Shim Shimmeny: Evaluating the Security and Privacy Contributions of Link Shimming in the Modern Web

Frank Li
Georgia Institute of Technology / Facebook[*]

Abstract

Link shimming (also known as URL wrapping) is a technique widely used by websites, where URLs on a site are rewritten to direct link navigations to an intermediary endpoint before redirecting to the original destination. This "shimming" of URL clicks can serve navigation security, privacy, and analytics purposes, and has been deployed by prominent websites (e.g., Facebook, Twitter, Microsoft, Google) for over a decade. Yet, we lack a deep understanding of its purported security and privacy contributions, particularly in today's web ecosystem, where modern browsers provide potential alternative mechanisms for protecting link navigations without link shimming's costs.

In this paper, we provide a large-scale empirical evaluation of link shimming's security and privacy contributions, using Facebook's real-world deployment as a case study. Our results indicate that even in the modern web, link shimming can provide meaningful security and privacy benefits to users broadly. These benefits are most notable for the sizable populations that we observed with a high prevalence of legacy browser clients, such as in mobile-centric developing countries. We discuss the tradeoff of these gains against potential costs. Beyond link shimming, our findings also provide insights for advancing user online protection, such as on the web ecosystem's distribution of responsibility, legacy software scenarios, and user responses to website security warnings.

1 Introduction

Prominent websites, such as online social networks, forums, and messaging platforms, support user-generated content with URLs linking to external destinations. Security and privacy concerns arise when other site users navigate these links. First, the source of the link navigation can be revealed to the destination site through the HTTP referrer, potentially leaking user information via the referrer's URL path and parameters [28]. Additionally, the navigation itself may not be as secure as possible, as users may provide HTTP URLs for websites supporting HTTPS. Finally, the destination may be malicious, such as for malware, phishing, and spam sites.

Link shimming, also called URL wrapping, is a technique that websites can use to protect users from these link navigation threats, as well as for analytics purposes. With link shimming, a website rewrites the URLs displayed on its pages to direct link navigations first to an intermediate endpoint. This navigation "shimming" allows the intermediate endpoint to deploy click-time security and privacy protections (and analytics), before navigating to the original destination.

For over a decade, popular online services have been deploying link shimming, including social networks (e.g., Facebook [7], Twitter [42]), email and messaging platforms (e.g., Gmail and Google Hangouts [5], Microsoft Outlook [20, 21]), search engines (e.g., Google [5], Yahoo [15, 32]), and security products (e.g., Symantec [40], Proofpoint [34], Barracuda [3]). Despite the technique's popularity, there has been little investigation into its purported security and privacy contributions, particularly in today's web ecosystem. Modern web browsers support security and privacy mechanisms that could possibly serve as alternatives to link shimming, without link shimming's potential costs. Thus, there is a question of whether beyond analytics, link shimming serves meaningful security and privacy purposes today.

In this paper, we investigate this question by conducting a case study of link shimming as deployed at Facebook, providing a large-scale real-world evaluation of how users engage with link shimming and its effectiveness at protecting users. We start by analyzing over 6 billion clicks on shimmed links over a month-long period and assess what privacy gains link shimming provides given modern browser privacy mechanisms. Then we evaluate how users engaged with 328M link shim warnings they encountered in that same period, which aimed to protect them from malicious destinations.

On the privacy side, we find that legacy browser clients, while a minority, are still prevalent. We observe nearly 4% of investigated browser clients without any browser privacy features to substitute for link shimming, and between 7-32% of clients (depending on the browser type) providing limited

[*] The author was a visiting researcher at Facebook at the time of this work.

support and still benefiting from link shimming. While the raw percentages may be small, nearly 200 million browser clients are affected, with a skew towards certain subpopulations including mobile-centric developing countries.

We then analyze the effectiveness of link shim warnings at protecting users from visiting suspicious destinations. Modern browsers likewise employ blocklists (e.g., Google Safe Browsing) and interstitials. While link shim and full browser warnings are conceptually similar, link shim warnings arise in different contexts (within a web page) and involve some different security concerns. Our analysis expands upon the existing literature on browser warning effectiveness. We find that user adherence to these warnings is high (around 80%), which is similar to the adherence rates observed for full browser (e.g., Chrome, Firefox) interstitials [1]. Additionally, we identify that only 3% of warned sites were ever in Google's Safe Browsing blocklist. Thus, leveraging link shim warnings with site-specific detection methods and policies can provide broader navigation protections than relying only on browser blocklists. Finally, we evaluate the clickthrough decisions that users make, identifying that users do not appear to be making arbitrary decisions, but are able to avoid malicious destinations to a minor extent. However, they still often make insecure choices, which potentially argues against using user warning outcomes as false positive signals and argues for higher friction warnings.

Ultimately, our results indicate that link shimming can serve meaningful security and privacy purposes when deployed at scale, even for today's web. It does involve potential costs, which we describe, noting that websites must evaluate the tradeoffs themselves. We conclude by discussing insights gained for improving link shimming deployments, as well as for more broadly advancing user online protection.

2 Background

Here, we describe how link shimming operates, as well as the modern browser mechanisms that could serve as potential alternatives to link shimming.

2.1 Link Shimming

Link shimming is used by many prominent online services [3, 5, 5, 7, 15, 20, 21, 32, 34, 40, 42]. While the implementation details and contexts of each service's deployment may differ (discussed further in Section 3.3), they all intermediate on URL navigations using the same technique and can serve similar security, privacy, and analytics functions. Here, we detail how Facebook deploys link shimming.

Facebook uses link shim's navigation intermediation as an opportunity to 1) preserve the privacy of where navigations originated from by minimizing HTTP referrers, 2) improve the security of the navigation method itself through upgrading the network protocol to HTTPS if possible, and 3) secure users from malicious navigation destinations. External-navigating URLs are shimmed on the Facebook website (including the mobile version[1]), as well as in content Facebook distributes (e.g., in email notifications).

To implement link shimming, Facebook uses a Facebook-controlled[2] endpoint (e.g., `facebook.com/linkshim.php`) that takes two URL parameters: 1) the destination URL, and 2) a one-time browser-specific hash, which we will discuss shortly. The Facebook webpage does not directly embed an external (non-Facebook) URL, but instead embeds the link shim endpoint with the external URL passed as a URL parameter. For example, `example.com` would appear on the Facebook platform linking[3] in reality to `facebook.com/linkshim.php?u=http%3A%2F%2Fexample.com&h=HASH`. At click time, users navigate first to the intermediate endpoint for security and privacy evaluation. Below, we discuss how Facebook's implementation of link shimming manages these checks.

Protecting HTTP Referrers: When the link shim endpoint redirects the user to the final destination, browsers (even if legacy) will set the HTTP referrer to the intermediate endpoint. Thus, the original full referrer is hidden, which could have leaked sensitive user information through the referrer URL path and parameters [28]. As examples, the original referrer URL path could reveal what specific Facebook page (e.g., user profile or group page) contained the link, and the URL parameters could contain sensitive user tokens (note that the URL parameters for the link shim endpoint are not sensitive). With link shimming, the destination only observes from the referrer that the navigation source is related to Facebook. In practice, preserving this level of referrer information is valuable for many online services, such as for supporting external analytics, logging, and caching optimizations [28].

Upgrading to HTTPS: To provide stronger link navigation security and privacy, link shimming upgrades HTTP URLs to HTTPS if the destination site supports HTTP Strict Transport Security (HSTS) [14], indicating that connections to the site should always be over HTTPS anyways. The list of HSTS sites is collected from the Chromium browser HSTS preload list [33], as well as HSTS headers from domains crawled by Facebook [22]. We note that while there are HTTP URLs that could be safely upgraded to HTTPS even though the site does not support HSTS, avoiding false positives for non-HSTS sites is challenging due to corner cases which result in broken navigations [12].

Handling Malicious Destinations: If the destination URL is detected as malicious server-side, the intermediate endpoint redirects to one of two warning pages (translated based on user account settings or IP geolocation). For URLs detected as

[1] Facebook's mobile (i.e., Android, iOS) apps do not broadly use link shimming, as the apps can implement link navigation protections directly.

[2] In some link shimming deployments, the intermediary can be a different entity than the deploying website, such as if the intermediary is a third-party security service.

[3] We briefly note that the Facebook website preserves the original destination URL when copying or hovering over the link in the browser, allowing users to still properly inspect where the link navigates to.

