Predicting Impending Exposure to Malicious Content from User Behavior

Mahmood Sharif
Carnegie Mellon University
mahmoods@cmu.edu

Jumpei Urakawa
KDDI Research, Inc.
ju-urakawa@kddi-research.jp

Nicolas Christin
Carnegie Mellon University
nicolasc@cmu.edu

Ayumu Kubota
KDDI Research, Inc.
kubota@kddi-research.jp

Akira Yamada
KDDI Research, Inc.
ai-yamada@kddi-research.jp

ABSTRACT

Many computer-security defenses are reactive—they operate only when security incidents take place, or immediately thereafter. Recent efforts have attempted to predict security incidents before they occur, to enable defenders to proactively protect their devices and networks. These efforts have primarily focused on long-term predictions. We propose a system that enables proactive defenses at the level of a single browsing session. By observing user behavior, it can predict whether they will be exposed to malicious content on the web seconds before the moment of exposure, thus opening a window of opportunity for proactive defenses. We evaluate our system using three months' worth of HTTP traffic generated by 20,645 users of a large cellular provider in 2017 and show that it can be helpful, even when only very low false positive rates are acceptable, and despite the difficulty of making "on-the-fly" predictions. We also engage directly with the users through surveys asking them demographic and security-related questions, to evaluate the utility of self-reported data for predicting exposure to malicious content. We find that self-reported data can help forecast exposure risk over long periods of time. However, even on the long-term, self-reported data is not as crucial as behavioral measurements to accurately predict exposure.

CCS CONCEPTS

• **Security and privacy** → *Network security*; *Human and societal aspects of security and privacy*; • **Computing methodologies** → Neural networks;

KEYWORDS

Exposure prediction; network security; proactive security

ACM Reference Format:
Mahmood Sharif, Jumpei Urakawa, Nicolas Christin, Ayumu Kubota, and Akira Yamada. 2018. Predicting Impending Exposure to Malicious Content, from User Behavior. In *Proceedings of 2018 ACM SIGSAC Conference on Computer & Communications Security (CCS '18)*. ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3243734.3243779

1 INTRODUCTION

John typically uses his mobile device to browse a couple of news websites, read and post social media updates, and check his web mail. Today, however, there is a soccer game he really wants to watch. His TV subscription does not include a streaming option for the relevant channel; so, instead, he frantically looks for a free streaming website. His mobile browser issues several warnings, but John proceeds undeterred, and, after a number of unsuccessful attempts, manages to find a streaming site—although the website and game commentary are in Syldavian, a language he does not speak, he can watch his favorite team on the long commute. Sadly for him, a relatively common way of monetizing illicit streaming sites is to get their visitors to download malware [61], and John's phone gets compromised in the process. Over the following week, John loses access to his email account, gets invoiced for premium calls [27], and funds go missing from his bank account.

Could this have been avoided? Traditionally, mobile users have relied on blacklists and anti-viruses for protection. However, such tools suffer from several limitations: they are prone to false positives and false negatives, and cannot protect users until a site has been confirmed as malicious and has been included in a blacklist (or a specific signature has been included in an anti-virus database). In other words, attackers have a "window of opportunity" between, at least, the deployment of the malicious site and its inclusion in a blacklist (see Sec. 4.2). Further, as in John's example above, determined users may elect to willfully ignore warnings.

Perhaps, in John's case, a better approach would have been to observe that John's behavior *right before his phone got infected* was very different from his usual browsing patterns. On that day, John was quickly browsing through many pages, some in languages he does not speak, was spending very little time on each page, and, as a by-product of his repeated searches, was downloading numerous advertisements. All of these could have been indicators that John was engaging in risky behavior.

