frames are unable to maintain session state across multiple requests.

Site-keyed: Partitions persistent third-party storage by first-party eTLD+1, as per Figure 1c. Similar to elements of classic Safari ITP and Firefox's newly-announced Total Cookie Protection. Expected to match compatibility and cross-site tracking of page-length. Included to estimate residual potential for cross-time tracking under partitioned storage within a given time window of persistence (in our case, for the entire experiment).

Page-length: Isolates third-party storage in ephemeral partitions, as per Figure 1d. Similar to recent Brave and Safari ITP policies. Expected to show compatibility scores in line with the permissive baseline and tracking protection scores in line with blocking.

It should be stressed that these experimental policies, despite derivation from and obvious relation to deployed real-world policies, are intended as comparison points between archetypal approaches, not between specific browser implementations.

3.1.3 Crawl Execution. We deployed two instances of each tested policy to verify behavioral consistency and provide similarity-score baselines (see Section 3.2.2). The crawlers maintained independent, persistent user profiles for each policy instance to maintain state across all sequential page visits. The main crawl was repeated once (two iterations total) to provide data on cross-time tracking across return visits. All crawls were performed in parallel and simultaneously from a single network vantage point. Each page visit was performed in a freshly launched, non-headless (i.e., rendering to the *Xvfb* headless display server) browser instance. Navigation was allowed to time out after 30 seconds. Assuming no navigation timeout, our crawlers waited for 30 seconds after the `DOMcontentloaded` event (i.e., main document fetched and parsed but subresources not fully loaded yet) before tearing down the browser instance. No simulated user interactions were attempted.

3.1.4 PageGraph Instrumentation. We use PageGraph, an instrumentation system built into an experimental branch of Brave, to record internal page behaviors. PageGraph patches the V8 JS engine and the Blink HTML rendering engine to capture and annotate a graph of each HTML document's DOM structure and the events that constructed and modified it. Nodes represent entities such as DOM elements, scripts, HTTP resources, storage mechanisms, and a selective subset of builtin and DOM-provided JavaScript APIs. Edges represent relationships between nodes such as DOM structures and script interactions with DOM elements, DOM events, JavaScript APIs, and HTTP requests. The set of non-structural edges in each of these graphs constitute the dynamic behaviors of the originating page. Behavioral-edge-set similarity can be quantified using Jaccard index scores to provide a useful proxy for behavioral compatibility among compared storage policies.

[4] https://hacks.mozilla.org/2021/02/introducing-state-partitioning/

3.2 Primary Evaluation Methodology

We evaluate our policies' privacy and compatibility performance using full-scale quantitative stateful tracking metrics, full-scale quantitative site behavior similarity metrics, and randomly-sampled qualitative assessment of site breakage. All quantitative metrics analysis focuses on third-party frames not flagged as advertising content. First-party frames are loaded from the same eTLD+1 as the main page URL (per the Public Suffix List [5]); all other frames are third-party. Classification of ads relies on the community-maintained EasyList [2]. The exclusion of first-party and advertising content eliminates noise from our evaluation: first-party storage is not affected by our experimental policy changes, and advertising content is known to change frequently.

3.2.1 *Quantitative Privacy Assessments.*

Tracking Potential. The central metric we use to quantify potential for stateful cross-site and cross-time tracking by third-parties is the *potentially identifying cookie flow* (PICF). A *cookie flow* is the combination of an HTTP cookie and a third-party eTLD+1 receiving that cookie. We consider cookie flows *potentially identifying* when the values are at least eight (8) characters long and are **globally unique** to a single browser profile during our stateful crawls. There are other forms of third-party storage available (e.g., local storage), and other channels by which identifying tokens can be transmitted to third-parties (e.g., custom HTTP headers, query string parameters). But we use cookies as our representative measure of stateful tracking because they are unambiguous in structure, ubiquitous as tracking IDs, and essentially unrestricted by stock Chrome, our baseline. (Both our page-length storage and site-keyed implementations apply their storage policies to **all** forms of third-party storage, not just cookies.)

Cross-Site Tracking. Identical PICFs seen across multiple distinct top-level sites visited represent potential for cross-site tracking by the associated third-party domain. We aggregate cross-site PICFs to count the total number of top-level sites across which each distinct third-party domain seen could have tracked our crawler profiles, giving us summary scores of "cross-site trackability" by which to compare all our storage policies. These scores can be visualized using cumulative sum curves, as shown in Section 4.2.