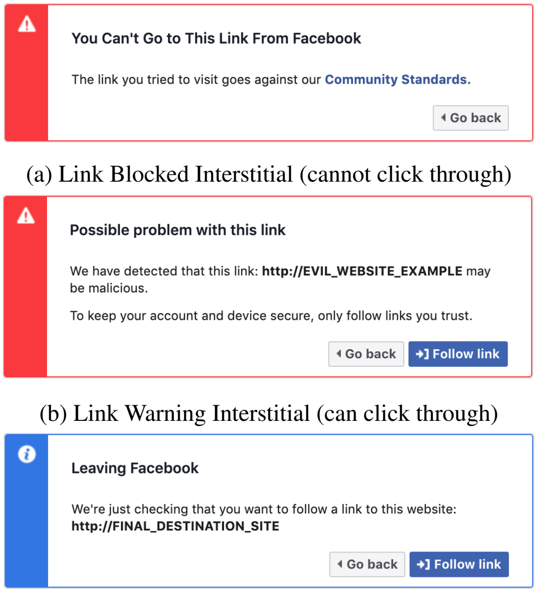
malicious with high confidence (based on automated classifier scores or manual actions), users encounter the blocked access warning in Figure 1a that prevents them from navigating to the destination. However, a subset of URLs are detected as likely malicious but with fewer direct signals. These suspicious URLs receive the second warning shown in Figure 1b, which provides less navigation friction (due to the lower confidence detection) by allowing warning clickthrough to the destination if desired.

Preventing Open Redirection: If the destination URL is not blocked, a final safety check is needed to prevent attackers from abusing the link shim endpoint as an open redirector [23]. During open redirection, a redirection page blindly redirects to any destination URL passed to it. Attackers can leverage this behavior by sharing links to the open redirector that navigate elsewhere, whereas the user clicking on these links may expect to arrive at Facebook based on the URL's domain (this is particularly concerning for phishing attacks).

To protect users from unexpected navigations, the second URL parameter passed to the link shim endpoint is a hash derived from the Facebook cookie values stored for the browser displaying the shimmed link. This hash is also one-time and randomized, such that every generated shimmed link will use a unique hash, thus preventing this hash from being useful for user tracking (e.g., by ISPs or destination sites). When visiting the link shim endpoint, if the hash is not provided or does not match the current browser, a redirection warning as displayed in Figure 1c is shown to inform the user they are leaving the Facebook website (translated as with malicious URL warnings), allowing for click through to the final destination. Thus, this hash prevents attackers from generating shimmed links (that appear as Facebook URLs) that openly redirect, without needing to prompt users on every redirection. However, note that these redirection warnings can appear in benign situations as they cannot be distinguished from potential attacks. For example, if a user Alice directly copies a shimmed link generated for her, and benignly shares it with another user Bob, Bob will encounter the redirection warning when clicking on the shimmed link. Note that to limit warning prompts in benign scenarios, Facebook's link shimming does employ some heuristics, such as permitting shimmed links shared between Facebook friends.

2.2 Modern Browser Protections

Link shimming provides navigation security and privacy protections, but also requires additional redirection hops, increasing navigation latencies. Modern browsers support several mechanisms that could serve as potential alternatives for link shimming's security and privacy functions, without impacting navigation latencies. Ideally, online services could rely on these mechanisms instead of deploying link shimming. When online services first began deploying link shimming over a decade ago [7, 32], many of these mechanisms did not yet exist, so the security and privacy value of link shimming was more prominent. However, we will evaluate whether value remains given modern browser protections, which we will describe here.



You Can't Go to This Link From Facebook

The link you tried to visit goes against our Community Standards.

(a) Link Blocked Interstitial (cannot click through)

Possible problem with this link

We have detected that this link: **http://EVIL_WEBSITE_EXAMPLE** may be malicious.

To keep your account and device secure, only follow links you trust.

(b) Link Warning Interstitial (can click through)

Leaving Facebook

We're just checking that you want to follow a link to this website: **http://FINAL_DESTINATION_SITE**

(c) Link Redirection Interstitial (can click through)

Figure 1: Users can encounter three different types of warnings when clicking on a shimmed link. The warnings shown here are for desktop browsers. Warnings for mobile browsers present equivalent information with mobile-centric designs.

Protecting HTTP Referrers: HTTP headers and HTML features in modern browsers support varying levels of control over the referrer [28][4].

- (Coarse-grained Control) Starting with HTML5, web developers can set the `rel` attribute for anchor (i.e., `<a>`) tags to the value `noreferrer`, which prevents sending the HTTP referrer [27]. This is a coarse-grained mechanism, either allowing the full referrer or preventing it from being sent.
- (Flexible Control) More recently, anchor tags supporting the `referrerpolicy` attribute allow for three options: no referrer, sending only the origin of the referrer rather than the full referrer, and using the full referrer [25].
- (Flexible Control) Most recently, Referrer Policy allows a `<meta>` tag to specify fine-grained referrer control [29], including specifying different referrer values depending on the navigation source and destination.

In practice, maintaining at least the referrer origin is valuable for online services, as it is used for external analytics, logging, and caching optimizations [28]. The latter two flexible control features allow online services to preserve these functionalities while reducing privacy leakage. Compared to these two features, link shimming does not provide any additional referrer privacy benefits. In contrast, the first feature

[4] We briefly note that there are other hacks for mangling the referrer, but they are not compatible across all browsers or JavaScript environments [41].

only allows for either total referrer privacy with lost functionality, or no referrer privacy. Here, link shimming allows online services to maintain functionality without sacrificing referrer privacy.

Upgrading to HTTPS: Browsers supporting HTTPS Strict Transport Security (HSTS) [14] allow web servers to indicate that all connections to the server should be over HTTPS. Without HSTS, legacy browsers lack the context *a priori* to make a reliable decision on if and when to use HTTPS. For upgrading navigation protocols to HTTPS, link shimming primarily benefits such legacy browsers.

Handling Malicious Destinations: Browsers already employ URL blocklists, and display browser interstitials when users navigate to blocked sites. For example, Google's Safe Browsing [13] blocks malware, phishing, and unwanted software sites, and is used by various browsers including Chrome, Firefox, and Safari. For link shim warnings to benefit users over full browser warning, users must adhere more to link shim warnings, or link shim warnings must cover a broader set of dangerous URLs. This broader coverage is particularly plausible as an online service can leverage its specific vantage point for identifying additional malicious destinations, and also detect URLs that violate site-specific policies [8].

Preventing Open Redirection: If link shimming is not used, the open redirection concern is no longer relevant and we do not need to consider browser-provided alternatives.

3 Method

In this section, we detail what data we use for our study and limitations of our method.

3.1 Data Collection

To evaluate how users interact with link shimming, we collect telemetry specifically for our study from Facebook link shim navigations and warning displays. The data spans a month long period, from August 14 to September 16, 2019. This telemetry consists of the following two datasets on what navigation actions occur and browser client characteristics (for understanding their influences).

1) Link shim navigations: We use the following telemetry from when users visit the link shim endpoint during our study.

- Event Timestamp
- Navigation Information: We use the redirection outcome (safe redirection to the destination, or a redirection to a warning), the destination URL, and whether the click came from a shimmed link on the Facebook website (based on the HTTP referrer). We also identify whether link shim upgraded the destination URL to HTTPS (as the site supports HSTS, as discussed in Section 2.1).
- Browser Client Differentiation: For our analysis, we only need to distinguish the different browser clients used when clicking on shimmed links, without identifying users involved. For this, we use the value of a persistent client-specific (not user-specific) Facebook browser cookie [37]. We filter out the 2% of link shim navigations that occurred without the cookie set, as here we cannot differentiate between different browser clients. Browser cookies can be cleared and reset, resulting in the same client with multiple cookie values. However, we note that Facebook has observed only <4% monthly churn rate for these cookies, thus the impact on our client-granularity analysis should be limited. This browser client granularity is most appropriate for our privacy analysis, which will investigate modern versus legacy browser populations. For our security analysis, we require distinguishing distinct warning encounters, which can be likewise done by considering different clients.
- Browser Client Characteristics: We extract the browser and OS names and versions from the HTTP user agent strings, similar to existing documented methods [26]. We additionally use the country-level geolocation of the request's source IP address.

Warning Type	# Raw	# Uniq
None	6.2B	5.3B
Blocked URL Interstitial	28.6M	24.4M
Suspicious URL Interstitial	288K	259K
Redirection Interstitial	299.4M	289.1M

Table 1: Dataset size. For each warning type, we list the raw number of warning displays as well as the number of unique displays, defined as unique (browser client cookie value, warning type, destination URL) tuples. The "None" warning type does not represent actual warnings, rather that the link shim navigations redirect directly to the destination URLs.

2) Warning clickthroughs: As shown in Figure 1, each warning provides a "Go back" button for users. The interstitials for suspicious URLs and redirections additionally allow users to click through a "Follow link" button. Whenever a user clicks on one of these buttons during our study, our telemetry uses the same data as with link shim navigations above (i.e., timestamp, navigation information, browser client differentiation, and browser client characteristics). In addition, we use the interstitial type shown and which button users clicked. Note that this dataset does not contain warning visits where a button was not clicked, but instead the user closed the browser tab or navigated backwards via browser navigation. However, the link shim navigation dataset indicates when link shim redirected to a warning page in the first place, allowing us to compute clickthrough rates.