Such a proactive approach is precisely what we propose. We devise a system that *predicts* if user behavior may lead to exposure to malicious content ahead of time (e.g., 30 seconds before exposure). This, in turn, allows for various kinds of interventions to prevent compromise (rate limiting, warnings, connection termination, ...), depending on the service provider's desired level of aggressiveness. We focus on mobile users, and leverage a combination of web observations and surveys from a large mobile service provider to build our predictive engine. In particular, our system exploits self-reported data about users' security behavior, past behavioral

CCS '18, October 15–19, 2018, Toronto, ON, Canada

ACM ISBN 978-1-4503-5693-0/18/10.
https://doi.org/10.1145/3243734.3243779

observations, and contextual features about users' browsing session to predict if they will be exposed to malicious content.

Contributions. Along the way to designing our system, our paper offers a number of contributions. First, leveraging three months worth of data from over 20,600 users of a large mobile provider, we document the level of exposure of mobile users to malice online (e.g., malware), showing that at least 11% of all users get exposed at some point over our collection interval (Sec. 4). Second, we demonstrate the limitations of webpage blacklisting—using Google Safe Browsing as a case study, we find that malicious pages are frequently accessed before being blacklisted: a large number of accesses occur a few days *prior* to blacklisting, and we do see much earlier (up to 87 days prior) accesses as well. Third, our measurements also demonstrate clear differences in browsing patterns (e.g., length of browsing sessions) between users that are exposed to malicious pages and those that are not. Fourth, by surveying these users through a questionnaire (described in Sec. 3), we build a logistic regression to estimate to what extent self-reported data can provide meaningful indicators of risk exposure over time (Sec. 5). Fifth, we design a long-term predictive classifier that determines the risk of exposure to malice for a given user over a month's horizon using features based on past behavior (Sec. 6). Sixth, and most importantly, we combine the knowledge we amass from all these experiments to design a short-term classifier built from features easily computable in real-time, that can predict exposure to a malicious page within 30 seconds or so with reasonable accuracy (Sec. 7). Still, as browsing sessions in which users get exposed to malicious content are rarer than non-exposed sessions, the prediction apparatus could suffer from a unacceptably large absolute number of false positives. We use an additional source of data to show that these "false positives" frequently appear to actually be true positives, that the blacklists we used for evaluation only learned about days after user exposure. For instance, the operating point (56% true positive rate; 3% false positive rate) is likely to be, after correction, at (93% true positive rate; 1% false positive rate), which is far more practical when looking at absolute numbers; this result further demonstrates the benefits of our approach compared to a purely reactive, blacklist-based solution. We finally discuss limitations and possible interventions that our system enables in Sec. 8.

2 RELATED WORK

Our research finds itself at the intersection of four different lines of work: measurement studies to determine to what extent mobile malware poses an actual threat, systems research that goes beyond simple blacklisting for protecting end-hosts, work on security-incident prediction, and research on human factors and their impact on security.

Prevalence of Mobile Malware. Researchers studied the ecosystem of mobile malware from its early days [27, 93]. For instance, Zhou and Jiang found that more than 80% of the 1,200 malware samples they studied were repackaged versions of legitimate software, and that more than 90% turned their hosts into bots [93].

Estimates of the prevalence of malware infections greatly vary. One work estimated the prevalence of malware from DNS traffic collected by a large American cellular provider [49], and asserted that less than nine in a million devices were infected. A different team collected information about running processes on users' devices, and reached drastically different conclusions [82]—they estimated that about three per thousand devices were infected. Each estimate has its own limitations: In the former work [49], malware that uses hardcoded IP addresses may not be detected (as it does not perform DNS lookups), while, in the latter work [82], the population sample may be biased. The true fraction of infected devices probably lies somewhere in the middle.

Protecting Systems and Networks. Numerous alternatives to blacklists and anti-viruses were proposed to help protect networked systems (e.g., [32, 33, 40, 56, 57, 62, 63, 67, 82, 83, 91]). For instance, Gu et al. proposed techniques to detect bots within networks [32, 33]. As another example, Nazca detects drive-by-download attacks by inspecting the collective traffic produced by devices on the network [40]. These previously proposed alternatives intervene only at, or shortly after, the time of exposure. Instead, we propose to predict events that may lead to infection and data compromise ahead of time.

