2019-04-01

After HTTPS: Indicating Risk Instead of Security

Matthew Wayne Holt
Brigham Young University

After HTTPS: Indicating Risk Instead of Security

Matthew W. Holt

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Daniel Zappala, Chair
Kent Seamons
Parris Egbert

Department of Computer Science

Brigham Young University

ABSTRACT

After HTTPS: Indicating Risk Instead of Security

Matthew W. Holt
Department of Computer Science, BYU
Master of Science

Browser security indicators show warnings when sites load without HTTPS, but more malicious sites are using HTTPS to appear legitimate in browsers and deceive users. We explore a new approach to browser indicators that overcomes several limitations of existing indicators. First, we develop a high-level *risk assessment framework* to identify risky interactions and evaluate the utility of this approach through a survey. Next, we evaluate potential designs for a new *risk indicator* to communicate risk rather than security. Finally, we conduct a within-subjects user study to compare the risk indicator to existing security indicators by observing participant behavior and collecting feedback. Our results suggest that risk indicators make users more confident in judging their risk and that participants prefer risk indicators over current security indicators. In addition, users take fewer risks in the presence of risk indicators, making this a promising direction for research and implementation into web browsers.

Keywords: security, privacy, risk, user study, web browsers, risk communication

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Daniel Zappala, for the continuous feedback and patient counsel in developing and improving this work, along with the other committee members whom I recognize for making this paper what it is. I also extend a very humble thank-you to Jennifer Bonnett in the department for recognizing one who needed help on the so-called BYU Spring Break holiday and for pleasantly accomplishing the paperwork and other formalities, even during a stressful season. Torstein Collett helped conduct the online surveys and user study and performed coding and analysis of their results. Justin Wu was an invaluable source of knowledge and insight into the academic process and thinking critically about the science of our experiments. My previous advisor, Dr. David Wingate, is also acknowledged for helping me hit the ground running at the start of graduate school, working with me through the process of writing my first academic paper, and bringing meaningful spiritual content into his classes. And to all others whom I worked with or next to, thank you for bringing a splash of vibrancy into our plain, cream-colored research lab, and for being supportive when I had questions, or was frustrated or excited, in the process of this work (you know who you are).

Also, thank you to whomever had the genius idea of bringing in the handwarmer, microwaving it, and placing it by the thermostat to turn on the A/C when it got hot in here.

Warning: This paper was written under the influence of music by Two Steps From Hell.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Web is changing. Until 2014, less than 25% of pages were reportedly loaded over encrypted HTTPS connections.[1] As of January 2019, however, Google reports that 80-91% of page loads in Chrome use HTTPS.[2] This can be explained largely due to several coordinated shifts in the ecosystem: (1) the practical cost to site owners for TLS certificates has dropped to zero, (2) the validation and renewal procedures can be entirely automated through a newly-standardized protocol called ACME [3], (3) updated laws and policies mandate the use of HTTPS,[3] and (4) public support in the form of cross-signed roots, donations, sponsorships, and widespread technical integrations have made Let's Encrypt the largest (most-issuing) CA in the world in a matter of months.[4]

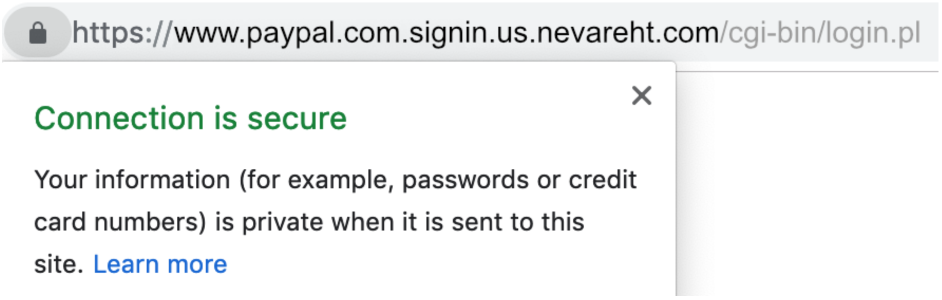
Although this progress is impressive, it comes with trade-offs. Historically, malicious sites would not use HTTPS due to its prohibitive cost and the tedious validation required to deploy it, and browsers started indicating secure connections by showing a lock icon next to the page URL to assure visitors that their connection was encrypted. Even though the absence of a lock icon does not stop users from taking risky actions [42], the presence of the padlock became the standard way to determine whether it was safe to enter your secret credentials, credit card details, or other sensitive data.

[1] Percentage of Web Pages Loaded by Firefox Using HTTPS, Let's Encrypt, `https://letsencrypt.org/stats/#percent-pageloads`

[2] Google Transparency Report, `https://transparencyreport.google.com/https/overview`

[3] White House Office of Management and Budget memorandum M-15-13: "A Policy to Require Secure Connections across Federal Websites and Web Services", `https://https.cio.gov/`

[4] Merkle Town, Cloudflare, `https://ct.cloudflare.com`



https://www.paypal.com.signin.us.nevareht.com/cgi-bin/login.pl

Connection is secure

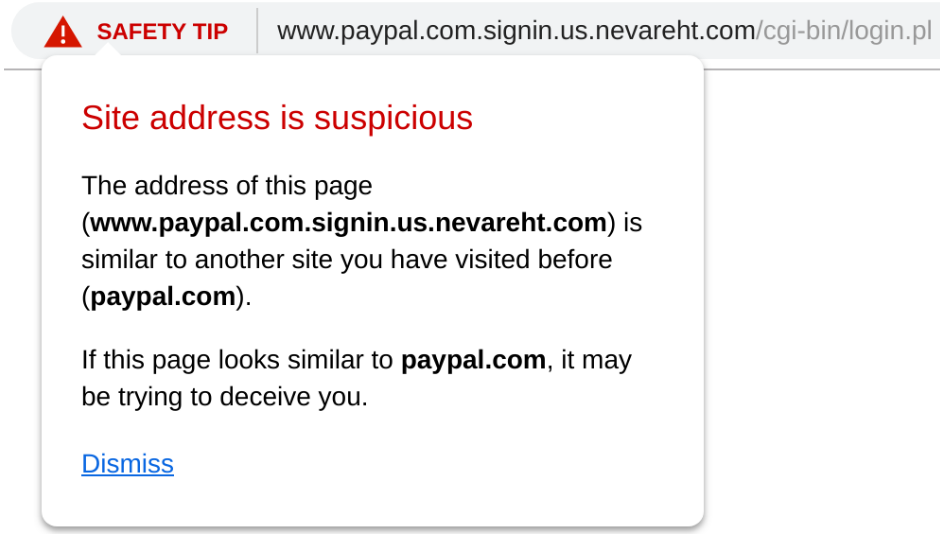
Your information (for example, passwords or credit card numbers) is private when it is sent to this site. Learn more

Figure 1.1: Current security indicators make malicious sites even more deceptive. (This domain was obtained from certificate transparency logs in 2018.)

However, researchers reported that phishing sites in 2016 used HTTPS nearly 3% of the time, and the rate is increasing quickly [30]; by 2017, 25% of the time; and in 2018, nearly 50% of phishing sites used HTTPS [50]. Other studies have shown that users wrongly interpret the padlock icon to mean that the site itself is secure, not that the ephemeral connection to the (possibly malicious) site is secure [39]. This growing use of HTTPS makes scams more effective, and it's one of the reasons why Chrome has begun the process of hiding the padlock icon in favor of negative indicators only: warning when a site does not use proper HTTPS, and showing no positive indicator when they do [41]. That avoids giving malicious sites extra legitimacy, but in a world full of HTTPS, there is little current browser indicators can do to be useful.

Clearly, new browser indicators are needed. Because indicators in their current form still provide helpful warnings about insecure connections and blacklisted sites, we do not suggest removing them or making drastic changes. Instead, we can repurpose them. But doing so requires looking at more than just the security properties of a connection. Fortunately, web browsers—acting as user agents—have access to a plethora of technical information crossing several layers as they look up, download, and render web pages. This information could be used to expand the capabilities of indicators to detect risky interactions on deceptive HTTPS pages.

This paper explores a new approach to browser indicators for an all-HTTPS world. It shifts the browser indicator paradigm from security and privacy to risk and deception.


SAFETY TIP www.paypal.com.signin.us.nevareht.com/cgi-bin/login.pl

Site address is suspicious

The address of this page (**www.paypal.com.signin.us.nevareht.com**) is similar to another site you have visited before (**paypal.com**).

If this page looks similar to **paypal.com**, it may be trying to deceive you.

Dismiss

Figure 1.2: Our approach uses a risk indicator that warns users of deceptive sites, with descriptive warnings explaining why the site may be deceptive.

Instead of toggling its appearance based on connection security, our indicator is designed to appear with interactions the browser deems risky. Although "risk" can be ambiguous and "risky interaction" is subjective, we refer to interactions that are more likely to expose the user, their information, or their device to abuse based on the technical characteristics of the site, context of the current interaction, and user's browser history. We use a privacy-preserving approach that makes purely local decisions and does not reveal a user's browsing history to third parties.

For example, suppose a user clicks a hyperlink to a page that redirects to four different sites before settling on a site they have never been to, where they enter information into a form. This interaction of data entry could be considered risky because similar patterns are common with phishing sites: the link might be malicious, redirects could be used for tracking and misdirection, and there is no good reason to trust a new site with a user's sensitive information. In today's browsers, this site would still appear legitimate, as shown in Figure 1.1, because it uses HTTPS and is not on a blacklist. Our approach has the browser indicate the risk to the user, as shown in Figure 1.2.

We view this work as complementary with ongoing efforts to protect users from a variety of threats they encounter when browsing the Web, such as phishing and malware. These attacks are difficult to prevent because they take advantage of the same characteristics as legitimate sites (similar domain names, valid certificates) and rely on user habits (e.g. trusting links) to succeed. While browsers and third-party security software attempt to block attacks, the most common solution in use today requires curating an extensive blacklist through reports of threats from both users and experts (sometimes aided by machine learning [52]). Because blacklist entries can only exist after a threat is identified, there is a delay between when a threat appears and when the blacklist starts protecting users. Our work has the potential to catch sites before they appear in a blacklist, while also protecting from a broader array of threats than just phishing and malware attacks.

Our contribution has three main parts. First, we describe a high-level *risk assessment framework* that is designed to expose not only phishing, but deception attacks in general, by adding contextual information that current security indicators do not consider, to determine if the current interaction is risky. We develop a new set of browser warnings based on this framework and survey users on the helpfulness of these warnings. Second, we design a *risk indicator* to communicate potential risks to users as a possible replacement for current security indicators. Finally, we conduct a user study which evaluates the effectiveness and perceived helpfulness of risk indicators and compares them to the existing security indicator.

Our results suggest that warnings issued by our framework are helpful overall; that users feel more confident about judging their risk after investigating the warnings than with current security indicators; and that users like having the second opinion or being made aware when something *"seems off."* Results from one of our surveys helped us choose a risk indicator design which has a positive sentiment but still conveys a mild sense of alarm. We also have some evidence that, given optional tasks, the presence of risk indicators can be correlated to fewer risky interactions.

Artifacts: We have created a companion website at `https://indicators.internet.byu.edu` that provides the source code, study materials, and data.

Chapter 2

Related Work

There is an extensive amount of prior research about browser warnings and security indicators, and we incorporate some of the recommendations for improving them into our own work. However, the object of our research is distinct from prior work which examines how noticeable browser indicators are [8] or how to improve warning adherence rates [38]. In our studies, we informed each participant to look for indicators, mostly eliminating the variable of whether the indicator is noticed; and the goals of our warnings are different from traditional/current warnings which encourage users to leave the site.

Similarly, there is a broad range of existing work to detect phishing. Although our work is not focused on detecting phishing, we do find several of these techniques useful in our goal, which is assessing risk. Interactions are risky if they may be subject to deception, and we view deception as when the user may not be aware of an interaction or when an interaction may not result in what the user intends. The browser, acting as a user agent, has unique technical insights that can expose deception.

Of the many works that are specifically about detecting deceptive sites or grading a site's credibility, they specifically focus on fact-checking the content of a page or filtering fake reviews [13, 15, 32]. By contrast, we do not analyze the content of the page because we view deception generally, without constraining the scope of threats to just fake reviews, false claims, or phishing. Our work is largely agnostic of any specific threat.

2.1 Browser Indicators and Warnings

Browser indicator refers to an optional icon next to the URL which is controlled by the browser to indicate a state of security or insecurity with regards to the current connection and sometimes blacklist status. Clicking indicators yields more information and site- or connection-specific settings.

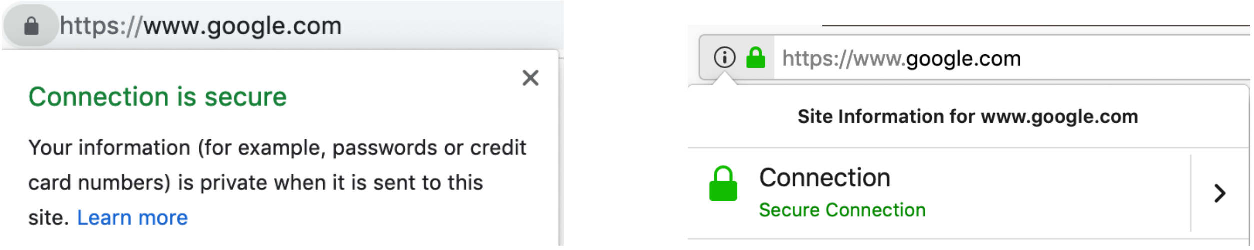
Browser warnings typically appear in the content area of the browser or spawn from an icon in the URL bar, depending on the context. Both of these are currently used in mainstream browsers to warn the user of threats or give them confidence in their activities.

Browser warnings have changed throughout the years. In 2013, Akhawe and Felt studied browser warnings at a massive scale using Chrome and Firefox telemetry and found that users get fatigued and ignore frequent warnings, that difficulty in bypassing warnings is not necessarily a deterrent to do so, and that most users who bypass warnings click through after a shorter time than those who heed the warning and leave [2]. As a result, warnings have generally become louder, less wordy, and more accurate. This is called "opinionated design" which was recommended in 2015 by the Chrome team [10] and again in 2019 by the Firefox team [51].

Since then, newer research confirmed that these improved browser warnings actually can be effective and that users are not oblivious to warnings' utility [38]. They also present evidence that habituation, or bypassing warnings due to fatigue, is not a significant factor in modern browser warnings: users do not consistently click through warnings anymore.

For the time being [41], browsers also show positive reinforcement in the presence of secure connections. Google Chrome and Firefox have a security indicator visible to users when the page is transmitted encrypted with TLS by showing a lock icon. Until recently, the word "Secure" was visible next to the URL on HTTPS pages [11] but was removed to reduce miscommunications [41].

While it is good that the page was loaded securely, the lock icon does not indicate that the web page or site is secure. This is contrary to what many users believe, who see



https://www.google.com

Connection is secure

Your information (for example, passwords or credit card numbers) is private when it is sent to this site. Learn more

https://www.google.com

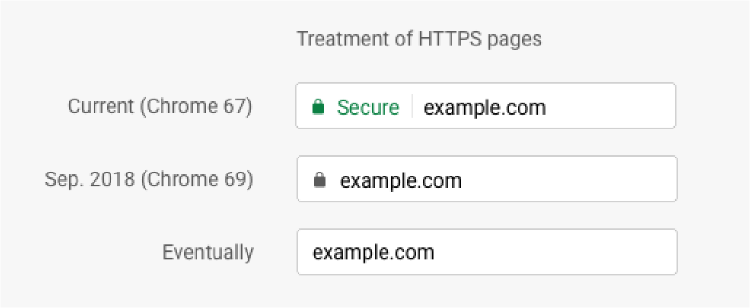
Site Information for www.google.com

Connection
Secure Connection

(a) Google Chrome's security indicator with on-click popup explaining the lock icon.

(b) Mozilla Firefox's security indicator has less verbiage on click but re-states the hostname.

Figure 2.1: Security indicators in two of the major browsers as of February 2019.