Cross-Time Tracking. PICFs seen on a given top-level site across multiple pages/crawls represent potential for cross-time, or visit-to-visit, tracking by a given third-party domain. We aggregate cross-time PICFs to count the total number of third-party domains which could have tracked our crawler profiles for each distinct top-level site domain visited, giving us summary scores of "cross-time trackability" by which to compare all our storage policies. These scores can be visualized using cumulative sum curves, as shown in Section 4.3.

3.2.2 *Quantitative Compatibility Assessment.*

We assess site compatibility across storage policies using a quantifiable proxy measure: similarity of internal page behaviors as reported by PageGraph. Our insight is to presume no storage-based breakage for permissive profiles and some unknown (but non-zero) amount of breakage on blocking profiles. If alternative policy (e.g., page-length storage) profiles produce content behaviors more similar to the permissive baseline than do the blocking profiles, then the alternate policy is less likely than blocking to cause breakage.

We model and compare content behaviors using the set of non-structural (i.e., action or event) edges in PageGraph representations of relevant frames. Similarity between edge sets can be measured using the Jaccard index: $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$. Index scores range from 0 (no intersection) to 1 (equality). We consider the score undefined when both sets are empty.

We compare content behaviors across identical frames loaded on identical pages across all tested policies. Frames and pages are identified and matched by full URL. The similarity score of the two permissive profiles provides the compatibility baseline: the presumed best-possible similarity score for that frame/page instance. The other profiles are each compared with a single permissive profile to provide similarity scores to compare against the baseline. The cumulative sum of all frame/page instance similarity scores for each profile can be visualized to show which policies track closest to the baseline across all visited pages (see Section 4.4).

We optimized the set of PageGraph node types included in our behavioral sets to maximize the distance between blocking policy scores and the permissive baseline score. Our intuition is that the baseline score provides a threshold of "reasonable" behavioral differences between two different instances of the same content loaded in different browsers at about the same time. The farther away from this baseline a policy scores, the greater the likelihood of unreasonable, or breaking, differences in behavior.

We identified 11 PageGraph node types relevant to behavioral analysis, a set small enough to be amenable to brute force optimization across its power set. Optimization relied on a random sample of 100 frame/site instances extracted from a preliminary full-scale crawl dataset, whose unoptimized similarity curves matched those of the entire data set, indicating a representative sampling. On this data subset we tested the blocking separation from the permissive baseline for every subset of relevant PageGraph node types. The results confirmed our intuition that the least helpful node types were structural elements like HTML elements and DOM text blocks; less intuitively, they also showed that PageGraph's set of instrumented DOM manipulation JavaScript APIs was similarly unhelpful. The final optimal node type set comprised scripts and PageGraph's selected JavaScript builtin APIs (e.g., date functions), HTTP resources, frame structures (DOM roots and frame-owning elements), and storage mechanisms (cookie jars, local and session storage buckets). Only edges (i.e., behaviors) linking these node types are included in the behavior similarity results presented in Section 4.4.

3.2.3 *Qualitative Compatibility Assessment.*

We further augment our quantitative assessment of site compatibility with blinded multi-grader manual analysis for website breakages within a random sample of sites loading popular third-party content. To select the URLs for this, we sorted third-party, non-ad-blocked frame URLs within our crawl dataset by the harmonic mean of the number of pages embedding that frame and the number of third-party cookies set for the frame's eTLD+1. This metric is higher for frames which appear on a large number of sites and have access to a large number of cookies. We selected the top 10 frame URLs with distinct eTLD+1s while filtering out frames appearing only on non-English sites, and frames without a content type of HTML or JavaScript.

We randomly selected 10 candidate page URLs for each frame URL from the prior step, resulting in a total of 100 candidate URLs.

We adopted a holistic approach to evaluate breakage rather than simply observing the behavior of the target frame, since a number of frames did not have real-estate presence on the webpage.

Our grading methodology is derived from a similar experiment by Snyder *et al.* [37]. We had two graders evaluate the policy variants in Section 3.1.2 for each candidate URL. We recruited five graders, each with background in web security. This resulted in each grader being assigned two blocks of 20 URLS, ensuring no block pair is graded by the same grader pair. The graders would visit a URL first with a permissive profile, the Chrome default. This visit is our *control* visit, followed by a visit to the same URL with each of the site-keyed, page-length, and blocking profiles. Every visit was with a fresh browser profile to ensure stateless browsing between tests/visits. Subsequent visits to the candidate URL after the control visit were randomly coded to eliminate grader bias. Graders were further instructed grade no more than 10 URLs in a single session to avoid fatigue.

In our holistic approach, each grader performed as many interactive actions on the URL within one minute, the average dwelling time for a typical web-user on a website [23]. Each grader followed a checklist of tasks to perform on the site (Appendix B).