Predicting Security Incidents. Prior work studied the feasibility of predicting future computer-security incidents [34, 42, 51, 69, 71, 72, 76]. Soska and Christin showed that, using publicly available indicators, they could reliably predict whether websites would be compromised within one year [76]. Hao et al. showed they can forecast malicious domain registrations [34]. Liu et al. demonstrated that one can predict if an enterprise will suffer future security incidents (e.g., server breach), using externally observed indicators (e.g., DNS misconfigurations) [51]. Sabottke et al. focused on predicting which vulnerabilities will be exploited using information collected from Twitter feeds [69], while Kang et al. proposed to predict what percentage of hosts within a country are likely to be infected by a particular piece of malware [42].

Different efforts borrowed methods from epidemiology to understand which users and enterprises are at a higher risk of compromise [12, 47, 81, 90]. Epidemiological methods do not predict the specific individuals that will suffer from compromise. Yet, they help develop an understanding of what factors are correlated with compromise (e.g., type of operating system [12]).

Researchers have also developed methods to predict if users will be subject to compromise or whether they will visit malicious websites [9, 10, 50, 59]. Among those, Canali et al. [10]'s work is the closest to ours. Using the browsing history of ∼160,000 users collected over three months, they built a machine-learning model to predict which users will visit malicious websites with about 87% accuracy using 74 features (e.g., mean volume of user's traffic). In contrast to prior work, while we briefly explore long-term prediction in Sec. 6, our main focus is to predict visits to malicious websites on *short* timescales (seconds rather than several days or months) so that rapid intervention can take place if so desired. Our work further complements system-level measurements with self-reported answers to surveys to estimate correlations between posited and actual user behavior on a large scale. Moreover, our behavioral data is collected non-intrusively via network monitoring, thus demonstrating the feasibility of predicting exposure even with limited visibility of user activity.

Human Factors Affecting Security. Human factors in computer security have been extensively studied (e.g., [14, 28, 45, 52, 58, 64, 65, 75]). Christin et al. found that users who run anti-viruses are more likely to put their devices at risk [14]; our results reproduce this finding. Some researchers attempted to enhance the ecological validity of user studies by monitoring users' behavior via software installed on their machines [28, 45]. Surprisingly, they found that experts do *not* necessarily behave more securely than non-experts [28, 45]. Among other findings, certain types of sites (e.g., streaming and pornography) present higher risks of infection than others [45, 61, 87]. Researchers monitoring residential and enterprise networks tested which behaviors are correlated with manifestations of compromise [52]. They found that visiting blacklisted websites is highly correlated with such manifestations, thereby motivating our approach of attempting to predict such visits.

Egelman and Peer developed the Security Behavior Intentions Scale (SeBIS) as an inexpensive mean to assess users' security behavior [24]. They found that different sub-scales of SeBIS are strongly correlated with certain computer security behavior (e.g., users who score highly on the so-called proactive awareness sub-scale are less likely to be phished) [11, 21, 22]. Others have questioned whether scales such as SeBIS are truly predictive of actual behavior in the field [86]. We explore this question using the proactive awareness sub-scale of Revised SeBIS (RSeBIS)—a revised version more robust to language translation [73].

3 DATA COLLECTION

We worked with KDDI, a large Japanese mobile Internet service-provider to get access to a large corpus of user data. Customers of this mobile service provider have the opportunity to opt-in to a certain level of data collection in exchange for rewards. In particular, users who opt-in consent to the mobile carrier logging HTTP accesses over the cellular network. In June 2017, we invited a subset of customers to participate in a research survey, and obtained valid answers from 20,895 distinct users. We then analyzed the survey responses and paired them with the HTTP activity logs.

3.1 HTTP Traffic Collection

We rely on logs collected between April 1, 2017 and June 30, 2017. During that period, out of our initial pool of 20,895 consenting participants, 20,645 distinct smartphone users appear in the logs. The others presumably did not use data over cellular during that period.

Each log entry contains a timestamp of the HTTP request, the URL accessed, the content of the HTTP *Referer* field, the number of bytes uploaded and downloaded, the user-agent string, and a (unique) user ID corresponding to the customer.