Treatment of HTTPS pages

Current (Chrome 67)	Secure example.com
Sep. 2018 (Chrome 69)	example.com
Eventually	example.com

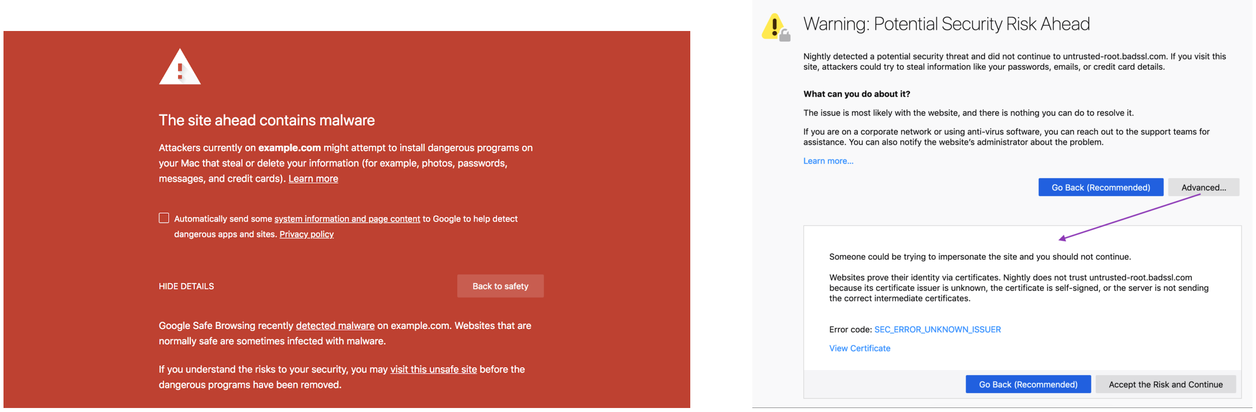
Figure 2.2: Chrome's plans for secure indicators as of Q1 2018 [41].

the lock icon and think this means the site itself is secure [39]. In fact, malicious sites can now easily and automatically obtain domain-validated TLS certificates to procure positive reinforcement in browsers; this is one of the most substantial reasons new research and new methods are needed. It is mainly for this reason that Chrome, and perhaps eventually other major browsers, are slowly phasing out the security indicator and will only show warnings when something is amiss with the connection [11], as shown in Figure 2.2.

Clicking the icon in Chrome yields more information that explains that the connection is secure, but it also implies that the site will keep the user's sensitive information private, with no real technical or even intuitive reason to substantiate that claim (see Figure 2.1a). Mozilla Firefox shows a similar lock icon, with less verbiage on click, as in Figure 2.1b.

As for deception and malware, Both Chrome and Firefox rely on Google Safe Browsing[1] to warn users of malware or deceptive sites. Safe Browsing is essentially a blacklist which

[1] Security/Safe Browsing, Mozilla, `https://wiki.mozilla.org/Security/Safe_Browsing`


The site ahead contains malware

Attackers currently on **example.com** might attempt to install dangerous programs on your Mac that steal or delete your information (for example, photos, passwords, messages, and credit cards). Learn more

□ Automatically send some system information and page content to Google to help detect dangerous apps and sites. Privacy policy

Google Safe Browsing recently detected malware on example.com. Websites that are normally safe are sometimes infected with malware.

If you understand the risks to your security, you may visit this unsafe site before the dangerous programs have been removed.

(a) Google Chrome

Warning: Potential Security Risk Ahead

Nightly detected a potential security threat and did not continue to untrusted-root.badssl.com. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.

What can you do about it?

The issue is most likely with the website, and there is nothing you can do to resolve it.

If you are on a corporate network or using anti-virus software, you can reach out to the support teams for assistance. You can also notify the website's administrator about the problem.

Someone could be trying to impersonate the site and you should not continue.

Websites prove their identity via certificates. Nightly does not trust untrusted-root.badssl.com because its certificate issuer is unknown, the certificate is self-signed, or the server is not sending the correct intermediate certificates.

Error code: SEC_ERROR_UNKNOWN_ISSUER

(b) Mozilla Firefox [51]

Figure 2.3: Examples of interstitial warnings in mainstream web browsers as of March 2019.

is maintained by Google for the purpose of hiding sites from search results and showing warnings if a site is known to be malicious. The warnings in Chrome and Firefox are quite loud, filling the entire page (Figure 2.3).

Microsoft SmartScreen[2] works similarly on Windows with the Edge browser. It shows a warning if a site is on a blacklist or if a site exhibits suspicious behavior, although what that comprises is not publicized. Screening downloads is more nuanced: if a file is on a blacklist, the user is shown an error, and if the file is not on a whitelist, the user is shown a warning.

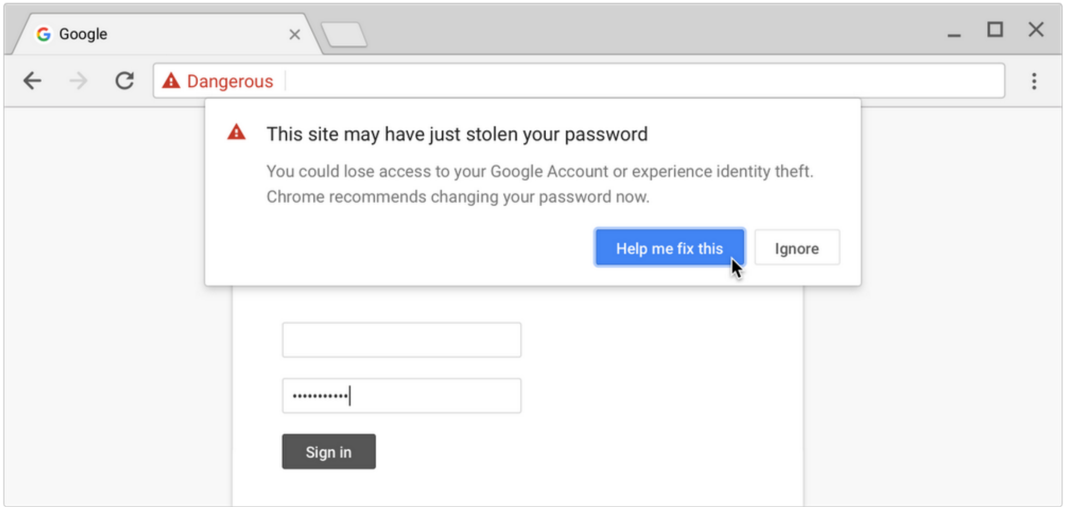
Safe Browsing and SmartScreen become effective fairly soon after a new threat emerges, but not instantaneously, and their proprietary natures make it tricky for other browsers to adopt. Further, reporting malicious sites can be risky. Malicious sites can be reported by users, but this often requires visiting the malicious site and using an in-browser feature.[3] Reporting can be more safely done out-of-band,[4] but doing so loses critical context,[5] making reports less comprehensive.

[2] Windows Defender SmartScreen, `https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-smartscreen/windows-defender-smartscreen-overview` (Microsoft)

[3] How to report a phishing Web site, `https://support.microsoft.com/en-us/help/930167/how-to-report-a-phishing-web-site` (Microsoft)

[4] Send a Report to Google, `https://safebrowsing.google.com/safebrowsing/report_general/` (Google); Report unsafe site, `https://www.microsoft.com/en-us/wdsi/support/report-unsafe-site` (Microsoft)

[5] For example, deceptive redirect chains.



This site may have just stolen your password

You could lose access to your Google Account or experience identity theft. Chrome recommends changing your password now.

Figure 2.4: An example of Chrome's predictive phishing warning [4].

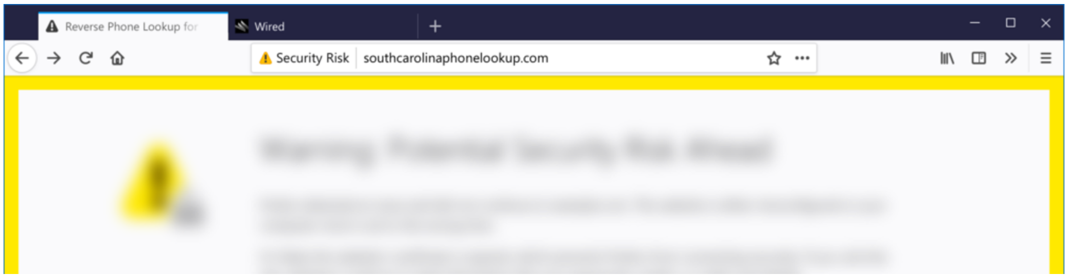


Figure 2.5: Mozilla Firefox's redesigned warnings use color scaled to risk level [51].

Google Chrome has one extra warning type that is tied to a phishing-specific countermeasure called predictive phishing protection [4]. The details of how it works are not published, but its approach is similar to ours in that it detects emerging phishing threats and shows a warning in a pop-out similar to what we design, which originates outside the content area of the page (Figure 2.4). However, as far as is published, this feature only applies to Google sites and activates only after entering your credentials [4].

Firefox recently redesigned their warnings to more effectively communicate potential risks, which is more like our own approach. Some Firefox warnings outline interstitial warnings in a solid color appropriate to the level of risk, and the security indicator is replaced with an alert icon and text that says "Security Risk" (Figure 2.5) [51].

In general, warnings about phishing, TLS errors, or malicious sites and downloads are a tricky problem. Most browsers (and related work which recommends new warnings [27]) show full-page interstitials, but malicious sites can impersonate these kinds of warnings because they appear inside the content area of the window. In those cases, the “Back to safety” button (or equivalent) can actually return a user to a phishing page where they think they are protected.

2.2 Phishing & Deceptive Sites

We consider phishing detection to be related work mainly because phishing sites share many attributes common to other kinds of deceptive sites; and although not all risky interactions are only on deceptive sites, interactions on sites which are trying to be deceptive inherently pose greater risk. Thus, our paper builds on published phishing work, but not with the goal of classifying phishing sites with high accuracy. Instead, we are interested in features that are common among deceptive sites; or, in other words, we are interested in the *inputs* used to classify phishing more than the *output* of classifiers.

Numerous methods for combating phishing have been proposed previously, all of which fall into two main camps: prevention and detection [21]. Precise definitions of the two vary, but in general, prevention’s goal is to prevent the user from ever reaching a phishing site, while detection’s goal is to detect when the user has reached a phishing site and to warn them accordingly. Much prevention work covers detecting phishing emails [12], but we note that these days there are many other effective ways to spread malicious links globally, especially by using social media [49]. Because of the many venues with which deceptive links can spread, our work is strictly detection-focused, where the user has presumably already accessed a malicious link and is now loading it in a web browser. Since the boundary of prevention and detection is fuzzy and often overlap, here we examine related works from both angles.

Whitelists. A whitelist is a list of approved sites that have been cleared for access. At a global scale, whitelists can become unwieldy, but Han et al. suggest a personalized whitelist that correlates a user’s input historically to each site and warns if certain features have changed, such as the IP address for the domain or the web site's TLS certificate [5]. This technique does not handle legitimate DNS updates, HTML changes, or certificate renewals particularly gracefully.

Blacklists. A blacklist is a list of flagged sites that have been reported as unsafe or deceptive. Blacklists do not protect against emerging threats because new sites have to be seen and reported before they can be added to the list. Despite this limitation, blacklists are very effective in practice and are used by ad blockers and almost all major browsers. Blacklists are usually considered universal; i.e. a site that is malicious to one user is to any other also [35]. Blacklists are usually manually curated, imposing a huge maintenance burden, but Whittaker et al. demonstrate a machine learning approach used by Google to automatically maintain a phishing blacklist for web browsers [52]. Prakash et al. present PhishNet as a method for growing blacklists which uses enumeration of known phishing URLs to discover new ones; those URLs are then spliced into tokens for matching against a word blacklist, but then external resources are needed to score the legitimacy of candidates [34]. Once curated, a blacklist is then either distributed to clients (requiring regular updates) or hosted centrally (requiring network queries to check if items of interest are in the list). The former has scalability problems and the latter can have negative privacy implications.

Heuristics. Another few methods examine the behavior of a web page as it relates to user input [21]. For example, Joshi et al. propose a system which submits wrong data and inspects the response before submitting the real information [20]. Shahriar and Zulkernine propose a similar system that monitors website behavior in the presence of random inputs [43]. These techniques are useful, as they require phishing sites to imitate not only appearance but also behavior of the genuine site; and they are effective when phishing sites are not true to the actual site's behavior. However, they are likely to fail if the phishing site verifies the user's input by submitting it to the genuine site in the background and proxying back the response, which is now easily automated [7], even bypassing some two-factor authentication. Phishing sites may also be incentivized to always fail login attempts to derive alternative credentials from a user, thus conducting an enumeration attack.

Machine learning (ML). Several feature-based defenses against phishing have been developed, including the use of machine learning systems—especially support vector machines, neural networks, and other gradient-descent methods [6, 21, 23].[6] The vast spread of ML work uses hundreds of features ranging from measuring the spelling accuracy on a page, HTML formatting, and URL contents to domain name WHOIS information, form properties, and site popularity [40]. Not all features are globally the same; Fette et al. suggest using number of previous visits in browser history as a feature [12]. A technique by Varshney et al. indirectly leverages machine learning and popularity metrics by obtaining results from a trusted search engine to determine if the page can be trusted [47]. Malicious pages that use evasion techniques such as images and executable scripts can fool text-based feature classifiers, and legitimate pages that make heavy use of images or scripts to fill the page can be misclassified as fraudulent in an over-correction for false negatives. Many machine learning classifiers cannot easily explain why a certain classification was given; a whole field of machine learning has spun off to tackle this problem, known as "interpretable deep learning."[7] In addition, assumptions made about some features used by classifiers are now being shown to be incorrect, rendering the system less reliable; for example, phishing domains are not necessarily short-lived [22]. The difficulty of retraining learned models to adapt to changing threats is acknowledged by many of these works.

Hostname analysis. A high-level deception technique known as squatting refers to taking advantage of users who make mistakes typing or reading website addresses. Variants include typosquatting, soundsquatting, and combosquatting. When typosquatting, attackers use

[6] It is worth noting that feature- and heuristic-based approaches are distinct because features alone do not make classifications/predictions. A feature is something like "the number of characters in the URL." Heuristics (or rules) augment features with clearly-defined logic, whereas ML models synthesize features in a way that is often difficult to explain.

[7] Interpretable ML Symposium, `http://interpretable.ml/`

domains that result from common typos of popular names [46]. Similarly, soundsquatting is using domains that sound similar to popular names, but are spelled differently [31]. Combosquatting involves domains that combine a trademark with some other word to look legitimate [22]. Szurdi et al. developed a tool which, depending on the configured mode, uses purely static or lexical features, domain WHOIS and DNS information, and possibly other information obtained by crawling, to detect squatting [46].

Password managers. Password managers can be an effective mechanism to prevent phishing and work similarly to a whitelist: first-time logins are stored with the password manager along with some or all of the URL. As long as the password manager is used consistently for data entry instead of entering it manually, the password manager should not auto-fill information into a site with a URL it does not recognize, thus preventing accidental data entry into fraudulent sites [14]. They generally rely solely on DNS when deciding to allow automatic data entry on a page, so they handle changes to underlying DNS records better than an explicit name-to-IP mapping. However, users might get frustrated when a password manager refuses to auto-fill form information in the event of a phish and force the password manager to fill the credentials manually. The role of password managers is to make data entry on legitimate pages easier, not to block data entry on deceptive sites.

2.3 Comparison to Related Work

The primary novelty which distinguishes our work is how we frame the browser's role in protecting users from threats. The traditional approach taken by the related work frames it as, **"If browsers identify a threat, they block it."** We frame it differently as, **"If browsers suspect there is a risk, they explain why."**

We have covered related work which addresses both the premise (identifying threats) and the conclusion (blocking/explaining threats, i.e. security UI). In our search, we only found one recent paper which connects both together [27], and its results are promising.

However, even though that paper's proposed warnings were preferred over current browser warnings, its warnings are still predicated on positively-identified threats rather than risk assessment. This is a fundamental limitation of the paradigm shared by all related works we found: it assumes users are willing to trust a blind determination, which we know is not correct [26, 38].

To overcome this limitation, we think a different approach is needed to help users avoid deception. The motivation for our approach is rooted in the belief that even if detection systems are 100% accurate, deception attacks will still succeed because detection alone is not enough to stop attacks. Because current browsers frame the problem as detecting threats and classifying sites as dangerous or benign, warnings derived in this way can only be so helpful, and data has shown that people click through such warnings [2, 10, 26, 54]. One reason users do so is because they have already decided that they trust the site [38] (and our results reinforce this conclusion). If it is true that users like making their own trust decisions, we believe that explaining reasons why they may be at elevated risk could be more effective than only detecting definite threats.

Users are more likely to avoid threats if they decide they are at risk instead of the browser deciding for them, which is why recent work recommends designing warnings that explain *why* the user may be at risk [26, 38, 51, 54]. One major limitation of the existing work on threat detection is the widespread use of opaque ML models. For the same reasons a doctor can't use such systems to diagnose cancer [25], we cannot use them to trigger warnings telling the user that they are under attack.