In total, our study's dataset contains 6 billion link shim navigations as well as 328M warning encounters, with the number of each warning type listed in Table 1. These values are for the raw number of warning displays though, and a user on a particular browser client may click on the same external link and witness the same warning multiple times (a behavior we explore in Section 5.4). Thus, we also list the number of unique warning experiences, where each experience is a distinct (browser client cookie value, warning type, destination URL) tuple.

3.2 Ethics

While we are not directly interacting with users, our study is an empirical investigation of an *in situ* system at Facebook that is. Thus, although IRB approval is not applicable to this research, we take care with our data collection and analysis, focusing only on using the information necessary for our evaluation (e.g., using IP country geolocation instead of the full IP address). We use telemetry specifically for this study on link shim's own actions (e.g., navigation protocol upgrades) and information readily sent by browser clients (e.g., HTTP user agent strings), and our dataset and analysis *do not* use user-specific data. We believe this study's results can help guide improvements to link shimming at Facebook and at other online services, as well as provide insights on advancing online user protection, thus benefiting users broadly.

3.3 Limitations

The data used for this study affords a large-scale real-world evaluation of link shimming. However, as we are evaluating an *in situ* system at Facebook, we are ultimately limited in the explorations we can conduct. These limitations include:

- This work is a case study of a particular implementation of link shimming. While many other online services also employ link shimming with similar functionalities, our results may not translate exactly to other scenarios. For example, users in an enterprise scenario may respond differently to link shimming than Facebook users, given the different deployment context and user population. Also, Facebook both deploys link shimming on its site and manages the intermediary. Users may respond differently to link shimming where a third party (e.g., a security service) serves as the intermediary. We expect our results to generalize most to link shimming deployed and fully managed by a consumer-facing online service. Note that our analysis investigates geographic influences to provide insights on link shimming for users around the world.
- This investigation is a snapshot in time, and exact results may change in the future. However, our findings provide guidance on future directions, and many conclusions should continue to hold true (as we will discuss in Section 6).
- The analyzed data cannot be publicly shared due to privacy constraints. We recognize that this restriction does limit replication. However, we believe that the insights from this work can still be valuable for the security and Internet community, providing empirical grounding on the effectiveness of a common practice. Furthermore, other organizations deploying link shimming can perform a similar analysis to investigate the impact of their systems.
- When studying warning adherence, we only consider suspicious URL and redirection warnings, as blocked URL warnings do not allow clickthrough. While suspicious and blocked URLs are conceptually similar, they are detected by different classifiers and hence are populations that may be characteristically different. Thus, results may not directly translate between the two warning types, although some insights related to user comprehension may still be applicable to both. While in theory, we could experiment with displaying suspicious URL interstitials for blocked URLs, allowing for a more direct comparison, we did not consider this ethically responsible as users may click through to high-confidence dangerous sites (subsequently, our results further support this decision).
- Our datasets may include link navigations by abusive actors, who do not necessarily behave like benign users. However, Facebook extensively deploys systems to detect, prevent, and remediate platform abuse. Thus, we believe the proportion and impact of abuse on our data should be limited.
- We evaluate the live system as is, and do not experiment with different warning workflows or designs. We discuss how future work can explore these directions in Section 6, although we note that our findings suggest that such optimizations will likely have some but limited impact on link shimming effectiveness.

4 Privacy Considerations

As discussed in Section 2.1, link shimming can help protect link navigation privacy by limiting information leakage through HTTP referrers, and upgrading HTTP URLs to HTTPS if possible. When the HTTP referrer is not protected, destination sites may learn sensitive user information from the referrer URL path and parameters [28]. As an example, users clicking external links on their own profile pages may reveal their identities to destination sites through the referrers pointing to the users' profile URLs. Similarly, HTTP web traffic lacks cryptographic security and privacy protections. Modern browsers provide mechanisms that could serve as alternative methods though, as outlined in Section 2.2, without link shimming's cost of additional navigation hops. In this section, we consider the extent to which link shimming provides privacy gains in today's web ecosystem.

4.1 Link Shimming's Privacy Value

For modern browsers, link shimming is not necessary for HTTP referrer protection and HTTPS upgrading, although it can still serve a security purpose, as we will explore in Section 5. However, an online service can benefit from using link shimming for legacy browsers. Here, we analyze the distribution of browsers and browser versions that navigate via a shimmed link, and evaluate the legacy browser populations that benefit from link shimming. We identify legacy browser versions (listed in Table 8 of Appendix A) through online documentation [25, 29, 30, 43].

HTTP Referrer Protection: For referrer privacy, legacy browsers arise in two scenarios. The first is when a browser lacks all referrer protection features, and link shimming is necessary for referrer privacy. For clarity, we will call these *fully legacy browsers*. The second scenario is when a browser only supports the coarse-grained referrer control feature and not the other two flexible control features, which we will call a *partially legacy browser*. In this case, online services can still benefit from using link shimming as it supports origin-level referrers (with practical use cases mentioned in Section 2.2) without fully sacrificing referrer privacy.

For 8 prominent desktop and mobile browsers, Table 2 shows the portion of clients that are fully legacy browsers, and the portion of shimmed URL clicks from such legacy clients. Table 3 depicts likewise for partially legacy browsers.

From Table 2, we observe a non-trivial population of fully legacy browsers. Even for browsers with more up-to-date populations, such as Chrome, Firefox, Safari, and Edge, at least 1% of clients and clicks are from fully legacy browsers. In contrast, browsers such as Microsoft's Internet Explorer (IE) and Opera (particularly the mobile versions) offer these referrer privacy features on a limited number of versions, resulting in a significant portion (over 30% in both cases) of legacy clients. In total, nearly 45M fully legacy browsers navigated shimmed links.

We observe from Table 3 that excluding IE and Edge, the investigated browsers all exhibit a significant fraction (between 7-33%) of both partially legacy browser clients and clicks. There are no IE and Edge partially legacy browsers, as versions of these two browsers either support flexible referrer control or none at all. In total, there are more than 130M partially legacy browser clients. Thus, link shimming allows online services that find value in maintaining the referrer origin to preserve referrer privacy for a substantial population.

HTTPS Upgrading: To evaluate the benefits of link shimming's HTTPS upgrading, we only consider shimmed link navigations that were successfully upgraded, indicating the destination site supports HSTS. Table 4 lists the proportion of distinct browser clients without HSTS support that conducted such HTTPS-upgraded navigations. Browsers such as Chrome, Firefox, and Edge implemented HSTS early on, and have minimal populations of legacy clients. However, Microsoft IE only introduced HSTS in its latest version (IE 11) and a non-trivial population still relies on legacy browsers. The same observation holds for Opera and the mobile-oriented Android and Samsung browsers. In total, we observed 1.5M HTTPS-upgraded link clicks from 800K legacy browsers that lacked HSTS (thus benefiting from link shimming).

Overall, this volume is small. This is in part due to limited HSTS deployment at websites [10, 16, 38], which restricts the number of opportunities where URLs can be confidently upgraded. Users may also tend to post HTTPS URLs for sites supporting HTTPS (e.g., the site's HTTP landing page redirects to HTTPS, and users share the HTTPS URL).

Browser	# Clients	% Legacy Clients	% Clicks
Chrome	917M	1.8%	2.4%
Firefox	28M	2.1%	1.6%
IE	27M	41.8%	22.0%
Edge	8M	1.0%	1.2%
Safari	93M	1.0%	2.2%
Opera	23M	30.8%	38.9%
Android	24M	1.6%	1.4%
Samsung	34M	13.4%	16.2%

Table 2: For popular browsers navigating shimmed links, we show the percent of clients without any HTTP referrer privacy protections (i.e., fully legacy browsers), and the percent of clicks from those legacy clients. Here, link shimming is necessary for referrer privacy.

Browser	# Clients	% Legacy Clients	% Clicks
Chrome	917M	8.9%	7.6%
Firefox	28M	19.1%	14.1%
IE	27M	0.0%	0.0%
Edge	8M	0.0%	0.0%
Safari	93M	20.6%	22.0%
Opera	23M	19.5%	18.6%
Android	24M	31.2%	24.4%
Samsung	34M	32.7%	32.2%

Table 3: For popular browsers navigating shimmed links, we show the percent of clients that are partially legacy browsers, and the percent of clicks from those legacy clients. Here, link shimming allows online services to preserve existing functionality from origin-level referrers without sacrificing referrer privacy.

Finding summary: Even though a majority of browser clients are on modern browser versions supporting HTTP referrer privacy features and HSTS, a substantial fraction are legacy browsers with no or limited support. Link shimming provides privacy benefits for these non-trivial populations, more-so for referrer privacy as browser clients more widely support HSTS.

4.2 Demographic Influences

Here, we evaluate how link shimming's privacy benefits vary across different client OSes and countries.