Limitations. The dataset does *not* include HTTP contents (e.g., data sent via HTTP POST) or HTTPS requests, and only includes HTTP requests whose content-type is *text/html*. In other words, we do not have visibility to image-, script-, or multimedia-content access; likewise, because collection only takes place over the cellular network, we do not have access to any Wi-Fi traffic.

On a more positive note, collection is completely passive. Users do not need to change any mobile settings, install specialized software, or undertake any form of action other than providing initial consent to participate in the data collection. Furthermore, prior work has found that most malicious traffic on the web is served over HTTP [40, 93]. As a result, we believe that the data we collected reflects the nature of the users' behavior, has high visibility to malicious traffic, and is ecologically valid. While the democratization of HTTPS using services such as Let's Encrypt [2] might increase the popularity of serving malicious content over HTTPS, our proposed methods can be adapted by using domain information only, instead of the entire URL. Further, corporate networks can perform HTTPS collection using "man-in-the-middle" proxies [19].

A potential concern is that users who consented for the collection of their data may be less privacy aware than others. This is a common problem when collecting privacy-sensitive data; without a control set, we cannot estimate how well the results generalize beyond our users. Nevertheless, prior work hints that attempts at generalizing may hold promise, as users' security-behavior intentions are independent of their privacy-behavior intentions [23].

Ethics and IRB. We worked with Carnegie Mellon's Internal Review Board (IRB) and the KDDI's legal team to ensure that our usage of logs was ethical and respectful of users' privacy. The logs we use in our analysis are stored at the mobile carrier, in a secure facility, unreachable from the Internet. Thus, physical access is required every time an experiment is conducted.

Inspired by similar sensitive measurement experiments described in the literature (e.g, [54]), we wrote experimental code remotely, and tested it on synthetic records. Subsequently, one of our co-authors with authorized access to the secure facility ran the code on real data. Only aggregated results were then brought back for analysis. Any personally identifiable information from customers (e.g., IP addresses) was expunged or coded before records were exchanged. The User ID in the logs is an internal number unique to each subscriber, and is not directly linkable to any personally identifiable information (e.g., IMSI or phone number). Although a correspondence table may exist at the mobile carrier, we did not need it, and did not access it.

3.2 HTTP Log Processing

We define a *browsing session* (or, simply, a "session") as a temporally contiguous set of HTTP requests made by the same user. We consider a session effectively terminates when either 1) the associated user-agent changes (denoting the user switched browsers), or 2) the user is idle—i.e., does not engage into a subsequent HTTP request—for more than 20 minutes; we chose this specific parameter to be consistent with prior work on "click streams" [85]. (We varied this threshold in pilot experiments, and did not observe considerable changes in the 5- to 20-minute range). We point out that the goal of sessions is not to faithfully reconstruct users' web-surfing activity on their devices [88], but rather to define continuous windows of time in which users are surfing the web.

A small fraction of HTTP requests (<2.2%) in our dataset originated from traditional operating systems (e.g., Windows, Mac OS, . . .), connected to the network via tethering. We did not treat this traffic differently from other traffic, as our proposed methods are not limited to mobile devices, *per se.*

We initially marked each HTTP request as malicious or not based on a check of the Google Safe Browsing v3 (GSB, [31]) database. We will explain how we overcome some of the limitations of this approach—notably the fact that malicious URLs may not yet be in the GSB at the time of browsing—in Section 4. Because the GSB database is highly dynamic (entries are added and subtracted every day), we downloaded daily snapshots of the GSB database throughout our measurement interval. Specifically, for every hash prefix available in GSB on a given day, we queried the GSB API [31] to download the full hashes that start with the prefix.