There is much to be learned from these approaches, though. We hypothesize that by informing users about possible risky interactions, users will be more confident making their own trust decisions. We will apply many similar input features and rules found in prior work toward our own approach, but instead of combining them using an uninterpretable ML system, we'll expose the outcome of individual rules to the user. This allows for specific, educational warnings that justify why the user may be at risk, which Yang et al. commend in their work: “We suggest that integrating training in the warning interface ... and explaining why warnings are generated will improve current phishing defense” [54]. It also complements prior work by handling cases when phishing detection fails in at least three scenarios: 1) the threat is broader than phishing, 2) the threat is still emerging, or 3) the user clicked through the warning. In all of these cases, our method can still inform the user of the risk they may be taking. By giving the user new information, their judgment can be augmented with the coverage already provided by the techniques of prior work, resulting in another layer of protection against deception.

Chapter 3

Risk Assessment Framework

The first part of our approach is the *risk assessment framework*, which is used to control the display of risk indicators and the text of warnings. Its function is similar to the current rules which guide today's security indicators and browser warnings, but we formalize the assimilation of new rules and unify the various roles of warnings and security indicators to make it easier for browsers to adapt as the threats of the Web continue to evolve.

Not only do we wish to preserve the benefits of current security indicators and warnings which enforce strong HTTPS and block known phishing and malware, we aim to expose deceptive sites in general even before they are blacklisted, specifically to inform the user when their interactions are at an elevated level of risk. Our framework does this by consulting a set of rules as functions which consider the user's individual context and return whether the interaction may be extra risky.

Our reasoning is backed by Reeder et al.'s conclusion from last year, "Warnings have improved to the point that additional gains in adherence rates are likely only to be made by examining contextual factors and a wider variety of users' concerns, rather than through one-size-fits-all improvements" [38]. Thus, rather than attempting to classify websites, this framework classifies interactions. Instead of a globally-applicable decision, risk assessments are personalized to an individual in their local context.

Unlike most of today's indicators and warnings, our framework lends itself to unopinionated warning design. This is intentional; we are working in a space where the threats are both broad and uncertain. What the framework must accomplish is to explain why it believes the interaction may be risky and to give the user enough information to make their own decision. Only the user knows what they are trying to accomplish and therefore whether the risk is worth it. The framework helps answer the question, “Based on what we know at the time, is the current interaction more risky, and why?” This is an important aspect of our framework: it determines when a specific interaction is *risky* given the current context, as opposed to when a website has been confirmed as a *risk* for everyone. This allows us to reveal a wide variety of threats without having to rely on them being known and reported first.

This approach has several benefits. Its privacy-preserving design doesn’t require calling out to external resources or third-party services. Unlike blacklists, its defenses are immediately effective against new threats. And because it operates locally, it is always available, even if network connections are blocked. Its reliance on the user’s browsing history makes it robust against many tricks scammers would use to manipulate centralized authorities.[1] In addition to categories of risk that current browsers already protect against such as insecure connections, malware, and popups, the risk framework supports categories such as resource abuse, monetary theft, homoglyph and typosquatting attacks, history stuffing, and more.

The actual implementation is flexible, so developers can make adjustments based on requirements or experimentation. For example, a web browser vendor may want to protect its users from pages with deceptive billing practices.[2] Other protections that are found to be helpful (such as blacklists) can be added as well.

3.1 Framework

The risk assessment framework evaluates a set of rules (decided by the implementer) to determine if an interaction is risky.

An implementation of this framework is a function that takes as inputs at least three parameters: (1) the context of the user’s current visit, (2) the browsing history, and (3)

[1] This decentralization is important. Centralized enforcement potentially sets a dangerous precedent for censorship, especially as we venture into blocking threats that are not as clearly defined.

[2] Google Chrome started doing this recently. See “Notifying users of unclear subscription pages”, `https://blog.chromium.org/2018/11/notifying-users-of-unclear-subscription.html`

the interaction. These inputs are necessary to support the evaluation of rules, and web browsers—acting as user agents—have this information readily available.

Context of visit. The context of a visit consists of what the browser knows about the user's current visit to the site. This would include URL, page title, time of visit, duration of visit, interactions up to now, HTTP redirects, HTTP response headers, HTTPS validity, and how the page was accessed (for example: address typed, address pasted, link clicked from web page, link clicked from email, opened bookmark, submitted form, etc.).

Browsing history. The browser history includes all sites a user has previously visited, visit durations, lists of interactions, etc. In some situations, a user's browsing history is crucial for determining risk, since it is unlikely that an average user will repeat risky activities or revisit malicious sites. Presence in the history adds legitimacy to a site and often reduces the risk of an interaction, but we do not recommend treating this as a hard-and-fast whitelist. Empty browser histories could be augmented with trusted top-sites lists.

Interaction. For our purposes, interactions are anything performed or allowed through a user agent. They can be implicit or explicit. Possible interactions include loading, scrolling, selecting, copying, clicking, typing, entering data, submitting, downloading, and consuming resources.[3] Some will be more useful than others, depending on the implementation. Compound interactions, which are combinations of one or more interactions over time or in sequence, may also be defined, for example: reading as a function of time, scrolling, and selecting. Not all interactions are equal in magnitude and scope; reading a phishing page is not usually risky, but entering data into a phishing page is. Users initiate some interactions (loading, scrolling, etc), while websites initiate others (downloading, consuming resources, etc).

To perform a risk assessment, these inputs are fed into each rule to check for elevated risk. If any of the rules yields a positive return value, the interaction could be considered risky.

[3] Some browsers do show errors when a tab hangs or crashes, but this is different from treating high memory or CPU utilization as a threat. For example, a malicious webpage that mines cryptocurrency might not crash the tab, but the user should be warned about this interaction.

To ensure the most correct and complete picture of the current interaction, risk assessments should be executed in real-time as the interactions happen (or just before they are allowed to happen).

The output of the risk function is an *assessment*, which consists of an overall conclusion (boolean *Risky* or *Not risky*) and explanatory text (string) that justifies the decision. The text should be able to explain why the interaction is or is not risky. In practice, returning a list of reasons corresponding to each rule may be helpful in gaining insight into the overall decision, even if not all of that information is always shown to the user. The indicator should appear as soon as the risk assessment finishes if there is a positive result, optionally with an automatic pop-out of the warning text to grab attention or block a risky interaction.

3.2 Criteria for Risk Assessment Rules

Rules define the scope of risky interactions. They do not stop deception attacks, but they constrict the ways attackers can be deceptive if they want to avoid a “Risky” assessment.

Because this framework is flexible, rules are ultimately up to the implementer. However, in keeping with the design goals of the risk assessment framework, rules should have these qualities:

- **Local.** The rule can be evaluated locally and within the context of the browser, without needing remote or third-party resources. This is necessary for fast evaluation and to avoid attacks on the network which could block access to vital resources.

- **Privacy-preserving.** Evaluation of the rule does not reveal or leak data or metadata to any third party, and does not facilitate tracking or fingerprinting.

- **Robust against manipulation.** The rule cannot be easily duped by attackers to be circumvented without triggering another rule that is designed to detect and flag said manipulation.

- **Independent.** The rule should not depend on the output of another rule.

- **Accurate.** The rule expresses an elevated level of risk.
- **Explainable.** The rule can be reasonably explained in layman's terms so that it can produce informative warning texts.
- **Simple.** The rule is easy to describe and can be evaluated efficiently, without using excessive power or blocking the thread of execution.

While accuracy is a criteria, it may be difficult to measure for a given rule. Some rules use browser history and user interactions as an input, thus accuracy can vary per user, and gathering per-user browsing history and interactions for a study has strongly negative privacy implications. Thus measures of accuracy are limited to rules (or parts of rules) that identify deception in site characteristics (e.g. URL), though this is imperfect.

3.3 Examples of Rules

To demonstrate the feasibility of the risk assessment framework, we have curated an example set of rules which can expose many suspicious technical characteristics that browsers do not currently warn about. Our goal is *not* to prove that our rules are the best of all possible rules, nor that they are a complete set. Rather, this is a first cut at a feasible set of rules which we use to evaluate the perceived helpfulness of warnings produced by the framework and thus demonstrate the utility of our framework as a whole. Even though our rules are only examples, we chose them because they express patterns that are inherent with interactions that are considerably more risky, or are common among sites that are trying to be deceptive.

We primarily derive our choice of rules from prior work in the Internet measurement and engineering communities. To shape the formation of these rules, we examined past research, current browser warning designs, blog posts from security experts, alerting techniques from Facebook's prominent certificate transparency (CT) monitor,[4] and various patterns

[4] `https://developers.facebook.com/docs/certificate-transparency/`

used by about 50,000 sites on blacklists spanning 2 months from DNS-BH[5] and PhishTank.[6] In some cases, we tested rules on a collection of 35 million records from CT logs. After applying engineering judgment, common sense, and our criteria listed above, we settled on our example rules.

This is similar to how existing browser warnings were developed. For example, browsers warn when HTTPS connections use weak security, but what exactly constitutes weak security is based on the judgment of developers and security experts.[7] Note that we did not use techniques from deep learning systems, which can analyze many features and are often effective [1, 55], because their models are opaque and their predictions do not usually come with explanations.

These rules are not listed in any particular order of priority or severity, and the correctness of the framework is not affected by rule order. For convenience, we've grouped our rules into arbitrary categories, but rules need not be limited to these categories.

Connection-based Rules

Plaintext HTTP This rule triggers when the page is loaded over an unencrypted connection, i.e. it is not using HTTPS. The perils of plaintext HTTP are well-understood and widely accepted, and this is already a significant part of the logic controlling existing security indicators.

Invalid, weak, or partial HTTPS When a page or any of its resources are loaded over an HTTPS connection with broken security properties, or if any resources are loaded over plain HTTP, this rule triggers, similar to the logic of current security indicators. Users are warned if any of their connections provide weak confidentiality, authenticity, or integrity guarantees.

[5] `http://www.malwaredomains.com/`

[6] `http://phishtank.com/`

[7] See `cert_status_flags.cc` and `ssl_error_handler.cc` in the Chromium source code for an idea of how complex the mapping from technical attribute to threat is, at `https://chromium.googlesource.com/chromium/src`.

History-based rules

Sequential subdomain labels The domain of the page being visited consists of a sequential combination of subdomain labels that comprises the right-most labels of a hostname in the user's browsing history [28], including any replacements of the dot characters (label separator). For example, visiting a host `paypal.com.secure-signin.net` would trigger this rule if `paypal.com` is already in the browsing history. We did work to verify this rule is helpful. In our site blacklist, 199/655 of the phishing URLs that mentioned "paypal" also used "paypal.com" or "paypal-com". Over the course of just one week, alarms from Facebook's CT monitor configured for `apple.com` and `google.com` reported 469 hostnames suspicious in this manner being issued new TLS certificates (many of which did not appear on the updated blacklists, months later). We simply apply logic like what Facebook's CT monitor uses to the browser.

Adding or removing hyphens Adding or removing a combination of hyphens to or from the site's hostname matches another hostname in the user's browsing history. This was the tactic of at least one significant deceptive site in 2018 which spread widely on social media. Its manipulation of technical properties such as domain name and site design reportedly caused many readers emotional distress and even altered their belief systems [44, 53].

Typosquatting The hostname of the page being visited resembles a hostname in the browsing history in a way that could be attributed to a typing error [46]. This rule's obvious function is to warn against sites that are taking advantage of natural mistakes which would be difficult to notice if the typosquatting site is deceptive. Research has shown that there is a wide variety of domain squatting attacks in the wild [22, 31, 46].

Visited only recently and for short time This rule is preemptive. Anticipating that deceptive sites may try to manipulate a user's browser history to gain false legitimacy and thereby dodge certain rules in this framework, this rule triggers when a site being visited appears in the browsing history only recently and has a short cumulative visit duration. We expect that most legitimate site visits aren't much shorter than about 2 seconds.

Host-based Rules

Excessive hyphens The domain name contains more hyphens than most website owners would prefer to have in their name. For similar reasons as with the previous rule, this rule triggers when hyphens are used by sites trying to make their domain name recognizable but which are not desirable first choices by site owners. Like some of the other rules, it is difficult to claim exactly how many sites are using excessive hyphens in an attempt to be deceptive, but as a point of reference, 4.2% of sites on our phishing and malware blacklist contained 3 or more hyphens. By comparison, only 1 of the Tranco [24] top 50,000 have 3 or more hyphens, and of the whole top million, only 3 do.

Many labels in domain name The domain name contains more labels than most user-facing websites normally contain, ignoring eTLD [29]. It is difficult to consult a "ground truth" data source to know exactly how many sites use excessive labels for deception outside of phishing and malware, but as a point of reference, 9.2% of sites on the blacklists we used had 4 or more labels in the domain name (not including eTLD). Only 0.028% (14 sites) of the Tranco[8] [24] top 50,000 had 4 or more labels in the domain name.

Long domain name The domain name is longer than that of most common, user-facing websites. Normally, domain names are desired because they are easier to type, remember, and recognize, than IP addresses. Domains which are intended only for use as link targets, such as those used to spread deceptive sites, need not be short, memorable, or even recognizable since most users do not know how to interpret URLs [45]. Longer domain names are often part of an attacker's threat model because they obfuscate the true identity of the site. We examined the average length of hostnames from our sample of CT log data and the Tranco

[8] Available at https://tranco-list.eu/list/N73W.

top million (both of which we presume to be mostly benign), together with those on the public blacklist. The average hostname length in the CT log and blacklist were both 20, but the average hostname length in the Tranco list was only 15. For contrast, the average length of sites on the blacklist containing a recognizable site name such as PayPal, Google, or Apple was 31. When choosing a length boundary, consider that many hosts in CT logs are not accessed via web browsers but most sites in blacklists and the Tranco list are. By constraining the length of domain names to something reasonably comprehensible, attackers will be forced into a smaller space of possibilities when choosing deceptive domain names if they want to avoid a risky assessment.

Behavior-based Rules

Excessive redirects The URL changed 2 or more times before settling [29, 33]. Although HTTP headers are the primary mode of redirecting, other kinds of redirects should be considered, such as those from `<meta>` tags or JavaScript, which are executed within a certain duration of the source being downloaded and evaluated. This rule triggers for sites that maliciously use redirects to inject tracking, misdirect users, or stuff the history. Redirects from HTTP to HTTPS without a change of hostname or path don't need to be counted.

First visit by link This is the first time the user has visited this site, which was accessed by clicking a link. As this is almost always the case when unknowingly accessing a deceptive site, this rule can be highly effective in warning users before they engage with a new site for their first time, effectively adding a TOFU security model to web browsing, much like what SSH relies on. Because this rule does not depend on the hostname like several other rules, it is immune to clever or non-obvious, deceptive domain names that might not trigger the other rules. However, public computers or browsers without much browsing history may yield lots of positives from this rule. It may be advisable to only engage this rule after a regular browsing history has accumulated. To further reduce appearing too often, this rule might be restricted to riskier interactions such as data entry or downloading files.

Hostname language mismatch One or more characters contained in the domain name are not common in the page's language (according to either the HTTP Content-Language response header or user's language preferences in their browser, which are sent in the Accept-Language header). Operating under the assumption that IDNs should be used to match a site's or visitor's native language, this rule attempts to warn when sites use IDNs inappropriately, like when using Unicode characters to achieve visual similarity to a targeted domain name [17].

As a further consideration, rules might optionally be constrained in more ways than are described here. In practice, some rules might be more helpful only in conjunction with certain interactions or presence of the site's hostname in the browser history. For example, the rule "First visit by link" need not trigger unless the user starts to enter information into the page, to help reduce warning fatigue [2], as opposed to being predicated solely upon the site which is being visited.