After the visit to the URL, The graders scored each coded profile visit a score of **1** if the visit did not have any perceptible deviations from the control; **2** if there were some deviations from the control visit, but without any hindrance to their visiting experience or the tasks attempted on the site; and **3** if the visit had significant deviations from the control, preventing the graders from replicating their control visit activities.

Due to the highly subjective nature of the evaluation scheme, our graders evaluated the candidate URLs independently, unaware of the other grader's scores. Our graders had a high agreement percentage (94.67%). We also computed the Cohen's Kappa inter-rater reliability statistic [16] as 0.64, showing statistically substantial agreement between our graders [25]. We present the results of our manual evaluation in Section 4.5.

4 RESULTS

4.1 Stateful Crawl Statistics

Our stateful web crawls ran from September 12-16 2020 on a single Linux virtual machine (40 VCPUs, 100GiB RAM). Combined, the crawls visited 27,352 total pages using 8 user profiles and produced 280,219 PageGraph files (405 GB).

Error rates (Appendix A, Figure 5) were acceptable if somewhat amplified by PageGraph internal consistency assertion failures. Errors in this case refer not to page breakage but to hard failures of the crawl itself, such as a network timeout or browser crash. PageGraph's instrumentation is expansive and tracks complex interactions between JavaScript execution, DOM manipulation, and network traffic. Whenever unexpected corner cases (or bugs) prevent it from establishing unambiguous context for an event or activity, PageGraph logs the issue and terminates the browser rather than recording unreliable data.

4.2 Privacy: Cross-Site Tracking Potential

Page-length storage eliminates stateful cross-site tracking as effectively as does blocking, as seen in Figure 2. The cumulative sum curves show the aggregate counts of sites across which third-parties could track users under different policies, calculated using the tracking-potential heuristics described in Section 3.2.1. Page-length, site-keyed, and blocking policies are roughly equal at preventing stateful cross-site tracking. This result is logical and unsurprising: if third-party storage is not available (or is partitioned by first-party site, or is strictly ephemeral), it cannot be used to pass identifying state across site boundaries.

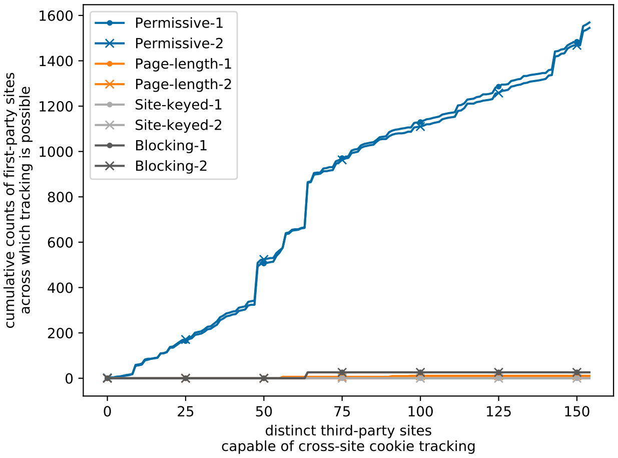

Figure 2: Of our tested policies, all but permissive essentially eliminated stateful cross-site tracking potential.

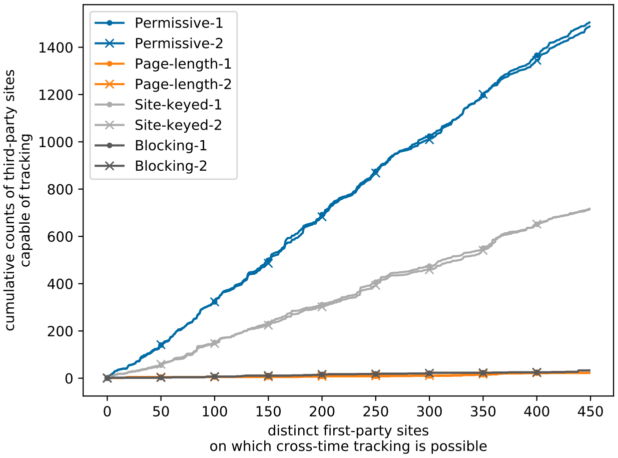

Figure 3: Our page-length policy significantly outperforms both permissive and site-keyed policies at reducing cross-time tracking potential.