GSB distinguishes between phishing and malware URLs. The former purportedly entice users to reveal private information, while the latter attempt to deliver unwanted programs to users' devices [31]. In practice, we noticed that certain entries that are labeled as phishing by GSB may lead users to download malicious software or extensions, while others may lead to ad- and click-fraud pages. Such entries can be construed as malware, as they may harm the users' devices or online (e.g., ad) services. In fact, using VirusTotal [15]—a popular service that combines reports from several blacklists—we found that 20 out of 25 randomly sampled domains that are classified as phishing by GSB are also classified as malware by one or more other lists. For example, hxxp://applicationg29.com, which was previously classified as a phishing domain by GSB, leads to a page for downloading a fake anti-virus when visited with certain URL parameters. Hence, the distinction may not be as critical as we originally thought; both types of URLs are harmful to the user, and potentially to online services. For the rest of this paper, we consider all entries as malicious.

We also divided URLs visited upon each HTTP request into categories (e.g., news, sports, ...). To do so, we relied on the taxonomy developed by DigitalArts, the main filtering provider in Japan, for their i-Filter filtering system [39]. Using manually labeled domains by DigitalArts, we trained a Convolutional Neural Network [92] to classify domain names into one of 99 topics. The neural network achieves ~90% accuracy in assigning URLs to the correct topic. Finally, we classified users between *exposed* and *unexposed*. Exposed users visited malicious pages (per the above definition) at least once during our collection interval; unexposed users did not.

3.3 Online Survey

A contribution of our work is to validate whether users' self-reported responses to computer-security surveys can help predict their behavior. Specifically, we are interested in exploring whether survey answers can predict users' exposure risk. To this end, we asked users who consented to participate in this research to respond to an online survey.

Recruitment. In June 2017, we invited 600,000 people drawn from the pool of eligible, customers (i.e., those who opted in to having data collected) to participate in a research survey. All customers were based in Japan. As an incentive for customers to respond, we offered them the chance to enter a drawing to win a 500 JPY (roughly equivalent to $5) gift card. 1,000 randomly selected respondents received this prize.

We initially received 23,419 user responses. While the 3.90% response rate we recorded is below response rates associated with online surveys (thought to be in the 10–20% range [53]) it is in-line with surveys conducted in similar online security studies (e.g., 3.4% [48]). Furthermore, given the large population sample we considered, a high survey response rate is unnecessary, as long as the pool of respondents is not significantly biased.

Demographics. The pool of respondents was slightly biased toward male users: indeed, 61.5% male and 38.5% female users responded. (The pool of solicited users was 55.6% male, 42.8% female, 1.6% unknown.) The respondents' median age was 43 years, with a standard deviation of 11.8 years. The demographics are not necessarily closely tracking the overall population. For example, elderly users might prefer traditional flip or feature phones rather than smartphones. Moreover, due to ethical reasons and the difficulty of acquiring guardians' consent, our pool of respondents excludes minors below 18 years of age. With this in mind, the demographics of our respondents mirror those of our solicited users fairly closely, showing no specific evidence of bias.

We were unable to immediately assert the proportion of iOS and Android users among respondents, since we did not collect this information in the survey responses. We elected not to perform user-agent matching after the fact, as it would have been very noisy: while Safari and Google Chrome users on iPhone and Android might be classified relatively accurately, numerous other browsers may not use user-agent strings very representative of the platform on which they run.

Finally, we eliminated responses that did not pass attention checks. This yielded valid answers from 20,895 distinct users.

Questions. Besides demographic questions (gender and age), we asked participants a number of behavioral questions. We speculated that self-reported behavior is correlated with actual behavior based on prior evidence. The behavioral questions we inquired about are:

(1) Whether users have encountered security incidents (e.g., stolen password). Suffering from security incidents is often correlated with users' advice sources [64], and hence might affect their behavior.
(2) Whether an anti-virus is installed on users' devices. Users running anti-viruses might be more likely to engage in risky behavior [14].
(3) The types of app marketplaces usually used. Participants had the option to select official (e.g., Google Play), mobile providers', or other, unofficial, marketplaces. Prior work has shown that unofficial marketplaces contain a high fraction of malicious apps [94]. Thus, we expected users who use such marketplaces to engage in more risky behavior.
(4) What step(s) the participants take their browsers warn them about a malicious webpage (e.g., always proceed, . . .). We showed participants the Chrome warning page. As Chrome has the largest market share among browsers [77], participants are most likely to be familiar with it. We expected participants who proceed on warning to be at higher risk.
(5) Participants' responses to the proactive awareness sub-scale of RSeBIS. Participants with high proactive awareness scores are less likely to fall for phishing than others [21]. Thus, we expected high-scoring participants to also be at lower risk of visiting malicious websites.
(6) Participants' self-confidence in their computer security knowledge. We used the questions from Sawaya et al. [73] to measure self-confidence in security knowledge. We expected confident participants to exhibit more secure behavior.