OS: For HTTP referrer privacy, we observe that link shimming's privacy gains for different OSes are (unsurprisingly) correlated with what browsers commonly run on those OSes. For desktop OSes, we find that less than 0.7% of browser clients on both Linux and Mac OS are fully legacy browsers, compared to 10.2% on Windows. We attribute this large portion for Windows to the prominence of Microsoft IE on Windows, with 42% of IE clients as fully legacy browsers (from Table 2). Meanwhile, Chrome, Firefox, and Safari are most prominent on Linux and Mac OS, and have small fully legacy

Browser	# Clients	% Legacy Clients	% Clicks
Chrome	47.4M	>0.1%	>0.1%
Firefox	1.3M	>0.1%	>0.1%
IE	0.7M	6.6%	4.2%
Edge	2.7M	0.0%	0.0%
Safari	9.6M	0.8%	1.0%
Opera	1.4M	24.5%	28.6%
Android	0.4M	21.9%	22.0%
Samsung	3.6M	6.6%	7.4%

Table 4: For link shim navigations upgraded from HTTP to HTTPS, we show the percent of browser clients that do not support HSTS, and hence benefit from the protocol upgrade.

browser populations. On Android and iOS, less than 3% and 1% of clients are fully legacy browsers, respectively. These mobile OS rates are higher than the desktop OS rates (excluding Windows) due to certain mobile browsers (e.g., Opera Mini) providing limited or no referrer privacy features. We observe similar trends for partially legacy browsers on different OSes so we elide the details.

For HTTPS upgrading, the story shifts due to smaller legacy populations. We observe that link shimming provides similar privacy gains for clients on different OSes. Android OS exhibits the largest legacy population, with 1.2% of its browser clients without HSTS support. For iOS and the desktop OSes (Windows, Linux, and Mac OS), less than 1% of their populations are likewise. The similarities between different OSes largely arises because the most prominent browsers across these OSes all have minimal legacy populations.

Geolocation: We investigate legacy browser usage for countries in our dataset with over a 100K clients. While there is naturally variation between countries, most countries exhibit legacy browser populations commensurate with the aggregate legacy browser proportions. For example, regarding HTTP referrer privacy, the US browser population consists of less than 1% fully legacy browsers and 14% partially legacy browsers. Over 80 countries had fewer than 7% fully legacy browsers and 25% partially legacy browsers. Meanwhile, 0.3% of US browsers lack HSTS support, and 69 countries have 2% or less of their browser population without HSTS.

We observe two notable outliers though.

1. In South Korea, 23% of clients were fully legacy browsers for HTTP referrer protection, the most among investigated countries. The legacy clients are primarily Microsoft IE, whose popularity there has been documented [35].
2. The second outlier involved certain African countries. For HTTP referrer privacy, Sudan, Ethiopia, Angola, and the Democratic Republic of the Congo all had over 20% of clients as fully legacy browsers and over 45% as partially legacy browsers (thus, legacy clients were a majority). The countries with the lowest rate of HSTS support were Ethiopia, Nigeria, Tanzania, Kenya, and Angola. Ethiopia had an anomalously high legacy population without HSTS support, with 32% of its browser clients as legacies. The other African countries listed had legacy proportions ranging from 5-15% of their populations. (Recall that for HSTS support, the US legacy browser proportion was 0.3%). For these African countries, we observe that mobile browsers are predominant and some, particularly Opera Mini, provide limited or no support for referrer protection and HSTS. We note that overall, most of the countries with the highest legacy rates are African or Middle Eastern countries.

Finding summary: Link shimming's privacy benefits vary across OSes and countries due to different legacy browser usage patterns for each. Notably, link shimming provides widespread privacy gains when there is extensive use of legacy Microsoft IE versions and certain mobile browser versions (e.g., Opera Mini) without HSTS and HTTP referrer privacy features. This most impacts the Windows OS, and countries that rely heavily on IE or mobile browsing (include many mobile-centric developing countries).

5 Security Considerations

Besides navigation privacy protections, link shimming can provide click-time checks of URL safety, warning users if the destination URL is dangerous. In addition, link shimming's design potentially creates an open redirection vulnerability [23] at the link shim endpoint. To prevent its exploitation, the link shim endpoint also warns users when redirections are potentially unexpected (i.e., the shimmed link was not generated for them, as described in Section 2.1).

Many modern browsers already employ their own blocklists and interstitials to block malicious sites. For example, Google's Safe Browsing [13] blocks malware, phishing, and unwanted software sites, and is used by Chrome, Firefox, and Safari. For link shim warnings to provide security value over existing browser interstitials, users must adhere more to link shim warnings, or link shim warnings must cover a broader set of dangerous destinations. In this section, we analyze how users engage with link shim warnings and the overlap between link shim and browser warned sites.

We note that prior studies [1, 2, 6, 9, 11, 36, 48] have investigated full browser warnings, particularly in Chrome and Firefox. Link shim dangerous URL warnings are similar in nature, but arise within the context of a particular webpage, whereas warnings from the browser itself may be more prominent (e.g., displaying danger indicators in the URL bar). Understanding how users engage with and adhere to these link shim warnings expands upon the existing literature, shedding light on a warning avenue that can be broadly adopted by various web services. In addition, link shim redirection warnings consider a security scenario not previously explored, but is broadly relevant to URL shortening and redirection services.

Comment on Statistical Analysis: Throughout this analysis, we compare population proportions. In most cases, our populations are large enough that small proportion differences (even less than 1%) are statistically significantly different (such as under a two-tail Z-test), even if they are not necessarily meaningfully different. Thus, we elide discussion of statistical analysis except in cases with smaller populations or where smaller proportion differences have implications.

5.1 Aggregate Warning Adherence

Here we explore warning adherence as measured through warning clickthrough rates. Recall from Table 1 that link shimming displays warnings for blocked URLs, suspicious URLs, and redirections. We do not consider blocked URL warnings as they do not provide a clickthrough option. As a user may click the same shimmed link multiple times and encounter the same warning, we define unique warning encounters as distinct (browser client-specific cookie, warning type, destination URL) tuples (i.e., a particular browser client encountering a given warning for a certain destination). We say that a user clicked through a unique warning encounter if they clicked through any of the associated warning experiences. We investigate repeat warning experiences in Section 5.4. We find that in our dataset, 89.7% of browser clients experience only one unique warning encounter of any type, 4.2% encounter two, and 1.8% encounter three. Overall, our warning clickthrough data is distributed broadly among our browser clients, rather than skewing towards the behavior of heavy hitting subpopulations.

Figure 1b depicts the suspicious URL warning, which notifies users about a potentially dangerous destination while offering a clickthrough option. We find that users click through 18.2% of unique warning encounters. This clickthrough rate (CTR) is similar to those of Firefox and Chrome browser malware and phishing interstitials [1], which range from 7.2-23.2%. Thus, the different warning context for link shim warnings (within a webpage instead of from the browser itself) need not impede warning adherence.

For redirection warnings, we observe an aggregate CTR of 23.0%. We note though that the warning (as shown in Figure 1c) indicates the user is leaving the Facebook website. Users may respond differently depending on whether they were already on the website or not (e.g., a user is emailed a shimmed link). When users are already on the Facebook website (determined as discussed in Section 3.1), the CTR increases to 43.6%. In comparison, the CTR is only 17.7% for those not on the Facebook platform. This difference suggests a level of user comprehension, where they are factoring in the context in which they encounter the warning. (We did not observe a similar difference for suspicious URL warnings, indicating the on-versus-off Facebook context was not an important factor there.)

Our data reveals warning adherence rates but not warning comprehension levels. Users may adhere to a warning due to the friction it causes, rather than fully comprehending the situation. Felt et al. [9] found that user adherence to browser SSL warnings (i.e., not clicking through) was indeed higher than comprehension of those warnings, suggesting that some users encountering link shim warnings likely adhered without full comprehension. In Section 5.5, we manually label a sample of URLs that users did and did not click through to, assessing if perhaps those decisions may be grounded in more reliable knowledge or expectations about the destination site's safety.

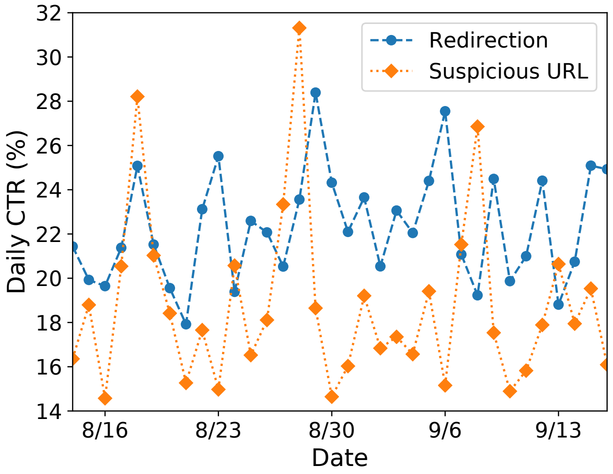


Figure 2: Daily warning clickthrough rates (as percentages). While we observe clickthrough rate variations throughout our study, warning adherence remains consistently high. Note that the y-axis begins at 14%.