4.3 Privacy: Cross-Time Tracking Potential

Page-length storage also eliminates stateful cross-time tracking as effectively as does full third-party storage blocking, a significant improvement over site-keyed storage (Figure 3). These curves show cumulative counts of third-parties which could longitudinally track

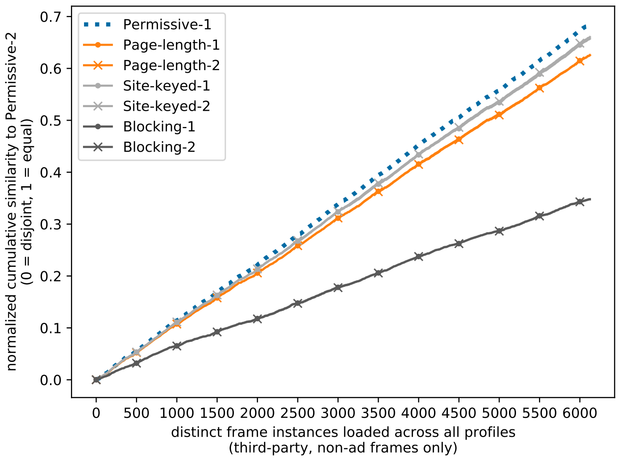

Figure 4: Our page-length policy produces page behaviors within third-party frames much closer to the permissive baseline than does the breakage-prone blocking policy.

return visitors across the Tranco 1k sites, as described in Section 3.2.1. Unsurprisingly, permissive policy allows the most cross-time tracking, as cross-site tracking ability implies cross-time tracking ability. If persistent third-party storage, even when partitioned by first-party site context, is still accessible on repeat visits, cross-time tracking is possible. Thus, page-length and blocking policies both provide stronger cross-time tracking protection than site-keyed policy can.

We additionally considered *local storage* as a medium for stateful cross-time trackability, to better consider the cross-time tracking vulnerability of defenses which block third-party cookies but allow partitioned third-party local storage. Here we apply our PICF extraction workflow to values written-to and read-from local storage in third-party frame context, dubbed *potentially identifying token sources* (PITS). Such tokens are not automatically transmitted to third-party domains as are cookies, but their presence identifies persistent third-party state that could be synced to a remote domain via XHR or similar using JavaScript code at will. We found that local storage PITS are significantly less frequent than PICFs across our dataset globally, but they are about equally common under site-keyed as they are under permissive, given an experiment-long time window of partitioned storage persistence (Appendix A, Figure 6). We conclude that strong constraints even on partitioned third-party storage lifetime (i.e., page-length rather than browser-lifetime length) are a good first-principle approach for eliminating cross-time tracking.

4.4 Compatibility: Quantitative Assessment

Page-length storage produces page behaviors much closer to the permissive policy baseline than does full third-party storage blocking, as shown in Figure 4. These curves show cumulative sums of similarity scores between one of our permissive crawl profiles and all other profiles, normalized to show 1.0 as the maximum possible score (perfect similarity on all instances). The curve showing the similarity scores between the two permissive profiles provides a baseline (i.e., the best scores observed). All pairs of same-policy curves show extremely high consistency. While even the baseline falls short of perfect similarity, there is a clear signal in the grouping of policies. The blocking policies produced the curves farthest from the baseline, as expected, well isolated from all the other policies. The non-blocking policies (site-keyed and page-length) both produced curves much closer to the baseline than to blocking. The stark separation of curves strongly suggests that the non-blocking policies induce significantly less overall deviation from "normal" behavior (and thus less breakage) than does blocking.

Profile	Total Deviations	Severe Deviations
Site-keyed	7	1
Page-length	8	1
Blocking	10	2

Table 1: Candidate URL deviations as assesses by holistic manual grading (n=100)

4.5 Compatibility: Qualitative Assessment

Our evaluation showed that, concerning qualitative end-user experience, the page-length profile performed reliably better than the blocking profile. As described in Section 3.2.3, we had two distinct graders independently perform manual evaluation for each of the three profiles: site-keyed, page-length, and blocking to assess each policy's potential for breaking sites on the 100 candidate URLs. The graders independently graded each candidate site on a scale of 1 to 3 for each of the three profiles to find any deviations from our control profile, permissive (the Chrome default). We conservatively considered deviation from the control visit as a form of breakage, resulting in a score > 1. We summarize the instances of graded breakage for each profile in Table 1.