We provide the entire survey in Appendix A. Since the users are based in Japan, we ran the survey in Japanese. For the proactive awareness sub-scale and the self-confidence questions, we used the RSeBIS Japanese translation of Sawaya et al. [73]. Other survey questions were translated by computer-security experts in our group who are fluent in both English and Japanese.

4 EXPOSURE TO MALICE

We next delve into the analysis of the HTTP logs. This analysis has three goals: (1) to determine to what extent mobile users are exposed to malicious content, (2) to demonstrate that there is a "window of opportunity" for miscreants to compromise devices before a page is blacklisted, and document plausible sizes of that window, and, (3) to explore differences in behavior across several dimensions between exposed and unexposed users. The analysis will serve as the bedrock for feature engineering in the predictive models we build and evaluate in Sec. 6–7.

4.1 Overall Prevalence of User Exposure

As discussed in Section 2, there is some disagreement in the research community regarding the actual prevalence of malware in the mobile ecosystem. Specifically, previously reported results [49, 82] differ by several orders of magnitudes in their estimates. This motivates our own investigation based on the logs we collected.

Among the 20,645 users for whom we have HTTP logs (between April 1, 2017 and June 30, 2017), 2,172 (11%) users accessed a malicious page at least once and were thus exposed. Most of these users (1,995) were exposed to pages that GSB classifies as a "phishing" page; 153 users were exposed to "malware" pages, and 24 users were exposed to "malware" and "phishing" pages. The GSB database also features entries for "unwanted" pages, but none of our users appear to have landed on such pages. Overall, the exposed users visited 3,491 unique malicious pages on 201 different domains.

In short, at least 0.81% of all users (those visiting "malware" or "malware and phishing" pages) were exposed to confirmed malware. We cannot, however, estimate the fraction of exposed users that were actually infected. Indeed, these users might have been protected by an anti-virus, or other content filters. More interesting to us is the fact that a considerable (11%) fraction of *all* users actually get *exposed* to questionable content, further motivating our research in attempting to prevent such exposure in the first place.

4.2 Window of Exposure

We next estimate the amount of time users might be exposed to malicious pages while defenses may not yet be in place. Our approach is similar to a prior approach to measure the preponderance of zero-day exploits in the wild, by retroactively checking evidence of malware signatures in telemetry data dating back to times before these signatures were known to anti-virus companies [8].

Similarly, we check if we observe logs of accesses to URLs before these URLs were included in the GSB database. Fig. 1 plots, as a function of time, the number of accesses to pages present in the GSB database during our measurement interval (April 1, 2017–June 30, 2017). Negative x-axis values denote accesses to a page *before* it was included in the GSB database, while positive values denote accesses occurring after inclusion.

We immediately point out that our analysis presents some limitations. First, because we work with a finite, 91-day interval, there is only one possible day for us to observe a URL being accessed 90-days prior (resp. after) inclusion in the GSB: the first (resp. last) day of our measurement interval. Conversely, there are 90 days during which we can observe an access one-day prior (resp. after) inclusion in the GSB. Thus, values closer to 0 on the x-axis of our plot will be over-represented. To account for this imbalance, we normalized all y values by dividing them by the number of days we could actually observe them—dividing the y-value corresponding to $x = 0$ by 91, the y-values corresponding to $x = \pm 1$ by 90, and so forth. For the sake of brevity, we omit this plot here: the overall shape of the curves does not markedly change.