Finding summary: Link shim warnings are able to effectively discourage the majority of users from clicking through to potentially dangerous or unexpected sites, exhibiting clickthrough rates comparable with (although not notably better than) Firefox and Chrome browser warnings. The effectiveness of link shim redirection warnings is also relevant to URL shortening and redirection services, which could employ similar warnings. We do note that a sizable minority still clicks through to the destination, and evaluate the safety of these decisions in Section 5.5.

5.2 Temporal Adherence Consistency

While a minority of users click through warnings in aggregate, it is plausible that clickthrough rates could spike at times. We investigate this by considering warning CTRs for unique warning encounters each day, depicted in Figure 2. We observe that for suspicious URL warnings, the CTR can vary widely, ranging between 14-32% of warnings. For redirection warnings, the CTR exhibits a smaller but still wide varying range of 18-28%. We note that these fluctuations are throughout our dataset's duration; there is not a clearly increasing or decreasing CTR trend, and the majority of users consistently adhere to the warnings across time.

We hypothesize that these daily CTR fluctuations may be due to the ever-changing set of URLs receiving warnings as time passes. As we will uncover in Section 5.5, users do appear to evaluate the safety of destination URLs to some extent, perhaps relying on prior experiences or knowledge of popular domains. The shifting patterns in URL usage (e.g., those used in attack campaigns) may result in different clickthrough rates. Another possibility is that different users are encountering warnings over time. Large-scale online phenomena (e.g., malicious campaigns or benign events like viral content) may reach different user subpopulations that exhibit varying clickthrough behavior.

Finding summary: Warning clickthrough rates fluctuate temporally, possibly due to changes in the URLs receiving warnings or the user subpopulations encountering those URLs. Despite these variations, the warnings still consistently discourage the majority of users from clicking through, exhibiting clickthrough rates over time that remain similar to those of browser warnings.

5.3 Demographic Influences

Here we consider how warning clickthrough behavior varies across different browsers, OSes, and countries.

Browsers: For suspicious URL warnings, we observe that browsers exhibit similar CTRs, ranging from 16% (Edge) to 22% (Chrome), suggesting that users perceived the danger warnings similarly across browsers. We note that Akhawe and Felt [1] found that Firefox users clicked through malware and phishing browser warnings at a notably lower rate than Chrome users. That difference may be due to different warning designs between the two browsers, whereas Facebook's link shimming presents a similar design for all browsers.

In contrast, redirection warning CTRs varied widely across different browsers, between 10% (Firefox) and 40% (Android). We note that browsers with lower mobile presence, such as Firefox, IE, and Edge, had the lowest CTRs, all less than 17%, whereas browsers with large mobile penetration (Chrome, Opera, Safari, Android, Samsung) all exhibited CTRs above 22%. We hypothesize that in this less alarming scenario, either the mobile UI conveys less risk or induces less friction, or the population of mobile users (such as in mobile-centric developing countries) perceives less danger.

OS: CTRs for suspicious URL warnings varied more for OSes than for browsers, between 16% for Android and 27% for Windows. Mac OS, Linux, and iOS had CTRs of 23%, 26%, and 22%, respectively. We observe that the desktop OSes are on the higher end of the CTR range, suggesting that for severe warnings (but not more benign ones like with redirections), the mobile environment (either the UI or the users) correlates with higher warning adherence. We note that Akhawe and Felt [1] also evaluated browser malware and phishing warning clickthroughs on desktop OSes, finding that Linux and Windows experienced the highest CTRs depending on the warning types. While our results are not directly comparable, we similarly found Linux and Windows had the highest CTRs.

OS influence on redirection warning CTRs trended in the opposite direction though. The mobile OSes, Android and iOS, experienced the highest CTRs at 39% and 29% respectively, where as the three desktop OSes were all below 26%. These results are consistent with our observations for redirection warnings across browsers.

# Encounters	Suspicious URL	Redirection
1	91.3%	96.6%
2	7.0%	2.5%
3	1.2%	0.5%
4+	0.6%	0.4%

Table 5: Number of times a browser client encounters the same warning for the same destination URL.

Country: We consider the warning clickthrough behavior for countries with at least 1000 suspicious URL warnings (given our set of such warnings is limited once divided among countries), and 100K redirection warnings. Note that warnings are translated, so country-level effects should not be primarily due to language barriers. We observe wide variation among countries, although we do not note any consistent geographic patterns, as found when analyzing link shim privacy considerations. Vietnam, Ukraine, Spain, and Egypt had the highest CTRs for suspicious URLs, with CTRs of 48%, 41%, 40%, and 39%, respectively, indicating that users in these countries often did not heed the warnings. On the low end, Russia and the US had CTRs of 8% and 14%, respectively, perhaps indicating populations more conscious of security concerns. For redirection warnings, while Egypt and Spain again had high CTRs of over 40%, the remaining countries previously discussed had CTRs below 30%.

Finding summary: We find that link shim clickthrough behavior is affected by demographic influences. Most noticeable are the differences between desktop and mobile environments, although which class experiences higher CTRs is not consistent for different warning types, potentially related to the warning severity. Similarly, browser influences appear primarily tied to their mobile prevalence. Finally, we observed wide behavioral differences between countries, potentially reflecting cultural norms or subpopulation security awareness.

5.4 Repeat Warning Encounters

Up to this point, we have considered unique warning encounters, defined as distinct (browser client-specific cookie, warning type, destination URL) tuples. However, a user may engage with the same warning-inducing shimmed link multiple times. For example, a user may click on a shimmed link, observe a warning, decide to return back, but then decide to re-click the link and click through the warning. Here we investigate these repeat warning encounters.

In Table 5, we observe that in over 90% of cases, browser clients engage with a link shim warning only once. Thus, once the user encounters the warning, they either proceed or return back, without re-clicking on the same link. However, a small minority of users do engage multiple times, with more users re-engaging for suspicious URL warnings than for redirection warnings (8.7% versus 3.4%, respectively). This difference is statistically significant under a two-tailed Z-test with $\alpha = 0.05$ ($p < 0.01$). We hypothesize that users may re-engage less with redirection warnings as the scenario may be easier to comprehend and make a final decision on (i.e., decide whether or not they had intended to navigate to the Facebook website or the final destination). Meanwhile, a user may have been initially alarmed by the suspicious URL warning without fully comprehending the situation, and later revisits the warning to better understand and potentially change their decision.

To explore this further, we look at what clickthrough actions users took during the repeat warning encounters and the consistency of their decisions. As shown in Table 6, in the majority of cases, clickthrough decisions were consistent across the multiple encounters. Users never clicked through the warnings in 78% and 71% of cases for suspicious URL and redirection warnings, respectively. For both warning types, users changed their decisions in only 7% of cases, eventually clicking through. In the remaining cases, the users always clicked through, likely simply representing repeat visits to the destination while ignoring the warnings.

We also evaluate over how long of a period users engage with the same link shim warning multiple times, looking at the time difference between the first and last warning encounter. We observe that 86% of repeat engagements with suspicious URL warnings happen within 10 minutes (with 62% happening within a minute), and only 9% happen for longer than an hour period. In comparison, 64% of repeat engagements are similarly within a 10 minute window for redirection warnings, and 29% are over at least a day. We hypothesize that the cause of this difference may be the same as with the difference in re-engagement rates between the two warning types. In particular, with suspicious URL warnings, the first warning may have alarmed users and after the initial response, they may be quick to revisit the shimmed link to better comprehend the situation. However, they are not revisiting the link after extended periods to check if the site is still suspicious. While the majority of those encountering redirection warnings may be behaving similarly, a substantial fraction are revisiting the link after a day, potentially because of willing navigation or the more benign circumstance conveyed by the warning.

Finding summary: Users rarely revisit warning-triggering links, and when they do, they tend to revisit quickly and make the same clickthrough decision. These observations suggest that the link shim warnings are typically conveying their high-level purpose, as users are not confused enough to need to frequently revisit the warnings and change their decisions.

5.5 Safety of User Clickthrough Decisions

Much of the existing literature [1,2,6,9,11,36,48], as well as this work up to this point, has viewed warning clickthrough as a strictly negative user action, assuming reliable detection of the malicious URLs receiving warnings. However, machine learning classifiers and other detection methods suffer from false positives, resulting in errant warnings displayed for users. A primary justification for allowing users to click through warnings is to support user autonomy and provide an easy avenue to proceed in the case of false positives[5]. However, this justification assumes that users can reliably determine false positives. Here, we analyze the safety of the URLs that users do and do not decide to click through to, for both of our warning types that offer clickthrough options (suspicious URLs and redirection warnings).

Warning Type	Never CT	Mixed CT	Always CT
Suspicious URLs	78.5%	7.0%	14.5%
Redirection	71.4%	6.8%	21.8%

Table 6: For browser clients which engaged with the same link shim warning multiple times, we consider whether their clickthrough (CT) behavior was consistent.