Considering the 10 deviations observed for the blocking profile, the page-length profile either scored similar (6 cases) or improved (4 cases) in terms of raw grader scores. In contrast to the site-keyed profile (7 deviations), the page-length profile again had either scored equal (4 cases) or better (3 cases). There were only 2 (0.67%, n=300) deviations, both non-severe (cases where graders scored a 2) on the page-length profile, where no deviations were reported on either site-keyed or blocking profiles. We tried to reproduce the reported deviations on subsequent visits later on by ourselves and could not do so. When we further explored for severe deviations (cases where graders scored a 3) there was a single case where the grader reported severe deviation on the website for blocking profile, and both page-length and site-keyed scores indicated no deviations. All other severe deviations were reported on a single URL across all three profiles due to a webpage crash, which after further debugging we concluded that it did not stem from our changes within the browser and did not overlap with the cookie-policy.

5 DISCUSSION

Limitations. Our quantitative assessments of tracking and compatibility are subject to the limitations and risks of automated web crawls. While the scale of our crawl is modest, we believe the Tranco 1k provides a realistic sample of popular, mainstream web content and thus meets our evaluation needs. Spidering 3 links deep past landing pages likewise provides reasonable sampling of site content without exhausting our time and space budget, as PageGraph can generate large volumes of data per page. All our crawlers were stateful and non-headless, giving them a fair chance at evading the most trivial forms of bot detection. More sophisticated bot detection depending on "human" interactions with page content should treat all profiles identically (as bots; we performed no interaction simulations). We thus believe that whatever impact bot detection had on our crawlers, it would have affected all our profiles similarly and not significantly skewed our results.

Implications. While all of our non-baseline policies did well at blocking potential cross-site tracking, page-length was clearly the winner at blocking cross-time tracking as well. And while cross-time tracking presents a much more subtle and less-recognized threat than cross-site tracking, we note that the latest iterations of both Safari ITP and Brave Shields take an aggressive stance at limiting the time for which third-party storage is retained (when allowed in the first place). This convergence is not accidental: limiting the lifespan, not just the cross-site accessibility, of third-party storage appears essential to preserving user privacy.

We saw that page-length performed about as well as site-keyed on our quantitative compatibility estimates, suggesting that the availability of functional storage is more important to user-facing compatibility than its longevity. In part this may be necessity: after all, third-party widgets have to work the first time the user visits a page, not just when they return. But this phenomenon may also be partly due to the trend of aggressively shortened third-party storage longevity pioneered by Safari ITP and continued with Brave's recent rollout of ephemeral third-party storage.

Our manual, qualitative compatibility evaluation, while limited in scope by the labor-intensive nature of the work, produced some suggestive results. We saw generally low rates of reported breakage, with correlation to policy fairly inconclusive. To some extent this inconclusive correlation may simply reflect the intersection of a small sample with limited interactions (e.g., no logins) and inevitable human inconsistencies. It surely also reflects a key difference between our quantitative and qualitative methodologies: the human graders were explicitly looking for unambiguous "breakage" as a user-visible phenomenon, while our quantitative metrics were instead measuring behavioral deviations from a known-good baseline to provide a heuristic upper-bound for *possible* breakage. But it may also reflect an evolution of third-party web publishers practices away from simply assuming that third-party storage (persistent or not!) is available. The fact that our manual assessment was performed some months after the initial data collection, after several browser vendors had announced new and improved stateful tracking protections, lends some credence to this hopeful view.

6 RELATED WORK

Stateful User Tracking. Storage-based user tracking, usually called "stateful" tracking and traditionally involving cookies, has been extensively studied since seminal work by Mayer and Mitchell [24] and Roesner *et al.* [34]. In subsequent years, large-scale, high-impact measurement studies of third-party tracking reported emerging threats like cookie syncing [6], quantified the breadth of cookie tracking across popular sites [22], and introduced widely used measurement frameworks adopted by much subsequent work [11]. Recent work continues to identify evolving and emerging stateful tracking threats in the areas of mobile web tracking [39], pixel tracking [13], and cross-device tracking correlation [41].

Cookie Syncing & Other State Transfers. Third-parties can collude to share stored user tracking identifiers and expand their tracking scope via *cookie syncing*, first measured in depth by Olejnik *et al.* [28] and more recently studied by Papadopoulos *et al.* [30, 31]. Our definition of potentially identifying cookie flows shares similarities with Falahrastegar *et al.*'s methods for measuring distinctive personal identifiers and the entities sharing them across the web [12].

Tracking identifiers can be passed across first-party domains using means other than stored state, as illustrated by Stopczynski *et al.*'s study of attempts to subvert Safari ITP in the wild [38]. At present, such attacks appear focused on reestablishing traditional cookie tracking rather than developing a new tracking paradigm.

Browser Fingerprinting. Measurements of and defenses against stateless tracking via "fingerprinting" have constituted a major category of web privacy research in the years since the seminal Panoptoclick project [10]. Fingerprinting was found to be more common in the wild than first thought [7] and often enabled by new, emerging technologies [21, 27]. More recent works [14, 33] have made conflicting claims about the efficacy of Panoptoclick-style fingerprinting in the wild, leaving its current threat status somewhat ambiguous.