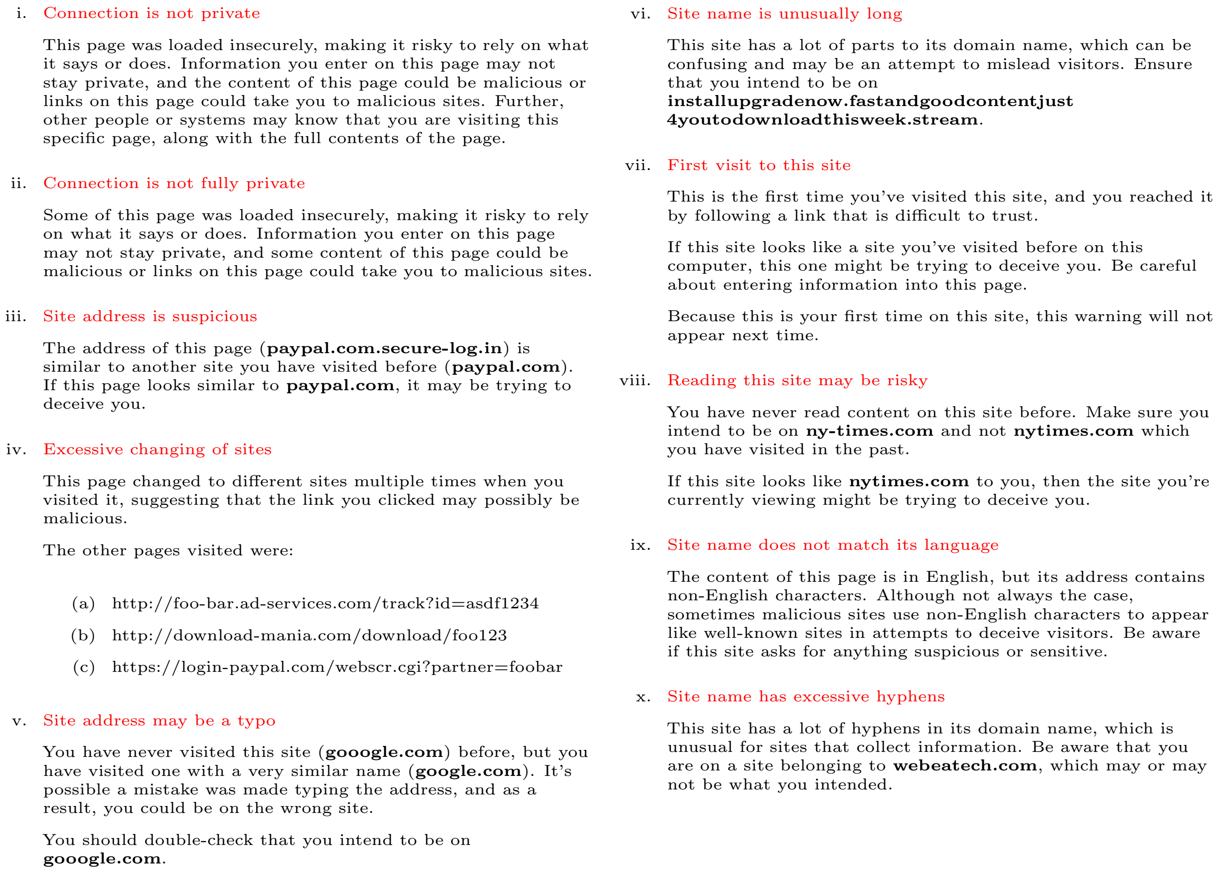
3.4 Evaluation

We ran an IRB-approved survey on Amazon Mechanical Turk among highly-rated workers to assess the perceived usefulness and likely responses of participants to example warnings generated and explained by some of these rules. We showed participants a sample warning text and asked three questions:

1. Would you want to be notified when you are (unknowingly) in a situation described by this warning?

2. Would you find warnings like this helpful to you while using the Internet?

3. Which of the following would you be most likely to do in response to this warning?



i. Connection is not private

This page was loaded insecurely, making it risky to rely on what it says or does. Information you enter on this page may not stay private, and the content of this page could be malicious or links on this page could take you to malicious sites. Further, other people or systems may know that you are visiting this specific page, along with the full contents of the page.

ii. Connection is not fully private

Some of this page was loaded insecurely, making it risky to rely on what it says or does. Information you enter on this page may not stay private, and some content of this page could be malicious or links on this page could take you to malicious sites.

iii. Site address is suspicious

The address of this page (**paypal.com.secure-log.in**) is similar to another site you have visited before (**paypal.com**). If this page looks similar to **paypal.com**, it may be trying to deceive you.

iv. Excessive changing of sites

This page changed to different sites multiple times when you visited it, suggesting that the link you clicked may possibly be malicious.

The other pages visited were:

(a) http://foo-bar.ad-services.com/track?id=asdf1234

(b) http://download-mania.com/download/foo123

(c) https://login-paypal.com/webscr.cgi?partner=foobar

v. Site address may be a typo

You have never visited this site (**gooogle.com**) before, but you have visited one with a very similar name (**google.com**). It's possible a mistake was made typing the address, and as a result, you could be on the wrong site.

You should double-check that you intend to be on **gooogle.com**.

vi. Site name is unusually long

This site has a lot of parts to its domain name, which can be confusing and may be an attempt to mislead visitors. Ensure that you intend to be on **installupgradenow.fastandgoodcontentjust 4youtodownloadthisweek.stream**.

vii. First visit to this site

This is the first time you've visited this site, and you reached it by following a link that is difficult to trust.

If this site looks like a site you've visited before on this computer, this one might be trying to deceive you. Be careful about entering information into this page.

Because this is your first time on this site, this warning will not appear next time.

viii. Reading this site may be risky

You have never read content on this site before. Make sure you intend to be on **ny-times.com** and not **nytimes.com** which you have visited in the past.

If this site looks like **nytimes.com** to you, then the site you're currently viewing might be trying to deceive you.

ix. Site name does not match its language

The content of this page is in English, but its address contains non-English characters. Although not always the case, sometimes malicious sites use non-English characters to appear like well-known sites in attempts to deceive visitors. Be aware if this site asks for anything suspicious or sensitive.

x. Site name has excessive hyphens

This site has a lot of hyphens in its domain name, which is unusual for sites that collect information. Be aware that you are on a site belonging to **webeatech.com**, which may or may not be what you intended.

Figure 3.1: Examples of warnings generated by some of rules which we trialed in our survey.

Each participant was shown one randomly-sampled warning text (Figure 3.1), and each warning text was sampled approximately 40 times. Our choice of 40 is grounded in both research and precedent; Virzi found that 90% of usability issues are uncovered with 10 subjects [48], and previous studies of a similar nature used 50-70 subjects [42]. The full survey is shown in Appendix A.1.

We elected to ask the participant about the warning text associated with a rule rather than about the rule directly, because the rules require some specific technical knowledge to understand, while the warnings are designed for anyone to be able to read; and we assume that the rules will be carefully chosen by experts in the field so that average users will not need to understand them. We did not explicitly explain that the warnings could be wrong, i.e. false positives, because users will not be told this when using their web browser; instead

	Want to be notified	How helpful is the warning					Most likely reaction					
Rule	Yes	Very unhelpful	Unhelpful	Neutral	Helpful	Very helpful	Investigate further	Leave	Leave and return	Be skeptical	Avoid entering info	Ignore warning
Plaintext HTTP	91.1%	4.4%	0%	11.1%	37.8%	46.7%	16.7%	45.2%	7.1%	9.5%	19.0%	2.4%
Partial HTTPS	95.1%	0%	4.9%	2.4%	63.4%	29.3%	26.8%	29.3%	2.4%	2.4%	39.0%	0%
Similar URL	97.7%	0%	0%	4.7%	39.5%	55.8%	37.2%	44.2%	0%	9.3%	9.3%	0%
Excessive redirects	94.1%	0%	0%	8.8%	44.1%	47.1%	17.6%	58.8%	5.9%	5.9%	11.8%	0%
Typosquatting	87.8%	6.1%	0%	10.2%	49.0%	34.7%	34.7%	42.9%	6.1%	6.1%	8.2%	2.0%
Long domain name	77.1%	2.9%	8.6%	17.1%	45.7%	25.7%	20.0%	45.7%	8.6%	11.4%	5.7%	8.6%
First visit	83.9%	3.2%	6.5%	9.7%	48.4%	32.3%	35.5%	25.8%	6.5%	0%	19.4%	12.9%
Reading site	88.9%	5.6%	2.8%	8.3%	44.4%	38.9%	27.8%	44.4%	5.6%	11.1%	0%	11.1%
Language mismatch	86.0%	0%	6.25%	25.0%	25.0%	43.75%	23.3%	30.2%	2.3%	23.3%	14.0%	7.0%
Excessive hyphens	86.0%	4.7%	7.0%	11.6%	60.5%	16.3%	41.9%	20.9%	2.3%	9.3%	14.0%	11.6%

Table 3.1: Results from the survey about warnings pertaining to various rules.

of suggesting the absolute presence of a threat, the warnings are carefully phrased to inform users to allow them to make their own judgment.

We hypothesized (1) that participants would prefer to be notified when they are in situations typical of deceptive sites, (2) that they would find the warning text helpful in making a decision, and (3) that their reactions to the warning would be reasonable in magnitude given the described risk.

Results are shown in Table 3.1. Across all warnings, 89% responded (to Question 1) that they would want to be notified when they were unknowingly in the situation described by the warning text. Although certain warnings were less popular than others, the vast majority of participants would like to be informed if one of these rules triggered. Although we aren't sure from this survey alone why the results were so positive, we have two theories: users might like to be informed and make their own decision (some results from our later user study back this up), or participants did not consider that warnings might appear even when they are not actually at any risk. We discuss the difficulty of false positives later in this paper.

The least popular warning was associated with the "Long domain name" rule, to which 22.9% responded that they would not want to be notified in that situation. Upon reviewing this, we realized a slight error in this warning text, which was written for the

"Many labels in domain name" rule, but the domain name chosen for the warning was simply a long domain without many labels.

Some of the warning texts contained domain names that were pulled from actual risky scenarios, such as phishing names for popular sites. These names could easily look suspicious, especially to a trained eye. To help ascertain whether the results were skewed toward our hypothesis by our choice of domain names, we re-ran the survey on those warning texts with a single set of changes: we replaced the suspicious-looking domain names with values that are legitimate so that experts at least would conclude that there is no actual risk (i.e. false positives). For example, the warning based on the "Excessive hyphens" rule was changed to use the domain of a real German pizza shop. We found that there is no statistically significant difference between the results of the two surveys. From this, we think our initial results were not biased toward positive answers due to the names being suspicious.

Across all warnings, 83% of participants responded that warnings like these would be either Helpful or Very Helpful to them (Question 2).

We also asked participants what they would most likely do in response to seeing the warnings (Question 3). The results suggest that participants think they are able to at least somewhat assess their situation and make reasonable decisions accordingly. For example, 45% participants who were shown a warning for "Plaintext HTTP" would leave the site and not come back, while only 19% said they would simply avoid entering information into the site. Conversely, 29% of participants who were shown a warning for "Invalid, weak, or partial HTTPS" said they would leave the site and not come back, while 39% said they would avoid entering information into the site. This seems reasonable since, in practical terms, the former depicts a more risky situation than the latter, and entering data into a site can generally be considered more risky than simply visiting a page.

Chapter 4

Risk Indicator

Our second contribution is a browser indicator (comprised of an icon, label, and togglable text) that is designed for warnings (or “tips”) which grab attention, educate users, maintain parity on both desktop and mobile screens, and is hard for malicious sites to mimic with high fidelity. It differs from current browser indicators primarily in its objective: rather than informing about connection security or trying to prevent the user from continuing, the goal of the risk indicator is to bring something unusual to the user’s attention, entice them to click to get more information, and to help them decide for themselves if they are at risk.

Unlike current connection security indicators which are often incorrectly associated with a site’s safety [38], our risk indicators build on these preconceived notions by providing information about the user’s interaction with the site. In addition, the risk indicator shows no positive affirmations, uses more informative (and less opinionated) description text, and provides a UI capable of replacing full-page interstitials, including the ability to show the warning text automatically when a serious or blocking issue needs attention.

The risk indicator can share an implementation on mobile and desktop browsers due to its low screen space requirements. We believe it also holds promise as an alternative to full-page interstitial warnings (Figure 2.3) which can be easily mimicked by malicious sites, because the risk indicator and its warning text do not appear solely in or over the content area of the browser. Upon clicking a risk indicator, the warning text appears in a container element separate from the page which overlaps at least slightly with the URL bar (Figure 1.2)—current versions of Firefox and Chrome already do likewise with security

ID	Design
!ST	⚠ SAFETY TIP
iST	ℹ SAFETY TIP
!R	⚠ RISK
?R	❓ RISK
!PR	⚠ POTENTIAL RISK
?PR	❓ POTENTIAL RISK
!C	⚠ CAUTION
?C	❓ CAUTION
!A	⚠ ADVICE
iA	ℹ ADVICE

Table 4.1: Different icon and label designs we trialed for the indicator.

indicators (Figure 2.1). For warnings so severe that user actions need to be blocked before continuing, browsers can still implement this blocking behavior and automatically pop-out the warning text without using the content area of the page.

4.1 Methodology

Like current indicator designs, the *risk indicator* icon and label appears to the left of the URL. The icon and label only appears when the risk framework yields a "Risky" assessment. It does not show a padlock, EV certificate subject name, or any other positive reinforcement; trust decisions are left up to the user. The objective, then, is to effectively convey the information so that the user can make an informed decision about how risky their interaction is.

In designing a new indicator, we started with previous research and current browser indicator designs, then explored ways to communicate risk rather than security. In keeping with prior work [11], we chose red as the color but varied the icon and label in a Mechanical Turk survey to inform the best combination (Figure 4.2) for an eventual user study.

We started with ℹ, ⚠, and ❓ icons. Variations on the label included SAFETY TIP, RISK, POTENTIAL RISK, CAUTION, and ADVICE. We surveyed 10 combinations (Table 4.1), some of which are not so different from Firefox's newest indicator design (Figure

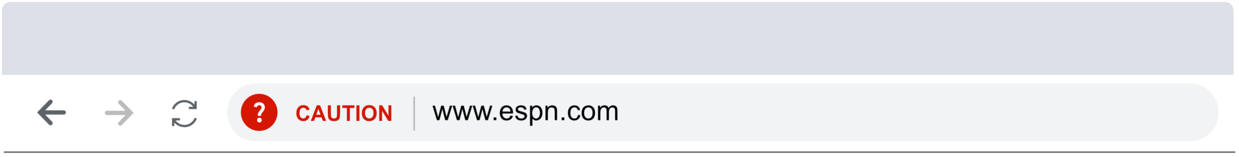


Figure 4.1: One of the example risk indicator designs we surveyed.

2.5) [51]. Our objective was to find the combination which got the users attention so that they would want to click the indicator, but which did not overly alarm them.

Figure 4.1 shows an example screenshot of a risk indicator we trialed in our survey; the full list of actual images is in Appendix A.2. We chose to show the risk indicator in place next to a sample domain name, to ensure that participants would understand the context in which this would appear. The domain name was carefully selected to be one that is fairly well-known and trusted among the majority of the demographic. This way, we can be more certain that responses which indicate alarm or suspicion are more likely due to the indicator and not the domain name.

4.2 Evaluation

We prepared a brief, IRB-approved, between-subjects survey that we ran on Amazon Mechanical Turk about the various risk indicator designs (Table 4.1). Each indicator style was sampled about 40 times. We showed an image of the risk indicator and explained that it was at the top part of a web browser. We then asked four questions:

1. What do you think the indicator (pointed to) means?
2. Suppose you visited a website and this indicator appeared. How safe you would feel about the current website?
3. Which of the following would you be most likely to do after seeing the indicator?
4. What do you think the source of the indicator is?

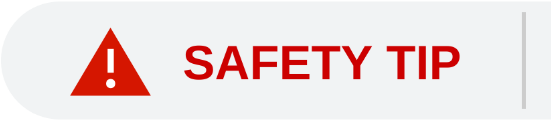


Figure 4.2: The icon and label we felt was the best choice among the designs we tested.

Our objective was to inform the design of the user study by finding the indicator that yielded the most desirable results. Ideally, the presence of a risk indicator would (1) make users feel less safe (although we anticipated that some participants might feel *more* safe if they believed the browser was protecting them, but the results don't suggest this); (2) would elicit a moderate response—preferably clicking the indicator to get more information, but not leave the site and never come back or ignore the indicator entirely; and (3) would be most recognized as originating from the browser and not the website, a virus, or other sources.

Overall, the indicators caused 84% of respondents to feel Unsafe or Very Unsafe, while 12.5% felt no different about the site with a risk indicator. A few answered that they felt more safe with some indicators, but this was only 3%.

When asked what the respondent would do upon seeing their indicator, the results varied (Figure 4.3). Across all indicator designs, the top choice was "Leave the site and not come back," an answer we deem fairly extreme, at 49%. The preferred answer we would have liked to see is "Click it / get more information" because that will help the user make informed decisions before deciding to permanently leave a site. Two indicator designs, both using the "Safety Tip" wording, resulted in more than 50% of respondents who would have clicked to get more information, and we consider these to be our top contenders.