To assess the URLs, we manually inspect random samples that should reflect the larger population distribution (we cannot use existing URL classifiers for larger scale labeling as they are the source of the warning-inducing URL labels in the first place). For each warning type, we consider warnings shown in the last day of our dataset, and randomly sample 100 URLs that at least one user clicked through to, and 100 URLs that no user clicked through to. Within two days after our data collection ended, we manually labeled these URLs as malicious, unavailable, or benign, as described below.

- Malicious URLs include phishing, malware, and spam sites. We also include websites hosting content violating Facebook's policies (e.g., promoting scams) [8], although we note that they are a minority (22%) of our malicious URLs.
- Unavailable URLs are those that are no longer online or where a website is active but the specific content is no longer available. While we cannot definitively classify these links, we hypothesize that many were malicious given their short life span, a characteristic often exhibited by malicious domains [4]. Examples of unavailable URLs include removed Youtube videos (perhaps taken down by Youtube), links to Google Forms and Google Drive (often used for phishing and malware distribution), and URL shorteners links that no longer redirect (which typically occurs when the shortener blocks the destination for security reasons).
- We conservatively label remaining sites as benign, with many linking to online stores, sale promotions, and news articles. Attackers can use some of these types of sites for online abuse, but we did not identify explicit signals when inspecting the site itself.

Each URL is independently labeled by two labelers with domain knowledge, with agreement on 95% of URLs. For the other 5%, a third expert labeler served as the tie breaker.

[5] If desired, users can always still navigate to the destination in a new browser window.

Table 7 summarizes our URL labeling results. As expected, for suspicious URL warnings, the vast majority of associated URLs are either malicious or unavailable (many of which were likely malicious as discussed earlier, although we lack definitive proof). In contrast, redirection URLs are mostly benign, although we do detect malicious and unavailable sites. It is possible here that attackers are attempting to leverage the link shim endpoint as a redirector, distributing shimmed links to malicious sites that trigger these unexpected redirection warnings at recipients. The unavailability rate of suspicious URLs is more than twice that of redirection URLs, again hinting that many unavailable URLs are likely malicious.

For suspicious URLs, the sample of sites that users click through to consists of a larger proportion of benign URLs compared to the sample of those not clicked through to (15% versus 6%, respectively). Meanwhile, the two groups exhibit similar proportions of unavailable sites, and fewer sites clicked through to are malicious compared to sites not clicked through to (36% vs 42%, respectively). If we ignore unavailable URLs (as we lack definitive labels), the difference between the maliciousness proportions of URLs clicked through to and those not is statistically significant, under a two-tailed Z-test with $\alpha = 0.05$ ($p = 0.0394$). This difference suggests that to a small degree, users can decide to safely click through the warning. However, for suspicious URL warnings that users clicked through, between 36% (considering only malicious sites) and 85% (considering both malicious and unavailable sites) of the destinations were dangerous. Thus, users are often still making insecure decisions when not adhering to the warnings, and their decisions are unlikely to serve as reliable signals of false positive detections.

For redirection URLs, our data does not reveal where the user truly believed they were navigating to when clicking a shimmed link, and we cannot accurately assess whether they made a safe clickthrough decision or not. However, we do observe malicious (and unavailable) URLs that users do and do not decide to redirect to. Overall, only 3% of URLs that users clicked through to were malicious, compared to 11% otherwise, and the proportions of unavailable sites were similar between the two groups. Again, if ignoring unavailable URLs, the maliciousness proportions of URLs clicked through to and those not is statistically significantly different, under the two-tailed Z-test with $\alpha = 0.05$ ($p = 0.0203$). This difference is consistent with our observations for suspicious URL warnings, and reinforces the notion that users can notice where they are navigating to and avoid clicking through to malicious or unexpected sites, but only to a limited extent.

Finding summary: Our results indicate that user clickthrough decision making is not completely random or arbitrary. Users can recognize and avoid clicking through to dangerous sites once warned, but this recognition is ultimately very limited, and user clickthrough decisions are unlikely to serve as reliable signals of false positive detections. Users still frequently make insecure decisions. Thus, higher friction warnings may better protect users, a direction we explore further in Section 6.

Warning Type	CT	Malicious	Benign	N/A
Suspicious URL	Yes	36%	15%	49%
	No	42%	6%	52%
Redirection	Yes	3%	75%	22%
	No	11%	64%	25%

Table 7: For each warning type, we manually label a random sample of 100 URLs that users did and did not click through to (labeled as CT). We label each URL as Malicious, Benign, or N/A (Not Available).

5.6 Warning Coverage

Finally, we investigate whether link shim warned sites could have already been blocked by browsers. Here we consider Google's Safe Browsing [13] blocklist used by Chrome, Firefox, and Safari, as we can access historical data through StopBadware [39]. For the same URLs randomly sampled in Section 5.5, all of which received link shim warnings, we queried whether the URL was ever blocked by Safe Browsing, conducting the lookup approximately 1 month after the warning displays. Considering suspicious URL warnings only, 9 of the 200 sampled URLs had ever appeared in Safe Browsing. Considering both warning types, 11 out of the 400 sampled URLs were likewise in Safe Browsing prior. Thus, the vast majority of sites that link shimming warned about would not have been blocked by browsers using Safe Browsing.

This observation is not surprising however, as Safe Browsing only blocks certain types of malicious URLs (malware, phishing, and unwanted software domains) that it can accurately classify during web crawls. In comparison, Facebook is able to leverage its own data and vantage point to identify additional malicious URLs. These include malicious sites used specifically on the Facebook platform, which may not be visible to browser vendors, and those whose detection benefits from additional context (e.g., the social network graph, user behavior), particularly relevant for identifying spam and scams. Furthermore, Facebook warns on content violating site policies [8]. From a security perspective, part of link shimming's value comes from an online service's ability to leverage its own detection strategies and warn/enforce on its site-specific policies. We further discuss the web ecosystem's distribution of user protection responsibilities in Section 6.

This analysis only considered Safe Browsing, a prominent browser blocklist. There are other browser blocklists and other sites employ their own malicious URL detection techniques and policies. However, our conclusion about the URL coverage of link shim warnings should hold generally.

Finding summary: Online services can use link shimming to leverage their own malicious URL detection methods and policies, beyond relying on browser blocklists. Combined with the observation that link shim warning adherence is comparable to that of browser interstitials, link shimming can provide broader coverage of dangerous URLs.

6 Discussion

Here we discuss the implications of our study's findings, synthesizing promising directions for advancing online user protection moving forward.

6.1 Link Shimming Costs and Benefits

In this study, we investigated whether link shimming still meaningfully serves its purported security and privacy purposes given the modern web ecosystem. From our evaluation, we found that it can provide privacy benefits for substantial populations of legacy browser clients and security protections for users broadly. Our results consider Facebook users from around the world, which should generalize to online services serving similar consumer populations. However, link shimming does potentially incur several different costs or risks.

- User experience can be negatively impacted. Notably, link shimming adds additional redirection hops to link navigation, increasing navigation latencies. Also, the rewritten URLs may be less usable. While Facebook's link shimming design addresses certain concerns (regarding link display and copying, as mentioned in Section 2.1), other usability concerns may exist that warrant further exploration, as discussed further in Section 6.3.
- Those deploying link shimming must manage the additional complexities of link navigation, as well as increased network traffic due to the navigation intermediation.
- Like network intrusion detection systems (NIDS) or antivirus (AV), link shimming relies on monitoring data related to user actions. Such monitoring approaches may be double-edged swords; the ability to monitor can be used for protection as well as for tracking or analytics. This study evaluated link shimming's protection contributions, just as one might evaluate NIDS or AV detection effectiveness. Investigating data collection from deployers of these technologies, and privacy-preserving alternative methods, are interesting but separate research directions for future work.
- The party that intermediates shimmed link navigations can influence external web analytics and traffic. When the entity deploying link shimming owns or is the same as the intermediary (such as with Facebook), link shimming does not shift the balance of power on the web. However, if the intermediary is a separate entity, such as with security products or third-party analytics services, that intermediary may gain significant influence in the web ecosystem.

These costs are largely site specific (although the impact on user experience is more user dependent). Thus, websites must ultimately weigh the costs and benefits of deploying link shimming for themselves.

6.2 Limitations of Alternative Methods

Given link shimming's cost-benefit tradeoffs, one naturally wonders about alternative approaches to protecting users when navigating links. For navigation privacy, this study already considered modern browser features that do provide equivalent functionality as link shimming, finding that legacy browser clients are still prevalent in practice. While legacy populations may naturally shrink with time, websites could potentially invest in spurring the adoption of more modern browsers. For example, they could support the implementation of features missing from certain browsers, or engage in social campaigns to incentivize moving to modern versions. However, the software community has struggled so far to drive prompt and widespread updating for various types of software, largely due to usability and dependency concerns [17, 19, 24, 44–46].