Content Blocking. Published countermeasures against user tracking can be broadly categorized as either blocking tracking-related content (e.g., ads) before they enter the browser or changing browser implementations to mitigate unwanted effects from such content. As most ad and tracker blocking currently depends by filter lists, filter list assessments, improvements, and alternatives are a popular research area [15] in which the PageGraph instrumentation system has been used effectively [9, 18, 36]. Alternatives to blocking content, such as isolated multi-account containers, have also been proposed and compared against traditional ad blockers [17].

Browser Policies & Mechanisms. page-length storage belongs to another category of tracking countermeasure research, which focuses on evaluating and enhancing built-in browser security policies. Potential use (and evasion) of third-party storage blocking was discussed [19] before the era of modern tracking research. Subsequent work has included sophisticated policy enforcement prototype systems [8, 29] and practical fingerprinting countermeasures [20, 26]. Yu *et al.* described an elegantly generalized approach to tracking prevention at the data flow level using k-Anonymity, deployed in the privacy-focused Cliqz browser [40]. Our approach to quantifying tracking potential is loosely inspired by this data flow approach to defining privacy.

7 CONCLUSION

The days of the lose-lose dilemma presented to browser developers by third-party storage—maintain the status quo and enable mass user tracking, or block storage access and break a significant part of the useful web—may be numbered. The combination of cross-site partitioning and cross-time limiting or purging of third-party storage data appears to be effective, both at protecting user privacy (both cross-site and cross-time) and at maintaining compatibility with benign legacy content. We share our contributions with the browser research and development community: the design of our metrics for comparing the privacy and compatibility impact of storage policy changes; our instrumentation platform, made available as open source patches to Chromium (atop Brave's PageGraph), our automated and manual results presented in this work, and the complete automated crawl dataset.

REFERENCES

[1] 2020. Chromium Blog: Building a more private web: A path towards making third party cookies obsolete. https://blog.chromium.org/2020/01/building-more-private-web-path-towards.html. Accessed: 2020-10-12.

[2] 2020. EasyList. https://easylist.to/easylist/easylist.txt. Accessed: 2020-09-17.

[3] 2020. GitHub - disconnectme/disconnect-tracking-protection. https://github.com/disconnectme/disconnect-tracking-protection. Accessed: 2020-10-12.

[4] 2020. HTML Standard: 7.5 Origin. https://html.spec.whatwg.org/multipage/origin.html#origin. Accessed: 2020-10-12.

[5] 2020. Public Suffix List. https://publicsuffix.org/list/public_suffix_list.dat. Accessed: 2020-09-19.

[6] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juarez, Arvind Narayanan, and Claudia Diaz. 2014. The Web Never Forgets: Persistent Tracking Mechanisms in the Wild. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*.

[7] Gunes Acar, Marc Juarez, Nick Nikiforakis, Claudia Diaz, Seda Gürses, Frank Piessens, and Bart Preneel. 2013. FPDetective: Dusting the Web for Fingerprinters. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*. 1129–1140.

[8] Lujo Bauer, Shaoying Cai, Limin Jia, Timothy Passaro, Michael Stroucken, and Yuan Tian. 2015. Run-time Monitoring and Formal Analysis of Information Flows in Chromium.. In *Proceedings of the Symposium on Network and Distributed System Security (NDSS)*.

[9] Quan Chen, Peter Snyder, Ben Livshits, and Alexandros Kapravelos. 2021. Detecting Filter List Evasion With Event-Loop-Turn Granularity JavaScript Signatures. In *Proceedings of the IEEE Symposium on Security and Privacy*.

[10] Peter Eckersley. 2010. How Unique Is Your Web Browser?. In *International Symposium on Privacy Enhancing Technologies Symposium*.

[11] Steven Englehardt and Arvind Narayanan. 2016. Online Tracking: A 1-million-site Measurement and Analysis. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*.

[12] Marjan Falahrastegar, Hamed Haddadi, Steve Uhlig, and Richard Mortier. 2016. Tracking personal identifiers across the web. In *International Conference on Passive and Active Network Measurement*. Springer, 30–41.

[13] Imane Fouad, Nataliia Bielova, Arnaud Legout, and Natasa Sarafijanovic-Djukic. 2020. Missed by Filter Lists: Detecting Unknown Third-Party Trackers with Invisible Pixels. In *PETS 2020-20th Privacy Enhancing Technologies Symposium*.