From these results, we don't claim that people will actually take these actions for certain indicators; we simply wish to show that there are differences in how users perceive them and in what they think they would do.

As for the source of the indicators, 59% correctly thought that the source of the indicator was the web browser, followed by 20.25% thinking the website, and 16.25% thinking a virus or malware, even after being told to look at the top part of a web browser. It is

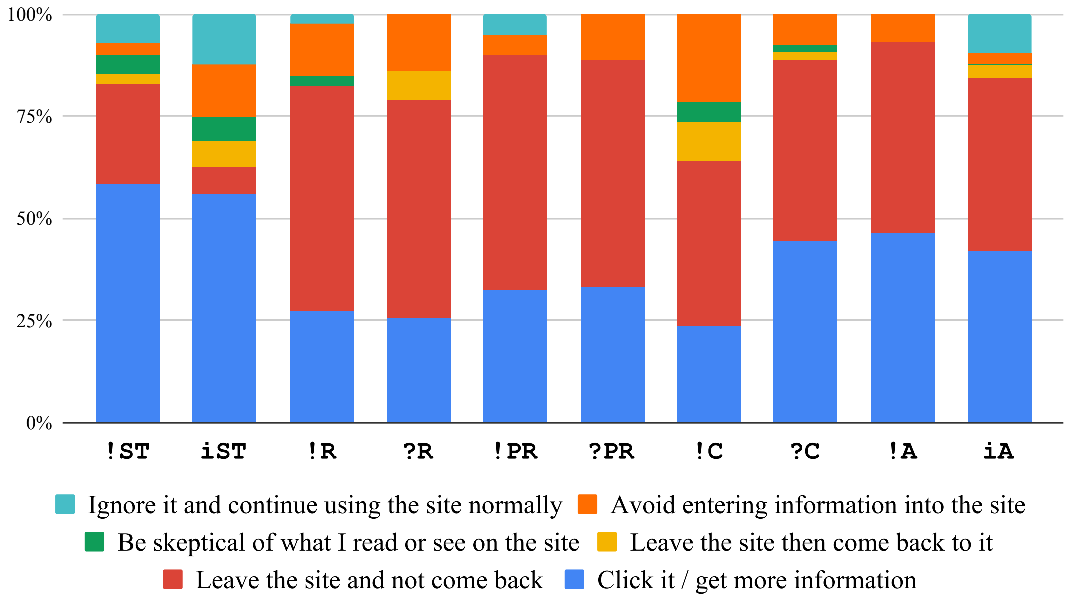


Figure 4.3: "Which of the following would you be most likely to do after seeing the indicator?"

unclear whether the respondents chose Website or Virus/Malware because they thought that something about the website or a virus triggered the browser to show the warning, thus causing it to be the "source" of the indicator, as if the browser was detecting that and protecting them.

Ultimately, the choice of indicator we used for the user study was subjective, as there were two or three candidates that could have worked well. We settled on ⚠ SAFETY TIP because it elicited responses in the most desirable range: respondents suggested that they would feel somewhat unsafe after seeing it, would be most likely to click it, and were correct about the source of the indicator being the web browser. It also has a positive sentiment and is the least ambiguously-worded, while drawing attention to the urgency of one's safety, as opposed to simply an informational message.

The indicator with the fewest "Leave the site and not come back" answers was ⓘ SAFETY TIP, but it also had one of the highest "No different" answers to Question 2. We think either of these indicator designs could be good choices.

Chapter 5

User Study

We conducted an IRB-approved, within-subjects user study in which we invited 50 participants into our lab to use a simple web browser we made which exposed them to three treatments: (C) current browser security indicators, (R) our new risk indicator which pops-out the warning text when you click it, and (RA) our risk indicator that automatically pops-out the warning text; and then asked them to fill out a survey with which we collected both quantitative and qualitative data. Our objective was to understand if users were able to make better trust decisions with the risk indicator, and whether users felt that warnings like this were helpful toward making them aware of risky situations.

The demographics for our study consisted of university students recruited through advertisements across campus. The participants followed a near-normal distribution when self-reporting their skills and understanding of computers and technology.

5.1 Methodology

We built a simple, single-tab web browser that embedded a Chromium webview below a familiar-looking URL bar area which consisted of back and forward buttons, a stop/refresh button, and a rich text box for the URL which we modeled after Google Chrome 70. We did our best to preserve common Chrome functions with basic fidelity, including searching from the address bar. The browser logged relevant, high-level user activities such as clicking an indicator, dismissing its warning text, entering data into a form on a page (*sans* the actual data), navigating to a page, and switching treatments.

Before beginning, participants were told that the purpose of the study was to evaluate new kinds of browser warnings, and we showed them a picture of where indications of potential risks would appear. We instructed participants to approach each task as if they were on their own computer, but because prior work has shown that role playing biases participants' security behavior to be less secure [42], we also instructed them to use their real account and payment information to complete tasks. They were told that they could skip any steps if they felt uncomfortable.

Each participant repeated the task list three times, receiving a different treatment on each iteration, with the three possible treatment sequences being ordered round-robin. Due to effects of conditioning, we decided to only examine the effectiveness of each treatment from participants who received that treatment first; but we wanted them to experience the other two treatments so they could answer questions comparing them in the survey.

Treatment C was the current browser security indicator design, which shows a padlock on HTTPS pages and displays the scheme portion of the URL, even for `https://`. Treatment R was our risk indicator, which appears only in the presence of potential risks and which hides the URL scheme for `https://` but displays it for others. Treatment RA was also our risk indicator, but with an automatic pop-out of the warning details (as if the user had clicked it right away). Risk indicators and warning pop-outs used an easing function over .3 seconds which resulted in a subtle "pop" effect (the UI element was scaled from 0% to 110% for the first half of the animation, then reduced to 100% at the animation's end)—we felt this natural motion grabbed attention without being too distracting.

Participants were given four tasks, two of which simulated risky scenarios. The first task was benign: log into their university account and check their class schedule. The second task was risky: search for university news, click a search result that led to a press release about a change to a controversial university policy, and read the article. Although this article looked identical to a legitimate news page, it was fabricated for the study and was written to be convincing. The third task was benign: go through checkout on a popular e-commerce site, entering and saving their actual credit card information, but stop short of completing a real purchase. The fourth task was risky: log in on either a popular payment processing website, a popular email service provider, or begin creating an account on a personal finance reporting website. The full instructions gave participants the specific sites to visit and how to access them.

To simulate risks on the deceptive tasks, the URLs displayed in the address bar were manipulated. On Task 2, participants were shown a locally-sourced HTML page that looked like the real thing, and were informed after the study that the article was fabricated. On Task 4, participants were on the real site even though the URL looked phishy. We built in fail-safes to prevent accidental form submission on any task that could have side effects such as completing a purchase or creating an account. Due to these design measures, participants were never exposed to real risks.

Our hypotheses were that (1) users complete fewer risky interactions in the presence of risk indicators than with the current browser indicators; (2) in the presence of a risk indicator, users understand that their actions may be risky; (3) after clicking a risk indicator, users understand the specific risk they are taking; and (4) when users complete risky interactions, it is based on understanding the risk and deciding to proceed anyway.

5.2 Results

We obtained two sets of data from the user study: an exit survey which contained both qualitative and quantitative questions, and an activity log which we analyzed to measure behaviors. All open-ended responses were jointly coded by multiple researchers in our lab using a conventional content analysis approach [18]. Our primary takeaways from this study are:

- Subjects generally think the risk indicator warnings are more helpful than unhelpful.

- The majority of subjects preferred Treatment RA.

- The presence of risk indicators can be correlated with fewer risky interactions.
- Subjects have varied recollections of past experiences with browser warnings, but the vast majority remember dealing with them previously.
- Subjects report more confidence in assessing their own risk with the help of risk indicators than without it.
- The most requested improvements to risk indicators are: (1) a way to get more information, (2) more assertive, clearer warning texts, and (3) the ability to customize warning behavior.

Helpfulness. When told to consider that warnings could be wrong sometimes, 76% of participants responded that risk indicators are generally Helpful or Very Helpful. 10% said they were No Different, and 14% thought they were Unhelpful. We found that the helpfulness of warnings may depend on the risk being indicated. About Task 2, 70% responded that the warnings were at least Helpful, and for Task 4, 76%. 20% mentioned that they did not agree that they would be at risk in real life when the risk indicator appeared. Although some said this was because their email account held nothing sensitive, etc., warning texts could probably be written to be more convincing.

Preferred treatment. There was a clear favorite of the three treatments. 58% of the participants preferred treatment RA because of its brief explanation of the situation, and they *"actually noticed it."* 28% preferred R because it was *"still noticeable"* but *"not annoying."* The remaining 14% preferred current security indicators mainly for two reasons: they liked the positive affirmation of safety, and they were used to looking for the lock icon. Interestingly, multiple participants mentioned an *"unlocked"* icon, even though no major browser uses one.

Mitigating risk. We measured when participants entered data into the "deceptive" page on Task 4 and found a statistically significant result based on the first treatment they received.

Treatment	Preferred by	Representative Quotes
C	14%	“Having the lock constantly there helped me feel more secure.” “I’m used to it, and I already check for it.”
R	28%	“It feels more descriptive than the current security indicator, so I can be better informed while I browse.” “It warns me of potential errors or problems with the URL without being disruptive.”
RA	58%	“It catches your attention and explains why the site is risky.” “If I’m busy multitasking or not paying close attention, there are times I may have missed the risk indicator with no pop-out.” “...the automatic pop-out caught my attention and made me pay more close attention to the risk.”

Table 5.1: Treatments reported as most preferred by participants, along with quotes representative of each class.

Treatment C = 100% data entry. Treatment R = 75%, and Treatment RA = 71% ($p = 0.032$ Fisher’s exact, $V = 0.345$). We then followed their behavior through the next two treatments. For treatment cycle C-R-RA, data entry rate was 100-89-65%. For R-RA-C, 75-69-81%. For RA-C-R, 71-62-62%.

Past experiences. 90% of participants wrote about their past experiences with browser warnings. Some people chose to heed warnings specifically because they didn’t trust how they got there, or the task was frivolous enough that they lost interest. When ignoring or bypassing warnings, the vast majority of respondents said they did so because they trusted the site they were on, assumed they were on their intended site, or weren’t entering information anyway. Some described basing their decisions on how serious the warning read or on how sensitive or important their task was (e.g. logging into a junk email account is not a high-stakes activity). Not only does this back prior research [38], it backs the notion adopted by our approach which is that warnings should help users make their own trust decisions based on their current situation, rather than attempting to block tasks entirely.

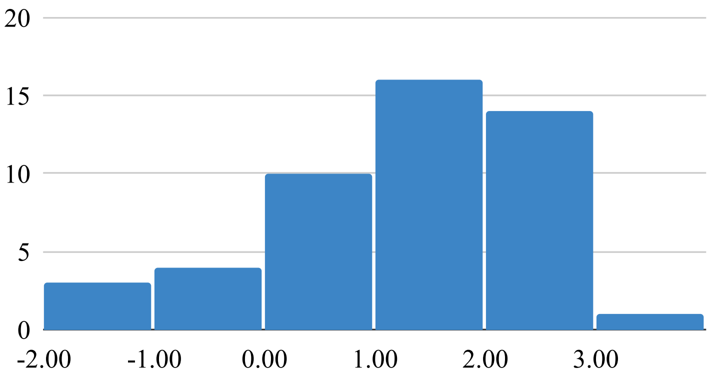


Figure 5.1: Histogram depicting individuals' shift of confidence from judging their risk without risk indicators, to doing so with risk indicators (on a scale of 1-5). A positive change represents an increase in confidence with risk indicators.

Confidence. Participants were asked how much they agreed that they would be confident in assessing their own risk, first without risk indicators, and second by investigating a warning from risk indicators. On a scale of 1-5 (Disagree-Agree), the average agreement for the former was 2.4 and, for the latter, 3.2. Figure 5.1 is a histogram of the individual differences on this scale, exhibiting a statistically-significant trend ($p < 0.001$ Fishers exact, $V = 0.481$) to being more confident with risk indicators.

Improvements. By far, the single-most requested feature (30%) is to offer advice on how to investigate the potential risks, like a "Learn more" link. Some participants even mentioned wanting this when recalling their past experiences with browser warnings, adding that current "Learn more" articles are too generalized. Overall, participants seemed eager to learn at a level they could understand, and we agree with Reeder et al. that "there may be an opportunity for SSL warnings and browsers more broadly to better educate users about the security properties of SSL" [38] (and about deceptive sites generally). Other common requests were to clarify or strengthen the warning text and to make warnings customizable.

Chapter 6

Discussion

We think there are several interesting points of discussion for this approach and inferences to draw from the results.

6.1 False Positives

Risk indicators probably will appear on benign sites, but they are also likely to reveal threats that current browser indicators miss because risk indicators support more scenarios that are not considered by browsers. Multiple respondents in the user study even said they would prefer false positives over false negatives: *"I'd rather have a false positive than have something slip through the cracks."* and *"These threats were serious and I didn't want to miss them accidentally."* Others, of course, remarked that false positives would be annoying, saying, *"It would drive me nuts. I'd honestly use a different browser."*

In designing our work, we found it difficult to concretely define false positives and false negatives, and we realized that mapping risky interactions to actual threats is an imprecise science. Even current browser warnings have this difficulty [38]. For example, in most browsers, an expired TLS certificate blocks page load and invokes an interstitial warning. Unless the client's clock is wrong, this is almost never a "false positive" in the technical sense, because the current time is, in fact, outside the certificate's `NotAfter` timestamp. Although this is definitely a security error, it is in practice unlikely that an attacker is using an expired certificate to conduct surveillance or manipulate traffic.

It may be obvious that a security certificate expired, but *why* is it expired? Is the site having technical difficulties? Maybe the operators forgot to renew it, or perhaps the CA refused to issue a new certificate due to validation trouble. It is hard to know for certain.

In this way, current browser warnings are not so different from risk indicators. A risk indicator may appear because a domain name has a lot of labels. But *why* does it have so many labels: is it a development server? A vanity subdomain? Or a deceptive site trying to hide its true identity? There are both benign and malicious possibilities.

Without more infrastructure, it seems difficult to know for sure how a technical characteristic relates to real threats, and so the warning takes on more of an advisory role in conveying when an interaction happens in a context where there is good reason to have less trust in the site. Hence, our approach uses a less opinionated design.

6.2 Unopinionated Design

Our warning style encourages users to learn more and conduct their own investigation. From an economic perspective, manually inspecting all the factors considered by our framework on every site is untenable and would lead to more lost time than simply becoming a victim and remedying the fallout [16]. However, the risk assessment framework automates the majority of this labor and invites user intervention only when necessary.

Carefully-worded, educational warnings and the context used by the risk assessment framework are crucial for more effective protections. As warning fatigue has decreased, "improving adherence rates may require addressing numerous smaller, more contextual issues. For example, users seem to consider decision factors like importance of their primary task and whether alternative sites with similar content are available; warnings could perhaps tip some decisions toward adherence by nudging users away from trivial tasks or pointing them toward alternative sites" [38]. In accordance with that, the rules used by this framework and the warnings produced from them may be able to help convince users when they are at risk, or at least advise them to adjust their current trust level.

Wording is one aspect of opinionated design, but choices offered to the user are another. A recent study by Mozilla Firefox UX researchers recommends “employing opinionated design, to an appropriate degree” [51]. They “encouraged users to make the safer choice by giving it a design advantage as the ‘clear default choice’” [9, 51]. We think there is a way to balance unopinionated wording with opinionated options.

6.3 Identity Mapping

One of the most common requests from the user study was to improve warning messages by being more specific about whether the site was really the site they thought they were on. For example, respondents wanted messages like, *“This is not one of Google’s authorized URLs”* when logging into a fake Gmail page. We like this idea because most users cannot interpret URLs effectively, and there is even contemporary work experimenting with removing them from the UI entirely [45].

However, creating and maintaining an accurate, authoritative mapping between legal entities and DNS properties is not trivial. This is similar to what Extended and Organizational Validation (EV/OV) certificates try to do, but their utility has been questioned in recent years [19]. This is an area for future research.