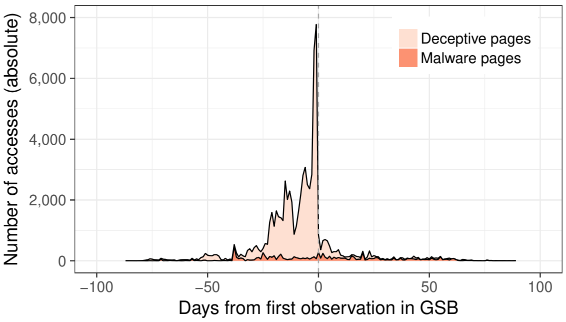


Figure 1: Total number of accesses to malicious pages as a function of the number of days to pages' first observation in GSB. The plot is stacked. Negative x-axis values indicate possible exposure to malicious pages not yet in GSB.

Second, we can estimate that a page ceases to be malicious when it is removed from the GSB database. However, we do not know when a page *starts* being malicious. In the best case, it could be on the day it was included in the GSB, in which case prior accesses would not be problematic. In the worst case, it could have been malicious more than 91 days prior to inclusion, in which case, *all* of the accesses we observe would be problematic.

With these caveats in mind, it is striking to see that certain deceptive pages were accessed a full 87 days before inclusion in GSB (leftmost point on the graph). We observe a first increase in accesses to malicious pages 38 days before inclusion in the GSB database, and a second increase 22 days before inclusion in GSB. We further observe a large spike two days before inclusion in the database—intuitively, an increased number of accesses to a malicious page increases its likelihood of being flagged by GSB, but at the same time, a large number of users are exposed to the page.

Unsurprisingly, we observe an immediate overall drop-off after inclusion in the GSB database, which shows that GSB-based mitigations appear to be quite efficient. Nevertheless, we do see a number of accesses even after a page has been included. These accesses could be due to users ignoring warnings, or to their browsers not using the GSB database to filter access. Particularly interesting to us is that accesses to malware-infested pages do not seem to significantly decrease after inclusion in the GSB database. This could plausibly indicate that users are deliberately visiting such pages (bypassing defenses in the process), or that, once infected with malware, a device keeps on (unbeknownst to the user) requesting HTTP content from malicious locations.

We observed close to three months of potential, unprotected exposure, to malicious pages, which could have led to successful infections or credential theft. The probability of exposure increases sharply about three weeks prior to pages' blacklisting, and even more sharply in the couple of days prior to inclusion. To summarize:

Finding 1. *There is evidence of non-negligible delays—potentially in the order of days or weeks—in certain pages being flagged as malicious by GSB.*

These measurements demonstrate the limits of defenses purely based on blacklists such as the GSB database, and motivate the need for additional proactive measures such as the ones we propose in this paper.

4.3 Behavioral Differences Between Users

Next, we are interested in using our logs to explore any differences in behavior between exposed and unexposed users. The idea is that such differences could be critical in identifying risky behavior, which we could then use for feature engineering in the design of a predictive classifier. Here, we are only concerning ourselves with empirically-measured differences—we do not (yet) integrate any data from our surveys.

Defining Malice. Ideally, we would be able to identify precisely when a user gets exposed to a malicious page, and we would be able to characterize her behavior long before, immediately prior to, and after exposure. This would in turn help us determine possible deviations that could be indicative of increased risk immediately prior to exposure, and characterize whether an infection might have occurred post-exposure (e.g., if we notice drastically different network access patterns). Unfortunately, we have no way of knowing precisely when a user is exposed. We know when a user requests a given page, and we know when that page is flagged as malicious by GSB, but, as discussed above and shown in Fig. 1, we do not know when the page actually *turns* malicious. For that reason, we parameterize the notion of a malicious page.

Definition 1. τ-malicious page: *Consider a user making an HTTP request to a URL u at time t. URL u is deemed τ-malicious if it appears in the GSB database at any time t' such that $t \leq t' < t + \tau$.*

τ-malicious pages are webpages that may *currently* be flagged as benign, but that will (within τ) be marked as malicious. The smaller τ is, the higher the danger, as the probability that the page is in fact already malicious increases as $\tau \longrightarrow 0$.