[14] Alejandro Gómez-Boix, Pierre Laperdrix, and Benoit Baudry. 2018. Hiding in the Crowd: an Analysis of the Effectiveness of Browser Fingerprinting at Large Scale. In *Proceedings of the International World Wide Web Conference (WWW)*.

[15] David Gugelmann, Markus Happe, Bernhard Ager, and Vincent Lenders. 2015. An automated approach for complementing ad blockers' blacklists. *Proceedings on Privacy Enhancing Technologies* 2015, 2 (2015), 282–298.

[16] Kevin A Hallgren. 2012. Computing inter-rater reliability for observational data: an overview and tutorial. *Tutorials in quantitative methods for psychology* 8, 1 (2012), 23.

[17] Xuehui Hu and Nishanth Sastry. 2020. What a Tangled Web We Weave: Understanding the Interconnectedness of the Third Party Cookie Ecosystem. In *Proceedings of the 12th ACM Conference on Web Science*. ACM, Southampton, UK.

[18] Umar Iqbal, Peter Snyder, Shitong Zhu, Benjamin Livshits, Zhiyun Qian, and Zubair Shafiq. [n. d.]. AdGraph: A Graph-Based Approach to Ad and Tracker Blocking. In *2020 IEEE Symposium on Security and Privacy (SP)*. 65–78.

[19] Collin Jackson, Andrew Bortz, Dan Boneh, and John C Mitchell. 2006. Protecting Browser State from Web Privacy Attacks. In *Proceedings of the International World Wide Web Conference (WWW)*.

[20] Pierre Laperdrix, Benoit Baudry, and Vikas Mishra. 2017. FPRandom: Randomizing core browser objects to break advanced device fingerprinting techniques. In *International Symposium on Engineering Secure Software and Systems*. Springer, 97–114.

[21] Pierre Laperdrix, Walter Rudametkin, and Benoit Baudry. 2016. Beauty and the beast: Diverting modern web browsers to build unique browser fingerprints. In *Proceedings of the IEEE Symposium on Security and Privacy*.

[22] Tai-Ching Li, Huy Hang, Michalis Faloutsos, and Petros Efstathopoulos. 2015. Trackadvisor: Taking back browsing privacy from third-party trackers. In *International Conference on Passive and Active Network Measurement*. Springer, 277–289.

[23] Chao Liu, Ryen W White, and Susan Dumais. 2010. Understanding web browsing behaviors through Weibull analysis of dwell time. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. 379–386.

[24] J. R. Mayer and J. C. Mitchell. 2012. Third-Party Web Tracking: Policy and Technology. In *Proceedings of the IEEE Symposium on Security and Privacy*.

[25] Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica: Biochemia medica* 22, 3 (2012), 276–282.

[26] Nick Nikiforakis, Wouter Joosen, and Benjamin Livshits. 2015. Privaricator: Deceiving fingerprinters with little white lies. In *Proceedings of the 24th International Conference on World Wide Web*. 820–830.

[27] Łukasz Olejnik, Gunes Acar, Claude Castelluccia, and Claudia Diaz. 2016. The Leaking Battery. In *Data Privacy Management, and Security Assurance*. Springer International Publishing.

[28] Lukasz Olejnik, Tran Minh-Dung, and Claude Castelluccia. 2014. Selling Off Privacy at Auction. In *Proceedings of the Symposium on Network and Distributed System Security (NDSS)*.

[29] Xiang Pan, Yinzhi Cao, and Yan Chen. 2015. I Do Not Know What You Visited Last Summer: Protecting Users from Third-party Web Tracking with TrackingFree Browser. In *Proceedings of the Symposium on Network and Distributed System Security (NDSS)*.

[30] Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos Markatos. 2019. Cookie Synchronization: Everything You Always Wanted to Know But Were Afraid to Ask. In *Proceedings of the International World Wide Web Conference (WWW)*.

[31] Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P Markatos. 2018. The cost of digital advertisement: Comparing user and advertiser views. In *Proceedings of the 2018 World Wide Web Conference*. 1479–1489.

[32] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. 2019. Tranco: A research-oriented top sites ranking hardened against manipulation. In *Proceedings of the Symposium on Network and Distributed System Security (NDSS)*.

[33] Gaston Pugliese, Christian Riess, Freya Gassmann, and Zinaida Benenson. 2020. Long-Term Observation on Browser Fingerprinting: Users' Trackability and Perspective. *Proceedings on Privacy Enhancing Technologies* 2020, 2 (2020), 558–577.