6.4 Customizability

Several participants requested the ability to customize warning behavior. This comprised two specific themes: making the automatic pop-out adjustable (some wanted this based on site or severity), and the option to configure site settings before allowing a risky interaction. We think both would be useful. The first could be vital for reducing warning fatigue or annoyance. The latter gives users more options than just leave or continue, which are the only two options presented even by most of today’s browser warnings. Respondents would like the ability to *“toggle settings for that website before you continue with it,”* which implies blocking the risky interaction, notifying the user, and allowing them to customize what the site is allowed to do (set cookies, download files, use a lot of CPU or memory, etc.) before proceeding.

6.5 Preconceived Trust

We noticed that several participants wrote about their experience with either the current security indicator or the new risk indicator as if they had already made a trust decision before consulting the indicator. This is concerning for two reasons.

1. Users might only consult security indicators if they notice something unusual about the page themselves first. As one respondent said, *"If I notice something suspicious on the website, I can check the security indicator to see if it's really the site I was intending to visit."* However, not all deceptive sites are visibly altered, so the user is unlikely to notice anything suspicious. Another said, *"When it warned me about sites that I visit frequently, I stopped paying attention to it,"* even though in our study the risk indicator never appeared on a site they have visited at all, let alone frequently. This tendency prevailed with other participants who wrote, *"I wouldn't want to have to deal with [the risk indicator] every time I log into a website that looks similar to one I've used before,"* and *"I thought that the site looked just like Gmail, so I kept using it,"* apparently assuming that malicious sites cannot look like legitimate ones.

2. When users don't notice anything wrong on a deceptive page, they will instinctively trust how they accessed the site, and subsequently trust further interactions, despite warnings from the risk indicator. This is exhibited by responses like, *"If I knew that I was on a safe site, then I could continue on that site without having to [dismiss the warning] a bunch of times,"* and *"I usually assume I know better than the alert system."*

Multiple participants across a breadth of technical skills shared this kind of thinking, rendering indicators of both kinds less useful to them since their mind was already made up. Our data confirm Reeder et al.'s recent findings, that "many respondents cited their trust in a site as a reason to proceed despite an SSL warning" [38].

6.6 Developer Considerations

It would be helpful for website developers to gain insights into how their sites are viewed by the risk assessment framework. Most browsers have a developer tools section, and we think an area dedicated to the assessment of technical properties of the site, as well as information related to the history of the display of the risk indicator on that site, could be used to help developers improve their sites. A reporting URI, like that which is implemented for Content-Security-Policy, could be specified by site owners so that clients can report when the risk indicator is shown on their site. Since attackers could also specify a reporting URI, browsers would need to be careful not to send any data in a report that is could reveal sensitive patterns such as the user's browsing habits. Even if attackers abuse this feature to successfully tune their misbehavior to avoid risky assessments, it forces them into tighter and tighter technical constraints within which to operate. If done correctly, these tools would make it easier for developers to improve their sites, cut down on false positives, and help more sites to adhere to best practices.

6.7 Browser Adoption

There are thousands of research publications dedicated to detecting phishing and many other kinds of deception on the Web, and we were only able to reference a small sample of them in this paper. The vast majority of them are quantifiably successful, in that they report results significantly better than the baseline. One question we asked ourselves as we sifted through the many contributions was, "With so much successful work, who is bringing these protections to the users of the Internet?"

We are keenly aware that many of the published methods are available in some form: source code, browser extensions, antivirus and security software, packaged as part of a larger distribution, or integrated behind-the-scenes into existing services. But it seems that few—if any—mainstream web browsers have adopted one or more of the published methods. Web browsers have made progressive advancements in their protections over the years, but with so many methods in published literature, why does it appear that so few get used?

If future work in this area is to be realized, we think this is worth some consideration. Although we do not have the answer, we suggest several theories.

- Mainstream browsers tend to be conservative in deploying new features and protections so as to avoid disrupting massive user bases, along with all the side-effects which come from that (including financial, technical, and legal aberrations).
- Patents might make some technologies unavailable by browser vendors.
- There are real logistical issues when shipping client-side software. Threats evolve quickly, and techniques proposed a few years ago might be obsolete already. Most major browsers ship stable-channel updates every few weeks or months. Attackers ship new sites every few minutes. This could put browsers at a disadvantage if their client-side protections are updated on the same release cycle.
- Bandwidth and privacy constraints prevent a browser from calling out to remote servers on every page load. (This is already implemented in some cases to enforce OCSP, but it puts heavy load on OCSP responders and is easily blocked.[1]) Thus, a feasible solution should be able to operate entirely client-side.
- Browser vendors may also have their sights set higher than just squeezing out more precision and recall from detection algorithms; instead of being very conservative, they may be very progressive by spending time implementing other new technologies. As one Firefox developer recently articulated, "Web Authentication (WebAuthn) is our best technical response to phishing, which is why we've championed it as a technology."[2].

[1] "High-reliability OCSP stapling and why it matters." Cloudflare. https://blog.cloudflare.com/high-reliability-ocsp-stapling/

[2] https://groups.google.com/forum/#!topic/mozilla.dev.platform/q5cj38hGTEA

We hope that browser vendors will be more motivated to further our research and adopt this work because our method is easy to tune, generalizes better across a wider variety of old and emerging threats (because we do not attempt to classify individual threats), and really gets at the crux of what we think warnings should be about: helping the user to make informed trust decisions.[3]

6.8 Limitations

Studies have shown that what people say they will do and what they will actually do in practice is different [36, 37]. This is called self-reporting response bias, but the bias is often systematic [36]. It can be difficult to draw conclusions from minor variations in our surveys because we only made minor textual and graphical changes. Since the purpose of our survey on risk indicator design (Appendix A.2) was mainly to inform our eventual user study, this bias was tolerable in our case.

The demographic of our user study was limited to local university students, so our results might not generalize to wider populations. However, many participants indicated they were not particularly tech-savvy, which is one trait that is likely more representative of a general population.

The study was also performed in a lab setting on our computer with a custom web browser, and with clear instructions to look for the indicators, but to approach the tasks as if they were on their own computer. We acknowledge that this is subject to the Hawthorne effect, and hope that future work in partnership with mainstream web browsers (especially a longitudinal study) will help provide greater insights into risk indicators.

A longitudinal study would also help answer questions about the effectiveness of this approach in the field because there is no universal ground truth we can use to evaluate risk assessment as described by our method. Because risk is assessed on a per-user, per-interaction

[3] We suggest that if browser warnings really were about blocking or disabling threats absolutely, there would not be ways to bypass or override them. Because bypassing warnings is possible and sometimes appropriate, then it follows that warnings must actually be about helping users make their own trust decisions.

basis, any definitive list of risky interactions would have to generate hypothetical scenarios and browser histories, then augment those with interactions, and experts would need to apply manual judgment to classify each data point. We began working on this but altered our evaluation to use surveys and a user study because not only was producing a generated data set tedious, but we determined that it has little transferable real-world value.

Although we think our user study was well-designed in that it asked participants to use their real credentials and payment information when completing the tasks, many participants noted that they continued to complete tasks because they knew they were in a lab study and they believed we wouldn't expose them to any actual threats. Despite our request that they approach these tasks as if on their own computer, this sense of security may have slanted their behavior in the study to be more cavalier.

Chapter 7

Conclusion

We introduced the idea of a risk indicator which is worth exploring as an alternative to security indicators. The risk assessment framework we outlined describes a way to carefully craft rules to control the appearance of the risk indicator and the associated warning text. Our user study evaluated the effectiveness of the risk indicator in a controlled environment and collected feedback about this approach to security. Our results show that when indicators are designed to use context beyond the security of the connection, they can help protect users in a Web that is fully transitioned to HTTPS.

Our work revealed several areas where additional work is needed:

- *Assessing risks*: Our work posits there is a gap between the threats that can be detected with near 100% certainty by current anti-phishing tools and purely benign pages. This gap includes sites that are more broadly deceptive than phishing and malware attacks, as well as phishing or malware attacks that have not yet been identified by existing tools or those which do not target a global audience (such as spear-phishing). More work can be done to better characterize this space of potential threats and to measure the use of deceptive techniques broadly.

- *Assessing risk communication*: Our work can be placed in a broader set of works on risk communication as related to computer security. More work is needed to study how well the warnings we developed communicate risk to end users and whether the warnings help users make choices (either acting on or ignoring the risk) that align with their understanding and context. This could best be accomplished with studies of the risk indicator in real-world scenarios, over longer periods of time, and with a more diverse population.

- *Risk indicator design*: Our work represents an initial design of the risk indicator. Our results narrowed down the choice for an icon and accompanying label to just a few among the choices we studied. Work could be done studying additional designs with larger samples. In particular, using an informational icon such as ⓘ may be more in keeping with current practices for browser indicator design. Likewise, there are a variety of practical questions to answer, such as handling the case when multiple rules are triggered for the same interaction.

- *Improving utility for users*: Users would like to see a variety if improvements in assessing threats. For example, they would like to see a better mapping from URLs to website ownership, customizable warning behavior, and a clear set of instructions for investigating their risk further. More work could be done in these areas in addition to identifying further areas for improvement.

Our contribution explored the approach of indicating risk and paved a foundation for more research. We think further work in this direction and continued cooperation within the Internet security and web browser communities will lead to more aware and informed users, and fewer successful deception attacks on the Web.

References

[1] Saeed Abu-Nimeh, Dario Nappa, Xinlei Wang, and Suku Nair. A comparison of machine learning techniques for phishing detection. In *Proceedings of the Anti-phishing Working Groups 2Nd Annual eCrime Researchers Summit*, eCrime '07, pages 60–69, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-939-5. doi: 10.1145/1299015.1299021. URL http://doi.acm.org/10.1145/1299015.1299021.

[2] Devdatta Akhawe and Adrienne Porter Felt. Alice in Warningland: A Large-Scale Field Study of Browser Security Warning Effectiveness. In *USENIX Security Symposium*, volume 13, 2013.

[3] R. Barnes, Cisco, J. Hoffman-Andrews, EFF, D. McCarney, Let's Encrypt, J. Kasten, and University of Michigan. Automatic Certificate Management Environment (ACME). Technical report, IETF, 2018. URL https://datatracker.ietf.org/doc/draft-ietf-acme-acme/.

[4] Yafit Becher and Emily Schechter. New Security Protections, Tailored to You. The Keyword, October 2017. URL https://www.blog.google/technology/safety-security/new-security-protections-tailored-you/.

[5] Ye Cao, Weili Han, and Yueran Le. Anti-phishing based on automated individual white-list. In *Proceedings of the 4th ACM Workshop on Digital Identity Management*, pages 51–60. ACM, 2008.

[6] Ammar Yahya Daeef, R Badlishah Ahmad, Yasmin Yacob, and Ng Yen Phing. Wide Scope and Fast Websites Phishing Detection Using URLs Lexical Features. In *2016 3rd International Conference on Electronic Design (ICED)*, pages 410–415. IEEE, 2016.

[7] Piotr Duszyski. Phishing NG. Bypassing 2FA with Modlishka. https://blog.duszynski.eu/phishing-ng-bypassing-2fa-with-modlishka/, January 2019.

[8] Serge Egelman and Stuart Schechter. The importance of being earnest [in security warnings]. In *International Conference on Financial Cryptography and Data Security*, pages 52–59. Springer, 2013.

[9] Adrienne Porter Felt, Robert W. Reeder, Hazim Almuhimedi, and Sunny Consolvo. Experimenting at scale with google chrome's ssl warning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2667–2670. ACM, 2014.

[10] 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 *Proceedings of the Conference on Human Factors and Computing Systems*, 2015.

[11] Adrienne Porter Felt, Robert W. Reeder, Alex Ainslie, Helen Harris, Max Walker, Christopher Thompson, Mustafa Embre Acer, Elisabeth Morant, and Sunny Consolvo. Rethinking Connection Security Indicators. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, pages 1–14, 2016.

[12] Ian Fette, Norman Sadeh, and Anthony Tomasic. Learning to Detect Phishing Emails. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 649–656, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-654-7. doi: 10.1145/1242572.1242660. URL http://doi.acm.org/10.1145/1242572.1242660.

[13] B. J. Fogg, Jonathan Marshall, Othman Laraki, Alex Osipovich, Chris Varma, Nicholas Fang, Jyoti Paul, Akshay Rangnekar, John Shon, Preeti Swani, and Marissa Treinen. What Makes Web Sites Credible?: A Report on a Large Quantitative Study. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '01, pages 61–68, New York, NY, USA, 2001. ACM. ISBN 1-58113-327-8. doi: 10.1145/365024.365037. URL http://doi.acm.org/10.1145/365024.365037.

[14] Amber Gott. Staying Safe from Phishing Attacks. https://blog.lastpass.com/2016/01/staying-safe-from-phishing-attacks.html/, January 2016.

[15] Olaf Hartig. Trustworthiness of data on the web. In *Proceedings of the STI Berlin & CSW PhD Workshop*, 2008.

[16] Cormac Herley. So long, and no thanks for the externalities: the rational rejection of security advice by users. In *Proceedings of the 2009 New Security Paradigms Workshop*, pages 133–144. ACM, 2009.

[17] Tobias Holgers, David E Watson, and Steven D Gribble. Cutting through the Confusion: A Measurement Study of Homograph Attacks. In *USENIX Annual Technical Conference, General Track*, pages 261–266, 2006.

[18] Hsiu-Fang Hsieh and Sarah E Shannon. Three approaches to qualitative content analysis. *Qualitative Health Research*, 15(9):1277–1288, 2005. Sage publications Sage CA: Thousand Oaks, CA.

[19] Troy Hunt. Extended Validation Certificates are Dead. `https://www.troyhunt.com/extended-validation-certificates-are-dead/`, September 2018.

[20] Yogesh Joshi, Samir Saklikar, Debabrata Das, and Subir Saha. Phishguard: a browser plug-in for protection from phishing. In *Internet Multimedia Services Architecture and Applications, 2008. IMSAA 2008. 2nd International Conference on*, pages 1–6. IEEE, 2008.

[21] Mahmoud Khonji, Youssef Iraqi, and Andrew Jones. Phishing detection: a literature survey. *IEEE Communications Surveys & Tutorials*, 15(4):2091–2121, 2013. IEEE.

[22] Panagiotis Kintis, Najmeh Miramirkhani, Charles Lever, Yizheng Chen, Rosa Romero-Gómez, Nikolaos Pitropakis, Nick Nikiforakis, and Manos Antonakakis. Hiding in Plain Sight: A Longitudinal Study of Combosquatting Abuse. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 569–586. ACM, 2017.

[23] Anh Le, Athina Markopoulou, and Michalis Faloutsos. PhishDef: URL Names Say it All. In *2011 Proceedings IEEE INFOCOM*, pages 191–195. IEEE, 2011.

[24] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. Tranco: A research-oriented top sites ranking hardened against manipulation. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium*, NDSS 2019, February 2019. doi: 10.14722/ndss.2019.23386.

[25] Zachary C Lipton. The doctor just won't accept that! *arXiv preprint arXiv:1711.08037*, 2017.

[26] Nathan Malkin, Arunesh Mathur, Marian Harbach, and Serge Egelman. Personalized security messaging: Nudges for compliance with browser warnings. In *2nd European Workshop on Usable Security. Internet Society*, 2017.

[27] Samuel Marchal, Giovanni Armano, Tommi Gröndahl, Kalle Saari, Nidhi Singh, and N Asokan. Off-the-hook: An efficient and usable client-side phishing prevention application. *IEEE Transactions on Computers*, 66(10):1717–1733, 2017. IEEE.

[28] D Kevin McGrath and Minaxi Gupta. Behind phishing: An examination of phisher modi operandi. *LEET*, 8:4, 2008.

[29] Tyler Moore and Benjamin Edelman. Measuring the perpetrators and funders of typosquatting. In *International Conference on Financial Cryptography and Data Security*, pages 175–191. Springer, 2010.

[30] Lily Hay Newman. Phishing Schemes are Using Encrypted Sites to Seem Legit. `https://www.wired.com/story/phishing-schemes-use-encrypted-sites-to-seem-legit/`, December 2017.

[31] Nick Nikiforakis, Marco Balduzzi, Lieven Desmet, Frank Piessens, and Wouter Joosen. Soundsquatting: Uncovering the use of homophones in domain squatting. In *International Conference on Information Security*, pages 291–308. Springer, 2014.

[32] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 309–319. Association for Computational Linguistics, 2011.

[33] Bryan Parno, Cynthia Kuo, and Adrian Perrig. Phoolproof phishing prevention. In *International Conference on Financial Cryptography and Data Security*, pages 1–19. Springer, 2006.

[34] Pawan Prakash, Manish Kumar, Ramana Rao Kompella, and Minaxi Gupta. Phishnet: Predictive blacklisting to detect phishing attacks. In *2010 Proceedings IEEE INFOCOM*, pages 1–5. IEEE, 2010.