Even if legacy populations become negligible, link shimming can still protect users from malicious destinations. As discussed in Section 5.6, websites can use site-specific URL detection to cover a broader set of sites than browser blocklists, particularly by leveraging site-specific data and vantage points that are unavailable to browser vendors.

When leveraging site-specific URL detection, a website can evaluate and action on (e.g., block or remove) malicious URLs at other times besides click-time (as provided by link shimming). For example, services can detect malicious URLs at submission time, display time (e.g., when populating a page's content), or in the background on the server side. These methods attempt to eliminate bad URLs before they are displayed to users, and thus avoid displaying warnings to them as done with link shimming. However, they suffer from TOCTTOU (time-of-check to time-of-use) vulnerabilities where URLs are not detected as bad when analyzed, but are detected later on. This issue is particularly relevant when the URLs are no longer on the service's platform, such as when content with URLs is distributed to users via email notifications. Despite Facebook's use of these other approaches as defense-in-depth, millions of users still encounter malicious URL warnings via link shimming, demonstrating that the TOCTTOU vulnerabilities are a practical concern. We also note that some of these methods are computationally expensive. For example, evaluating URLs when displayed requires assessing every external URL on a page, even if the user will only click on a small number of them (if any). Link shimming provides time-of-use checks for only links actually visited by users.

Given the limitations of other approaches, there does not yet appear to be a complete alternative substitute for the security and privacy contributions of link shimming.

6.3 Improving User Protection

Our investigation of link shimming provided insights on improving the technique itself, as well as for improving user protection more broadly. Here we discuss these lessons and directions moving forward.

Legacy Scenarios: Our study highlights that for web security and privacy concerns, we should not dismiss legacy software scenarios, as they can represent a significant population. This is particularly true for certain subpopulations. For example, in certain countries, legacy browsers were the majority. Thus, online services should identify how extensively legacy systems are used by their users, and if the extent is substantial, they should develop strategies specifically for securing legacy users. This lesson likely carries over into other domains with fragmented software ecosystems, such as with smartphones and Internet of Things devices.

Part of link shim's design specifically aids legacy populations. However, the benefits could be furthered, such as by promoting wider adoption of HSTS or the curation of reliable rules for site HTTPS upgrading. This would allow for broader automated HTTPS upgrading for legacy browsers. While efforts like HTTPS Everywhere [12] are promising, they currently fall short of the reliability needed for large-scale link shim HTTPS upgrading. Multiple studies [10, 16, 38] have observed that while raw HSTS deployment numbers remain small, adoption is progressing substantially, providing hope for broader future deployment. In general, promoting more up-to-date software, particular in subpopulations heavily reliant on legacy systems, would drive better Internet-wide security. However, our ability to do is likely limited, as discussed in Section 6.2.

Distribution of Responsibilities: The web ecosystem consists of various players with different vantage points for protecting users. While browsers can support security and privacy mechanisms for protecting users across sites, our study highlights the value of site-specific efforts. For example, Facebook can deter attacks specific to its platform, or leverage its data to identify malicious URLs in a different manner than done by browser blocklists (including detecting categories of malicious sites that are not accounted for by browser blocklists). Thus, web services can and should enhance user online protection, beyond the layer of security and privacy provided by browser vendors.

Human Factors with Website Warnings: This study considered Facebook's link shim implementation as is, without experimenting with different user interfaces. Future work can explore how users react to different website warning designs, as well as the experience of navigating through a shimmed link, particularly for different software stacks (e.g., different OSes or browsers). Additionally, follow-on work could investigate redirection warnings for URL shorteners.

Prior work on browser interstitials [9] found that different warning designs, such as warning colors and text, resulted in different adherence behavior. Similar efforts for link shimming and other website warnings would help guide real-world implementations. We do note that the warning adherence rates we observed are already high and similar to those of the full browser interstitials, whose user designs have received more attention and experimentation. Thus, user-oriented studies of link shimming and other website warnings should provide benefits, although potentially to a limited extent.

Additionally, our analysis of user clickthrough decisions in Section 5.5 indicates that users can avoid clicking through to malicious sites, but only to a limited degree. Thus, clickthrough decisions are unlikely to serve as reliable signals of false positive detections. A substantial portion of the sites that users clicked through to were malicious (possibly the majority of sites, as about half of our manually evaluated sites were already unavailable, with many likely malicious). This finding potentially argues for higher friction warnings where users are not provided with a simple clickthrough button, hopefully discouraging a larger fraction of users from visiting the likely malicious destination. There remains a philosophical tradeoff between user control or autonomy versus user protection. We note that even with link shim warnings that disallow clickthroughs, users can still ultimately visit the destination (e.g., copy-pasting and directly loading the URL). Thus, despite higher friction warnings, user autonomy still remains.

7 Related Work

Despite the prevalence of link shimming, to our knowledge, this study is the first to analyze the technique in practice. However, the components of our analysis touch on aspects considered in prior work. Here, we summarize the prior studies as they relate to each of these aspects.

HTTP Referrer Privacy: Nikiforakis et al. [31] investigated how referrer anonymizing services operated. These services proxy traffic for their customers to hide referrers, as also done by link shimming. Related, Weichselbaum et al. [47] studied CSP deployment by websites, including considering the CSP referrer policy. These studies looked at server-side deployment of HTTP referrer privacy protections, whereas our study provides an empirical evaluation of support by browser client populations.

HTTPS Upgrading and HSTS: Multiple studies [10, 16, 38] empirically evaluated the real-world deployment of HTTPS and HSTS for web servers, observing gradually increasing adoption. On the client software side, Luo et al. [18] analyzed the implementation of security mechanisms (including HSTS) by different mobile browser families, finding that HSTS was more broadly implemented than many other security mechanisms. However, some popular mobile browsers still lacked support. During our investigation into link shimming's HTTPS upgrading, we empirically assessed the real-world support of HSTS by actual browser clients, providing a different perspective on HSTS deployment in the wild.

Browser Warnings: A body of work [1, 2, 6, 9, 11, 36, 48] has studied the effectiveness of browser security warnings, how users react to them, and how warning designs impact adherence. Most relevant to our study, Akhawe and Felt [1] provided the first large-scale field study of browser security warning effectiveness in the wild for Chrome and Firefox. In certain regards, the link shim warnings studied in this work are similar to the browser malware, phishing, and SSL warnings previously considered. However, link shim warnings arise within the context of a web page, rather than from the browser itself, providing an opportunity for web services to deploy warnings themselves. We also consider warnings for potentially unexpected redirections, which are distinct from other browser warning types. Thus, our analysis extends the existing literature on warning effectiveness on the web.

8 Conclusion

In this paper, we provided a large-scale empirical evaluation of the security and privacy contributions of link shimming, a technique widely deployed by major online services, in today's web ecosystem. Using a real-world deployment as a case study, we first assessed the privacy gains that link shimming provides through masking HTTP referrers and automatically upgrading links to HTTPS. We found that while modern browsers support alternative privacy mechanisms, a substantial minority of users are on legacy clients benefiting from link shimming, with a skew towards certain subpopulations such as mobile-centric developing countries. We then analyzed the effectiveness of link shim warnings at alerting users to suspicious destinations or unexpected redirections. We observed high warning adherence rates similar to those of popular full browser interstitials, and broader site coverage than when relying on browser blocklists. Ultimately, our study indicates that link shimming can provide meaningful security and privacy benefits in today's web, and suggests directions for advancing online user protection.

9 Acknowledgments

We thank David Freeman, Neha Chachra, Yiannis Papagiannis, Gelin Zhou, Will Shackleton, Jun Zhang, Subodh Iyengar, Jennifer Martinez, Catherine Anderson, Liz Keneski, and Scott Renfro for providing feedback, discussions, and support in executing the study. We also thank the anonymous reviewers and our shepherd for constructive feedback on improving this paper. Opinions and findings expressed in this paper are those of the author, and do not necessarily reflect those of the research sponsor.

References

[1] Devdatta Akhawe and Adrienne Porter Felt. Alice in Warningland: A Large-Scale Field Study of Browser Security Warning Effectiveness. In *USENIX Security Symposium*, 2013.

[2] Hazim Almuhimedi, Adrienne Porter Felt, Robert W. Reeder, and Sunny Consolvo. Your Reputation Precedes You: History, Reputation, and the Chrome Malware Warning. In *Symposium On Usable Privacy and Security (SOUPS)*, 2014.

[3] Barracuda. Understanding Link Protection, 2018. https://campus.barracuda.com/product/essentials/doc/49055519/understanding-link-protection/.

[4] Leyla Bilge, Engin Kirda, Christopher Kruegel, and Marco Balduzzi. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis. In *Network and Distributed System Security Symposium (NDSS)*, 2011.

[5] Bennett Cyphers, Alexei Miagkov, and Andrés Arrieta. Privacy Badger Now Fights More Sneaky Google Tracking, 2018. https://www.eff.org/deeplinks/2018/10/privacy-badger-now-fights-more-sneaky-google-tracking.