[34] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. 2012. Detecting and Defending Against Third-Party Tracking on the Web. In *Proceedings of the USENIX Symposium on Networked Systems Design & Implementation*.

[35] Jörg Schwenk, Marcus Niemietz, and Christian Mainka. 2017. Same-Origin Policy: Evaluation in Modern Browsers. In *Proceedings of the USENIX Security Symposium*.

[36] 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*.

[37] Peter Snyder, Cynthia Taylor, and Chris Kanich. 2017. Most Websites Don't Need to Vibrate: A Cost-Benefit Approach to Improving Browser Security. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*.

[38] Martin Stopczynski, Erik Tews, , and Stefan Katzenbeisser. 2020. In Depth Evaluation of Redirect Tracking and Link Usage. In *PETS 2020-20th Privacy Enhancing Technologies Symposium*.

[39] Zhiju Yang and Chuan Yue. 2020. A Comparative Measurement Study of Web Tracking on Mobile and Desktop Environments. *Proceedings on Privacy Enhancing Technologies* 2020, 2 (2020), 24–44.

[40] Zhonghao Yu, Sam Macbeth, Konark Modi, and Josep M Pujol. 2016. Tracking the trackers. In *Proceedings of the 25th International Conference on World Wide Web*. 121–132.

[41] Sebastian Zimmeck, Jie S Li, Hyungtae Kim, Steven M Bellovin, and Tony Jebara. 2017. A Privacy Analysis of Cross-device Tracking. In *Proceedings of the USENIX Security Symposium*. 1391–1408.

Appendices

A SUPPLEMENTARY FIGURES

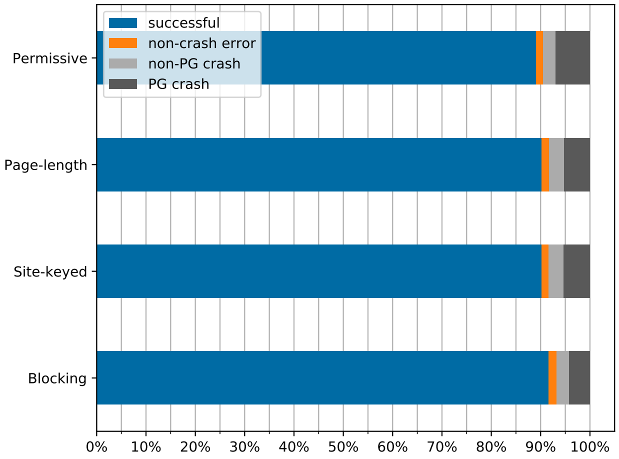

Figure 5: Crawl success rate varied modestly across policies but was always reasonably high.

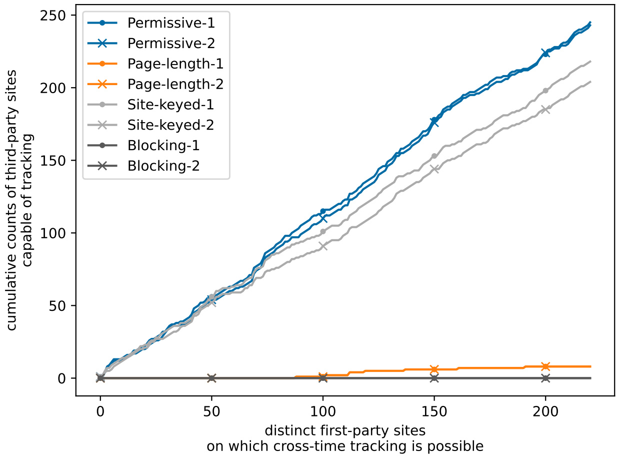

Figure 6: Potential use of *local storage* state for cross-time tracking is much lower than use of cookies, but nearly as frequent under persistent partitioning as under permissive.

B GRADER TASK CHECKLIST FOR MANUAL EVALUATION

1. Scroll the page, are there any obvious portions that did not load and/or break?
2. Are ads loaded, can you click on them? Do they behave as expected (redirect to the ad source/provider?)
3. Is there an embedded video? Can you play/stream it?
4. Is there any embedded social media (Facebook, twitter, Instagram, TikTok) content? Can you click them? Do they take you to the source site?
5. Is this a news or media portal site? Then ...
5.1. Can you search for articles using the search box (if present)?
5.2. Are there social media share buttons? Do they work as expected?
5.3. Are there newsletter sign-up forms/pop-ups? Can you submit them (do not use personal info)?
6. Is this a e-commerce site? Then ...
6.1. Can you search for a product using the search box (if any)?
6.2. Can you add product to your cart and initiate checkout (do not use personal info)?