[35] Swapan Purkait. Phishing counter measures and their effectiveness–literature review. *Information Management & Computer Security*, 20(5):382–420, 2012. Emerald Group Publishing Limited.

[36] Elissa M Redmiles, Ziyun Zhu, Sean Kross, Dhruv Kuchhal, Tudor Dumitras, and Michelle L Mazurek. Asking for a Friend: Evaluating Response Biases in Security User Studies. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1238–1255. ACM, ACM, 2018.

[37] Elissa M Redmiles, Sean Kross, and Michelle L Mazurek. How Well Do My Results Generalize? Comparing Security and Privacy Survey Results from MTurk, Web, and Telephone Samples. In *How Well Do My Results Generalize? Comparing Security and*

Privacy Survey Results from MTurk, Web, and Telephone Samples, page 0. IEEE, IEEE, 2019.

[38] 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 *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 512:1–512:13, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5620-6. doi: 10.1145/3173574.3174086. URL `http://doi.acm.org/10.1145/3173574.3174086`.

[39] Scott Ruoti, Tyler Monson, Justin Wu, Daniel Zappala, and Kent Seamons. Weighing Context and Trade-offs: How Suburban Adults Selected Their Online Security Posture. In *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*, pages 211–228. USENIX Association, 2017. ISBN 978-1-931971-39-3. URL `https://www.usenix.org/conference/soups2017/technical-sessions/presentation/ruoti`.

[40] Doyen Sahoo, Chenghao Liu, and Steven CH Hoi. Malicious URL Detection Using Machine Learning: a Survey. *arXiv preprint arXiv:1701.07179*, 2017.

[41] Emily Schechter. Evolving Chrome's security indicators. `https://blog.chromium.org/2018/05/evolving-chromes-security-indicators.html`, May 2018.

[42] Stuart E Schechter, Rachna Dhamija, Andy Ozment, and Ian Fischer. The Emperor's New Security Indicators. In *2007 IEEE Symposium on Security and Privacy (SP'07)*, pages 51–65. IEEE, 2007.

[43] Hossain Shahriar and Mohammad Zulkernine. Trustworthiness testing of phishing websites: A behavior model-based approach. *Future Generation Computer Systems*, 28 (8):1258–1271, 2012. Elsevier.

[44] Peggy Fletcher Stack. Creator of fake LDS apology does his own apologizing, acknowledges causing tremendous pain for black Mormons, 5 2018. URL `https://www.sltrib.com/religion/2018/05/23/creator-of-fake-lds-apology-does-his-own-apologizing-acknowledges-causing-tremendous-pain-for-black-mormons/`.

[45] Emily Stark. The URLephant in the Room. USENIX Enigma, January 2019. URL `https://www.usenix.org/conference/enigma2019/presentation/stark`. USENIX Enigma.

[46] Janos Szurdi, Balazs Kocso, Gabor Cseh, Jonathan Spring, Mark Felegyhazi, and Chris Kanich. The Long "Taile" of Typosquatting Domain Names. In *USENIX Security Symposium*, pages 191–206, 2014.

[47] Gaurav Varshney, Manoj Misra, and Pradeep K Atrey. A phish detector using lightweight search features. *Computers & Security*, 62:213–228, 2016. Elsevier.

[48] Robert A Virzi. Refining the test phase of usability evaluation: How many subjects is enough? *Human factors*, 34(4):457–468, 1992. SAGE Publications Sage CA: Los Angeles, CA.

[49] Arun Vishwanath. Habitual Facebook use and its impact on getting deceived on social media. *Journal of Computer-Mediated Communication*, 20(1):83–98, 2014. Oxford University Press Oxford, UK.

[50] Elliot Volkman. 49 Percent of Phishing Sites Now Use HTTPS. `https://info.phishlabs.com/blog/49-percent-of-phishing-sites-now-use-https`, December 2018.

[51] Meridel Walkington. Designing Better Security Warnings. Mozilla Blog, March 2019. URL `https://blog.mozilla.org/ux/2019/03/designing-better-security-warnings/`.

[52] Colin Whittaker, Brian Ryner, and Marria Nazif. Large-Scale Automatic Classification of Phishing Pages. In *Proc. Network and Distributed System Security Symposium (NDSS)*, volume 10, 2010.

[53] Benjamin Wood. No, the Mormon church did not apologize for having a history of racism; hoaxer says he meant fake message to spark discussion. Salt Lake Tribune, May 2018. URL `https://www.sltrib.com/news/2018/05/17/no-the-mormon-church-did-not-apologize-for-having-a-history-of-racism/`.

[54] Weining Yang, Aiping Xiong, Jing Chen, Robert W Proctor, and Ninghui Li. Use of Phishing Training to Improve Security Warning Compliance: Evidence from a Field Experiment. In *Proceedings of the Hot Topics in Science of Security: Symposium and Bootcamp*, pages 52–61. ACM, 2017.

[55] Ningxia Zhang and Yongqing Yuan. Phishing detection using neural network. *Department of Computer Science, Department of Statistics, Stanford University, CA, available at: http://cs229. stanford. edu/proj2012/ZhangYuan-PhishingDetectionUsingNeuralNetwork. pdf (accessed April 23, 2016).[Google Scholar]*, 2013.

Appendix A

Mechanical Turk Survey

A.1 Task #1

Suppose you were using your web browser normally and were shown this warning: *(sample from one of the following)*

i. Connection is not private

   This page was loaded insecurely, making it risky to rely on what it says or does. Information you enter on this page may not stay private, and the content of this page could be malicious or links on this page could take you to malicious sites. Further, other people or systems may know that you are visiting this specific page, along with the full contents of the page.

ii. Connection is not fully private

   Some of this page was loaded insecurely, making it risky to rely on what it says or does. Information you enter on this page may not stay private, and some content of this page could be malicious or links on this page could take you to malicious sites.

iii. Site address is suspicious

   The address of this page (**paypal.com.secure-log.in**) is similar to another site you have visited before (**paypal.com**). If this page looks similar to **paypal.com**, it may be trying to deceive you.

iv. Excessive changing of sites

   This page changed to different sites multiple times when you visited it, suggesting that the link you clicked may possibly be malicious.

   The other pages visited were:

(a) http://foo-bar.ad-services.com/track?id=asdf1234

(b) http://download-mania.com/download/foo123

(c) https://login-paypal.com/webscr.cgi?partner=foobar

v. Site address may be a typo

You have never visited this site (**gooogle.com**) before, but you have visited one with a very similar name (**google.com**). It's possible a mistake was made typing the address, and as a result, you could be on the wrong site.

You should double-check that you intend to be on **gooogle.com**.

vi. Site name is unusually long

This site has a lot of parts to its domain name, which can be confusing and may be an attempt to mislead visitors. Ensure that you intend to be on **installupgradenow.fastandgoodcontentjust 4youtodownloadthisweek.stream**.

vii. First visit to this site

This is the first time you've visited this site, and you reached it by following a link that is difficult to trust.

If this site looks like a site you've visited before on this computer, this one might be trying to deceive you. Be careful about entering information into this page.

Because this is your first time on this site, this warning will not appear next time.

viii. Reading this site may be risky

You have never read content on this site before. Make sure you intend to be on **ny-times.com** and not **nytimes.com** which you have visited in the past.

If this site looks like **nytimes.com** to you, then the site you're currently viewing might be trying to deceive you.

ix. Site name does not match its language

The content of this page is in English, but its address contains non-English characters. Although not always the case, sometimes malicious sites use non-English characters to appear like well-known sites in attempts to deceive visitors. Be aware if this site asks for anything suspicious or sensitive.

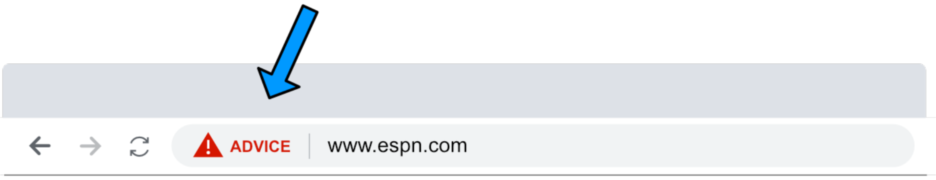
x. Site name has excessive hyphens

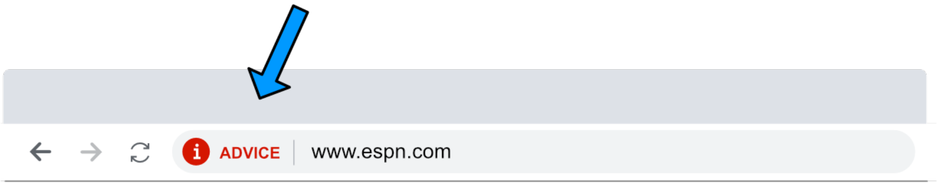
This site has a lot of hyphens in its domain name, which is unusual for sites that collect information. Be aware that you are on a site belonging to **webeatech.com**, which may or may not be what you intended.

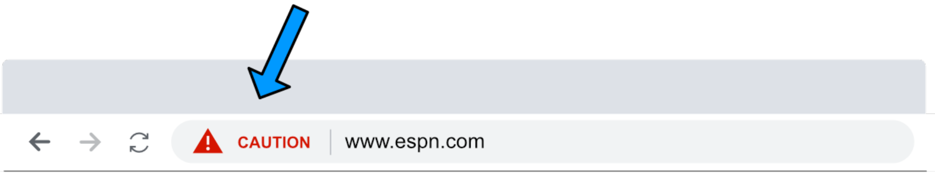
1. Would you want to be notified when you are (unknowingly) in a situation described by this warning?
   - Yes
   - No
2. Would you find warnings like this helpful to you while using the Internet? (Very Unhelpful—Very Helpful)
3. Which of the following would you be most likely to do in response to this warning?
   (a) Investigate further
   (b) Leave the site and not come back
   (c) Leave the site then come back to it
   (d) Avoid entering information into the site
   (e) Ignore the warning and continue using the site normally

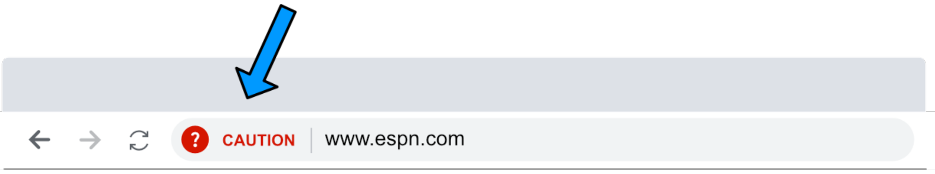
A.2 Task #2

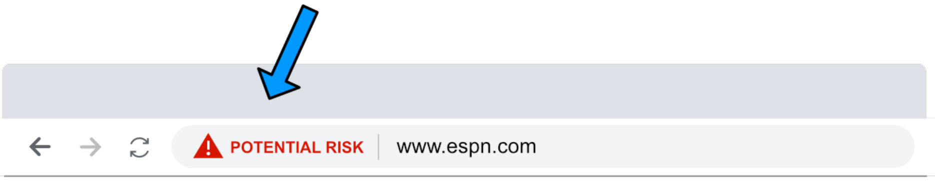
Look at this image of the top part of a web browser: *(sample from one of the following)*

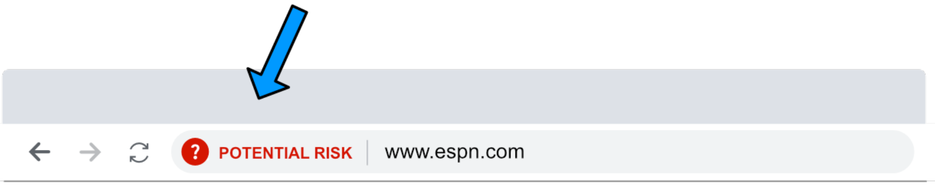


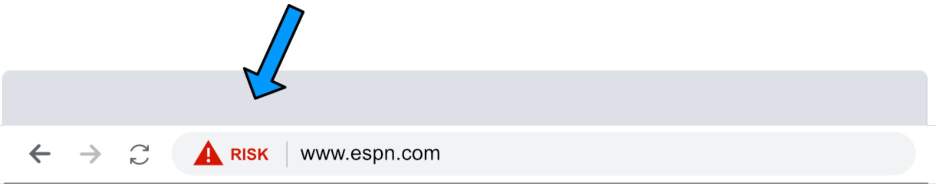


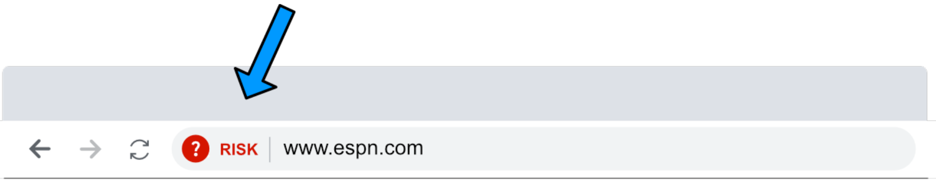


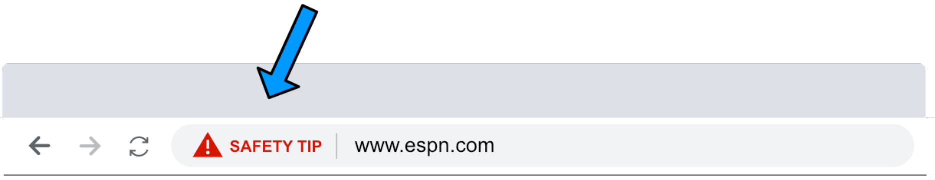


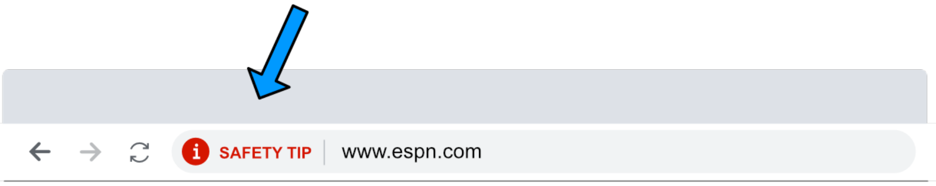












1. What do you think the indicator (pointed to) means?

2. Suppose you visited a website and this indicator appeared. How safe would you feel about the current website? (Very Unsafe—Very Safe)

3. Which of the following would you be most likely to do after seeing the indicator?

    (a) Click it / get more information

    (b) Leave the site and not come back

    (c) Leave the site then come back to it

    (d) Avoid entering information into the site

    (e) Ignore it and continue using the site normally

4. What do you think the source of the indicator is?

    (a) The website

    (b) The web browser

    (c) An ad

    (d) Virus or malware

    (e) A hacker

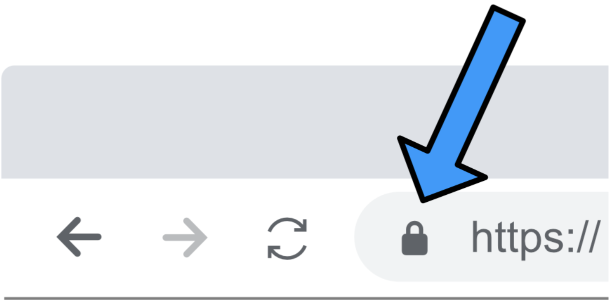
Appendix B

User Study Questionnaire

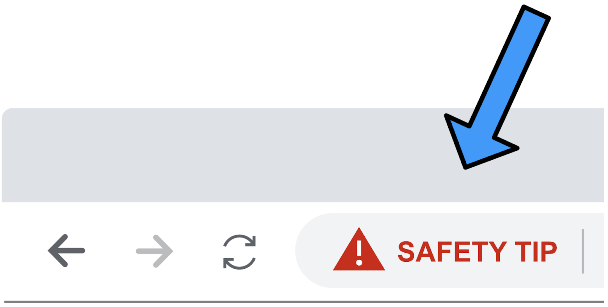
1. On a scale of 1 to 5, how would you rate your skills and understanding of computers and technology?