[6] Serge Egelman, Lorrie Faith Cranor, and Jason Hong. You've Been Warned: An Empirical Study of the Effectiveness of Web Browser Phishing Warnings. In *ACM Conference on Human Factors in Computing Systems (CHI)*, 2008.

[7] Facebook. Link Shim - Protecting the People who Use Facebook from Malicious URLs, 2012. https://www.facebook.com/notes/facebook-security/link-shim-protecting-the-people-who-use-facebook-from-malicious-urls/10150492832835766/.

[8] Facebook. Community Standards, 2018. https://www.facebook.com/communitystandards/.

[9] Adrienne Porter Felt, Alex Ainslie, Robert W. Reeder, Sunny Consolvo, Somas Thyagaraja, Alan Bettes, Helen Harris, and Jeff Grimes. Improving SSL Warnings: Comprehension and Adherence. In *ACM Conference on Human Factors in Computing Systems (CHI)*, 2015.

[10] Adrienne Porter Felt, Richard Barnes, April King, Chris Palmer, Chris Bentzel, and Parisa Tabriz. Measuring HTTPS Adoption on the Web. In *USENIX Security Symposium*, 2017.

[11] Adrienne Porter Felt, Robert W. Reeder, Hazim Almuhimedi, and Sunny Consolvo. Experimenting at Scale with Google Chrome's SSL Warning. In *ACM Conference on Human Factors in Computing Systems (CHI)*, 2014.

[12] Electronic Frontier Foundation. HTTPS Everywhere, 2019. https://www.eff.org/https-everywhere.

[13] Google. Google Safe Browsing, 2019. https://safebrowsing.google.com/.

[14] Jeff Hodges, Collin Jacson, and Adam Barth. RFC 6797 - HTTP Strict Transport Security (HSTS), 2012. https://tools.ietf.org/html/rfc6797.

[15] Jcunews. Disable Yahoo Search Result URL Redirector, 2019. https://greasyfork.org/en/scripts/381922-disable-yahoo-search-result-url-redirector.

[16] Michael Kranch and Joseph Bonneau. Upgrading HTTPS in Mid-Air: An Empirical Study of Strict Transport Security and Key Pinning. In *Network and Distributed System Security Symposium (NDSS)*, 2015.

[17] Frank Li, Lisa Rogers, Arunesh Mathur, Nathan Malkin, and Marshini Chetty. Keepers of the Machines: Examining How System Administrators Manage Software Updates For Multiple Machines. In *USENIX Symposium On Usable Privacy and Security (SOUPS)*, 2019.

[18] Meng Luo, Pierre Laperdrix, Nima Honarmand, and Nick Nikiforakis. Time Does Not Heal All Wounds: A Longitudinal Analysis of Security-Mechanism Support in Mobile Browsers. In *Network and Distributed System Security Symposium (NDSS)*, 2019.

[19] Arunesh Mathur, Nathan Malkin, Marian Harbach, Eyal Peer, and Serge Egelman. Quantifying Users' Beliefs about Software Updates. In *NDSS Workshop on Usable Security*, 2018.

[20] Microsoft. Enhanced User Experienced for Office 365 Advanced Threat Protection, 2018. https://techcommunity.microsoft.com/t5/Security-Privacy-and-Compliance/Enhanced-User-Experience-for-Office-365-Advanced-Threat/ba-p/201121.

[21] Microsoft. Office 365 ATP Safe Links, 2019. https://docs.microsoft.com/en-us/microsoft-365/security/office-365-security/atp-safe-links.

[22] Jon Millican. Upgrades to Facebook's link security, 2018. https://www.facebook.com/notes/protect-the-graph/upgrades-to-facebooks-link-security/2015650322008442/.

[23] MITRE. CWE-601: URL Redirection to Untrusted Site ('Open Redirect'), 2019. https://cwe.mitre.org/data/definitions/601.html.

[24] Antonio Nappa, Richard Johnson, Leyla Bilge, Juan Caballero, and Tudor Dumitraş. The Attack of the Clones: A Study of the Impact of Shared Code on Vulnerability Patching. In *IEEE Symposium on Security and Privacy (S&P)*, 2015.

[25] Mozilla Developer Network. <a>: The Anchor element, 2019. https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a#attr-referrerpolicy.

[26] Mozilla Developer Network. Browser detection using the user agent, 2019. https://developer.mozilla.org/en-US/docs/Web/HTTP/Browser_detection_using_the_user_agent.

[27] Mozilla Developer Network. Link Types, 2019. https://developer.mozilla.org/en-US/docs/Web/HTML/Link_types.

[28] Mozilla Developer Network. Referer header: privacy and security concerns, 2019. https://developer.mozilla.org/en-US/docs/Web/Security/Referer_header:_privacy_and_security_concerns.

[29] Mozilla Developer Network. Referer-Policy, 2019. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Referrer-Policy.

[30] Mozilla Developer Network. Strict-Transport-Security, 2019. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security.

[31] Nick Nikiforakis, Steven Van Acker, Frank Piessens, and Wouter Joosen. Exploring the Ecosystem of Referrer-Anonymizing Services. In *Privacy Enhancing Technologies Symposium (PETS)*, 2012.

[32] Pieter. How to Stop Google, Yahoo & Bing from Tracking Your Clicks, 2009. https://greasyfork.org/en/scripts/381922-disable-yahoo-search-result-url-redirector.

[33] Chromium Project. HTTP Strict Transport Security, 2019. https://www.chromium.org/hsts.

[34] Proofpoint. Targeted Attack Protection, 2019. https://www.proofpoint.com/sites/default/files/proofpoint_tap-datasheet-a4.pdf.

[35] Elaine Ramirez. South Korea's Next Presidential Election Might Finally End Its Bizarre Reliance On Internet Explorer, 2017. https://www.forbes.com/sites/elaineramirez/2017/03/03/south-koreas-next-presidential-election-might-finally-end-its-bizarre-reliance-on-internet-explorer.

[36] Robert W. Reeder, Adrienne Porter Felt, Sunny Consolvo, Nathan Malkin, Christopher Thompson, and Serge Egelman. An Experience Sampling Study of User Reactions to Browser Warnings in the Field. In *ACM Conference on Human Factors in Computing Systems (CHI)*, 2018.

[37] Alex Stamos. Preserving Security in Belgium, 2015. https://www.facebook.com/notes/alex-stamos/preserving-security-in-belgium/10153678944202929.

[38] Ben Stock, Martin Johns, Marius Steffens, and Michael Backes. How the Web Tangled Itself: Uncovering the History of Client-Side Web (In)Security. In *USENIX Security Symposium*, 2017.

[39] StopBadware. Clearinghouse Search, 2019. https://www.stopbadware.org/clearinghouse/search.

[40] Symantec. About Click-time URL Protection, 2017. https://support.symantec.com/us/en/article.howto125795.html.

[41] Geek This. Hide HTTP Referer Headers, 2017. https://geekthis.net/post/hide-http-referer-headers/.

[42] Twitter. About Twitter's link service (http://t.co), 2019. https://help.twitter.com/en/using-twitter/url-shortener.

[43] Can I Use. Link type noreferrer, 2019. https://caniuse.com/#feat=rel-noreferrer.

[44] Kami Vaniea, Emilee Rader, and Rick Wash. Betrayed by Updates: How Negative Experiences Affect Future Security. In *ACM CHI Conference on Human Factors in Computing Systems (CHI)*, 2014.

[45] Kami Vaniea and Yasmeen Rashidi. Tales of Software Updates: The Process of Updating Software. In *ACM CHI Conference on Human Factors in Computing Systems (CHI)*, 2016.

[46] Rick Wash, Emilee Rader, Kami Vaniea, and Michelle Rizor. Out of the Loop: How Automated Software Updates Cause Unintended Security Consequences. In *USENIX Symposium On Usable Privacy and Security (SOUPS)*, 2014.

[47] Lukas Weichselbaum, Michele Spagnuolo, Sebastian Lekies, and Artur Janc. CSP Is Dead, Long Live CSP! On the Insecurity of Whitelists and the Future of Content Security Policy. In *ACM Conference on Computer and Communications Security (CCS)*, 2016.

[48] Joel Weinberger and Adrienne Porter Felt. A Week to Remember: The Impact of Browser Warning Storage Policies. In *Symposium on Usable Privacy and Security (SOUPS)*, 2016.

A Legacy Browser Versions

Browser	**Coarse RP**	**Flexible RP**	**HSTS**
Chrome	16	51	4
Firefox	33	50	4
IE	11	11	11
Edge	12	12	12
Safari	5	11.1	7
Opera	15	38	12
Android	2.3	51	4.4
Samsung	4	7.2	4

Table 8: We list the browser versions that began supporting referrer privacy (**RP**) mechanisms (for both coarse-grained and flexible control) and HTTP Strict Transport Security (**HSTS**), based on online documentation [25, 29, 30, 43]. Legacy browser versions are lower than those listed.