   1/Simple = No formal technical training; 3/Intermediate = Specific technical training for certain tasks; 5/Advanced = Received education in a tech/computer-related field or had years of industry experience

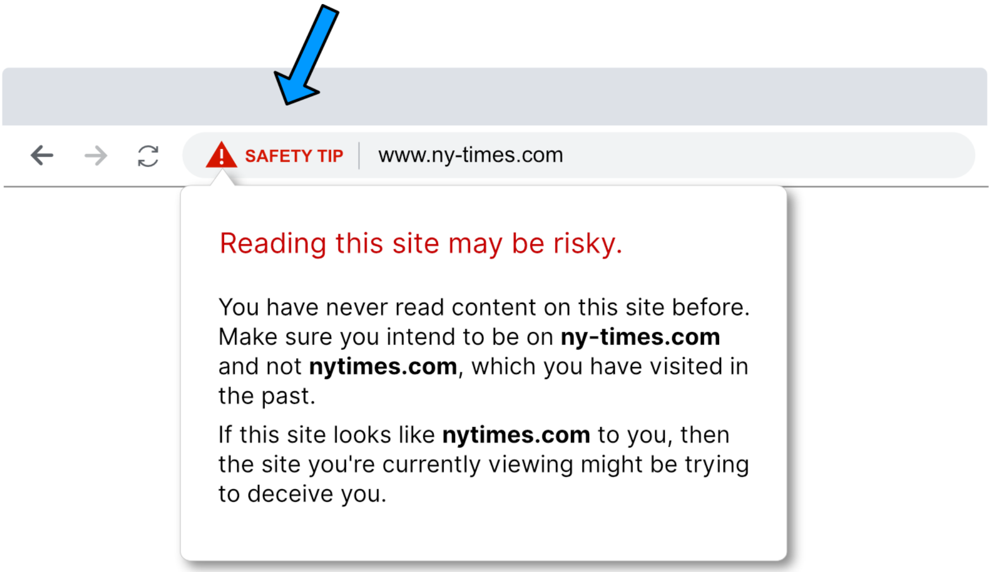
2. In this study, you experienced three ways for web browsers to warn you of potential threats. One of the ways is what browsers currently show, called a ”security indicator” which signifies a secure connection to the site:

   Another way was our new ”risk indicator” design, which signifies a potential threat:

   Another way was our new risk indicator which automatically pops out the details of the warning:



Reading this site may be risky.

You have never read content on this site before. Make sure you intend to be on **ny-times.com** and not **nytimes.com**, which you have visited in the past.

If this site looks like **nytimes.com** to you, then the site you're currently viewing might be trying to deceive you.

Which of these 3 ways do you prefer the most?

- Current browsers' security indicator
- Risk indicator
- Risk indicator with automatic pop-out

3. Why do you prefer that one the most?

4. If you completed any tasks even when you saw one of the new risk indicators that we have designed, it was mostly because:

   - I knew I was in a lab study so it must have been safe
   - I thought that's what I was supposed to do to finish the study
   - I did not agree that I would be at risk in real life
   - I did not see the risk indicator
   - I did not understand the risk
   - (I did not complete any task where I saw the risk indicator)
   - Other:

5. If you encountered our new risk indicator warnings in your daily web browsing like those you experienced today, how helpful do you think they would be? (Consider that these warnings could appear with interactions that are not risky, because computers may not be perfect.) (Unhelpful—Helpful)

6. When you saw our new risk indicator, did you understand that it signified potential risks that need investigating?

    - Yes
    - No
    - I did not see any risk indicators

7. Have you ever encountered a browser security warning when visiting a web page before today's study?

    - Yes
    - No
    - Not sure

8. If yes, could you describe what happened, and whether you sought more information about how to respond to the warning?

9. One variation of the new risk indicators you experienced involved the warning text dropping over part of the page automatically when a potential risk was detected. How would you describe your reaction when that happened? (check all that apply)

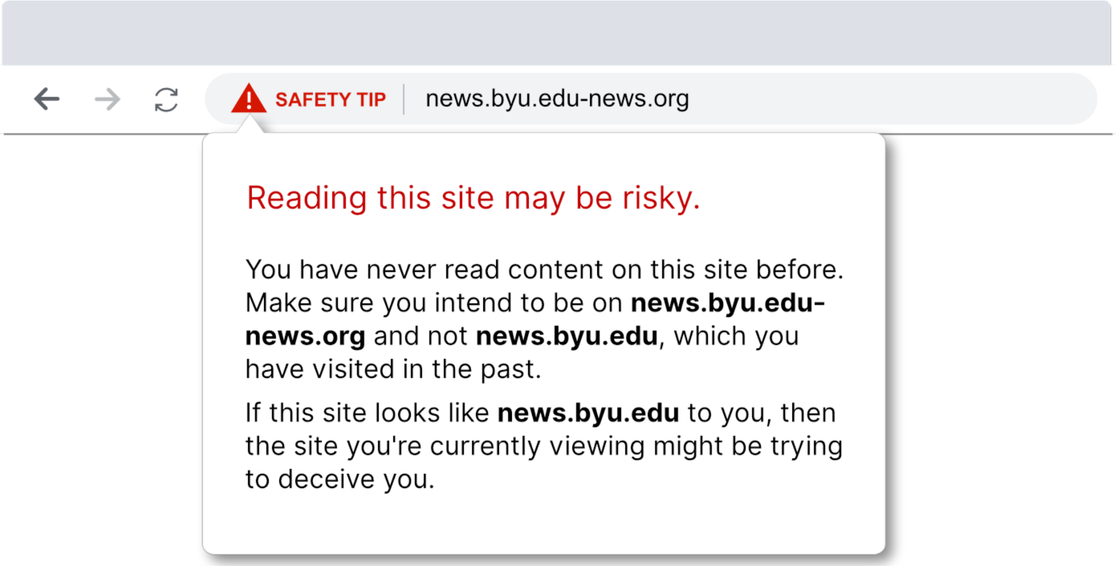
    - I was surprised
    - I was startled
    - I read the warning text
    - I dismissed it without reading it
    - I ignored it
    - I was annoyed
    - I felt protected
    - I felt more vulnerable
    - I didn't notice any drop-down warnings
    - Other:

10. What did you think was the source of the new risk indicators/warnings?

    - The website
    - The web browser

- A hacker
- A virus or malware
- An advertisement
- Other:

11. Task #2 asked you to read a BYU News article. The news site you visited was actually not genuine and the article was fake. A risk indicator appeared in an attempt to warn you of this. The questions in this section are about this warning:



Reading this site may be risky.

You have never read content on this site before. Make sure you intend to be on **news.byu.edu-news.org** and not **news.byu.edu**, which you have visited in the past.

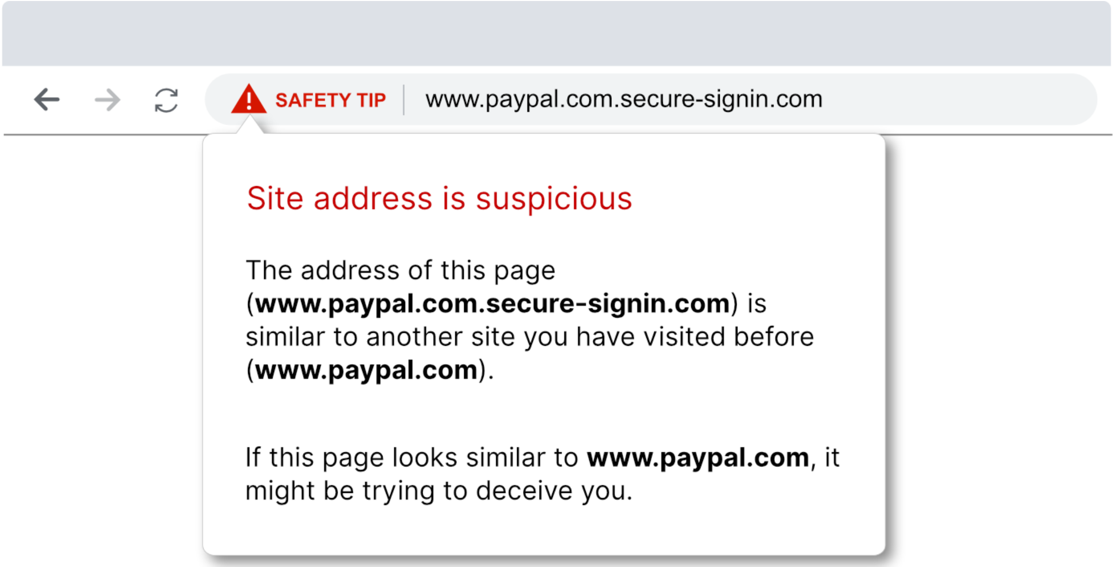
If this site looks like **news.byu.edu** to you, then the site you're currently viewing might be trying to deceive you.

If you encountered this warning in real life, how helpful would you find it? (Unhelpful—Helpful)

Which of the following would you be most likely to do in response to this warning? (check all that apply)

- Investigate further
- Leave the site and not come back
- Leave the site and then come back to it
- Avoid entering information into the site
- Ignore the warning and continue using the site normally

12. Task #4 asked you to log in to PayPal, Google/Gmail, or create an account at Mint. A risk indicator appeared because the link you clicked was manipulated to take you to a deceptive site that could have tried to steal your information. The questions in this section are about this warning:



www.paypal.com.secure-signin.com

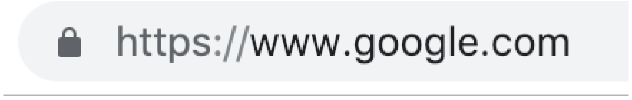
Site address is suspicious

The address of this page (**www.paypal.com.secure-signin.com**) is similar to another site you have visited before (**www.paypal.com**).

If this page looks similar to **www.paypal.com**, it might be trying to deceive you.

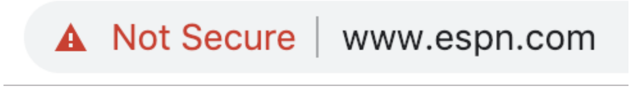
(Repeat the previous two questions)

13. Current safety indicators in browsers tell you the page is safe when it is encrypted:



https://www.google.com

A site that is not encrypted currently looks like this:



Not Secure | www.espn.com

In the future, every site will be encrypted, even deceptive or malicious ones. We're trying to find new browser indicators that will warn people of potential threats regardless of encryption. No system can be perfect, however, so some safe sites may have a warning of risk, and some sites that are not safe might not have a warning.

What are we doing right, in your opinion?

What are we doing wrong, in your opinion?

14. Please describe situations when you would you be most likely to ignore the warning and not click on it:

Examples: shopping, banking, social media, reading the news, leaving comments, etc.

15. Please describe situations when you would you be most likely to heed the warning by clicking on it and getting more information:

(same examples)

16. “I am confident that I would be able to judge whether I am at risk without the new risk indicators.” (Strongly Disagree—Strongly Agree)

17. “I am confident that if I chose to investigate a potential risk from the new risk indicator, I would be able to judge whether I am at risk.” (Strongly Disagree—Strongly Agree)

Appendix C

Study Coordinator Guide

1. Before the participants arrive:

   (a) Unlock the study computer.

   (b) If the custom web browser is not already open, open it. Otherwise, click Lab Study and then choose New Subject.

   (c) Place the participant instructions next to the keyboard.

2. Make sure the participants are qualified.

   (a) 18 or older

   (b) Have not previously participated in a user study about web browsers

   (c) Have not been informed as to the details of this study by someone who has taken it

3. When the participants arrive, read them the following:

   Welcome to our study. I am a study coordinator, and am here to assist you as needed.

4. Have participant sign the consent form.

   Please read the consent form we are giving you. The main points of this form are that:

   (a) *We are going to ask you to use a web browser on our computer to perform specific tasks that are common when using the Internet.*

   (b) *Please approach the tasks as if you were at your own computer, because you will be using your own account and payment information. Since the purpose of the study is not for you to complete all the tasks, you are allowed to skip any of the steps that you do not feel comfortable completing.*

(c) *At the end of the study, you'll be given a survey to fill out. Your answers may be published, but with no identifying information.*

(d) *You will receive $15 as compensation for your participation in this study. The expected time commitment is approximately 45 minutes.*

(e) *If you request it, we will delete all data collected from your participation.*

5. Take the participant with whom you will work to the study room. Ask the participant to sit down, and say:

   > *Please read through the instructions before you begin, and ask me if you have any questions about them. Otherwise, you can then start on the tasks.*

6. When the participant is done, load the exit survey on the computer. Direct the participant to begin the survey.

7. When they finish, debrief the participant:

   > *We had you perform various tasks that could have normally posed some risk to you or your personal information. In this study, our custom web browser simulated these risks and showed you warnings when it determined you could have been at risk if this had happened during normal use of the Internet.*
   >
   > *However, you were never actually at any real risk. Furthermore, the BYU News article you were asked to read is not real. It was fabricated for the purpose of this study.*
   >
   > *In reality, web browsers don't show warnings for a number of risks that exist on today's Web. Our research purpose today is to improve browser warnings by exposing potential threats that users are not currently warned about. We hope that because of your participation, we can help make a safer Web a reality.*
   >
   > *Are there any questions or concerns that I can address for you?*

8. Thank the participants for their time.

9. Help them fill out the compensation forms and take them upstairs to receive compensation. If it is at a time when the office is closed, direct them where to go at a later time to receive their compensation.

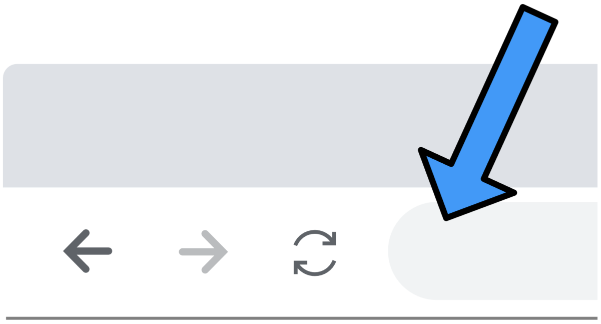
Appendix D

User Study Participant Guide

The computer in front of you is running a very simple web browser, similar to Google Chrome or Mozilla Firefox. You can use the Internet with it much like you're used to doing on your own computer, except there are no tabs.

We would like you to carry out some normal Internet-related activities using our web browser. There are 4 specific tasks we have described below for you to do. We'd ask you to follow the steps as listed, even if you know another way to do the task. However, the purpose is not that you finish every task; you do not need to complete all the steps to a task if you feel uncomfortable.

The purpose of this study is to evaluate new kinds of browser warnings to help users understand threats against them or their information while using the Internet. At times, you may notice indications of potential threats at the top-left of the browser by the site address (URL), in this area:



You will repeat the task list 3 times. Each time, a different variation of the indicator will be used when risks are detected. You may skip any task if you feel uncomfortable finishing it.

Please use our web browser for all tasks (i.e. don't use your own device) and do not exit the browser. However, you may reference your own phone or computer if you need to look up information (such as your password), in order to complete a task.

If you have any questions about the instructions, you may ask the study coordinator for assistance.

D.1 Task #1

a. Go to `google.com`.

b. Do a search for **byu homepage**.

c. Click on the first result for BYU's website.

d. Log in to myBYU.

e. Check on your class schedule, if you have classes.

f. Log out of myBYU.

D.2 Task #2

a. Go to `google.com` again.

b. Search for **byu news**.

c. Click on the first result, which should currently be a story about BYU honor code policy.

d. Read the article.

D.3 Task #3

a. Navigate to `ebay.com`.

b. Find a product that looks interesting to you and click **Buy Now**.

c. Begin going through **checkout as a guest using your credit card**. Fill out the forms completely as if you were going to buy the item. Enter your address and continue, then enter your credit card information and click **Done**, but do not actually complete the purchase (do not click "Confirm and pay").

D.4 Task #4

For this task, complete only the first set of instructions that applies to you.

D.4.1 If you have a PayPal account:

a. Go to `google.com`.

b. Do a search for **paypal login**.

c. Click the first result.

d. Log into your PayPal account and check the Recent Activity.

e. Log out of PayPal.

D.4.2 OR, if you have a Google/Gmail account:

a. Go to `google.com`.

b. Click the blue **Sign In** button.

c. Log into your Google account.

d. Check your Gmail.

e. Log out of your Google account.

D.4.3 OR, if you have neither:

a. Go to `google.com`.

b. Search for **mint banking**.

c. Click the first result.

d. Click the orange **Sign Up Free** button.

e. Fill out the form, but do not submit it.

D.5 Repeat these tasks

At the top of the browser, click “Lab Study” and then click “Next Treatment” — then repeat the tasks.

D.6 Repeat these tasks again

At the top of the browser, click “Lab Study” and then click “Next Treatment” — then repeat the tasks.

By the end of the study, you should have worked through tasks 1-4 three times.

D.7 Completing the Study

Tell the study coordinator you are done. You will be asked to fill out a short survey about your experience and then you will receive your compensation. Thank you for participating today!