Mathematically, using $\tau = 0$ (resp. $\tau \longrightarrow \infty$) would underestimate (resp. overestimate) the number of malicious pages—the "true" value for τ actually depends on each URL. Based on the results shown in Fig. 1, it appears fruitful to study behavioral differences between users exposed to τ-malicious pages for $\tau \geq 87$ days (oldest plausible appearance of a malicious page before it was labeled as such), $\tau = 22$ days (when number of accesses to webpages ultimately deemed malicious starts to significantly pick up), $\tau = 2$ days (when the spike in accesses is the highest), $\tau = 0$ days (when malice has been confirmed), and unexposed users.

Exposure Events. Fig. 2 shows the number of requests to malicious pages for each exposed user for various definitions of a malicious page, ranging from $\tau = 0$ to setting τ to the maximum value possible (91 days, the size of the measurement interval).

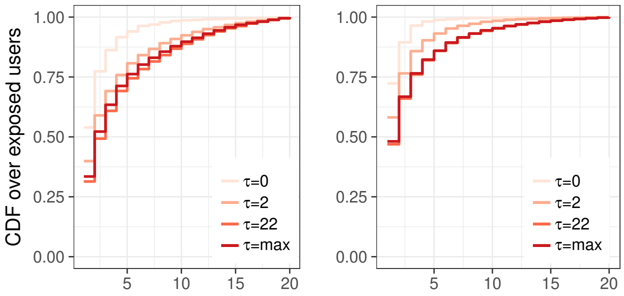


Figure 2: Number of requests (left) and sessions (right) to malicious pages for exposed users. For readability, we truncate the figure at 20 requests or sessions, even though a small fraction of users ($< 0.1\%$) request far more.

Depending on the value of τ, between a third to half of our exposed users access only one page deemed malicious; for positive values of τ, more than a quarter of our users make three or more requests to malicious pages. Fig. 2 indicates that, for positive values of τ, half to two-thirds of the exposed users perform all exposed requests within a single session depending on the value selected for τ; the remainder shows a long-tailed distribution. As the figure shows, this behavior is generally robust to changes in the value of τ we select, although, in line with Fig. 1, $\tau = 0$ seems overly conservative.

The limited number of exposure events for a third to half of our population motivates the need for short-term, *in-session* predictions for each user.

Finding 2. *A predictive classifier cannot* purely *rely on previous exposure, since a significant share of our user corpus shows a lack of "repeat" exposure.*

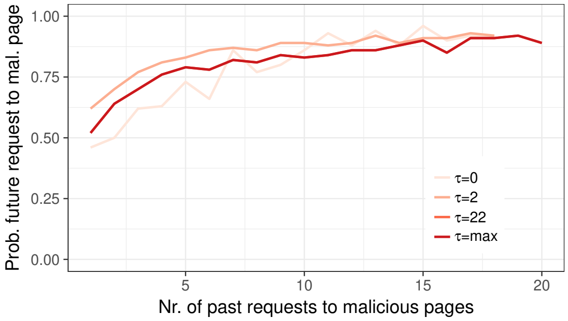
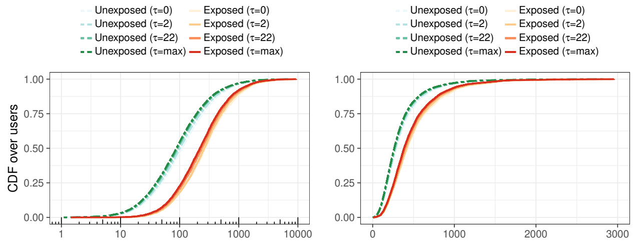


Figure 3: Probability of future visits to malicious pages based on the number of past requests to malicious pages. The overall trend is relatively independent of τ. Curves for $\tau = 22$ and $\tau = \max$ are overlapping.

Next, we calculate, in Fig. 3, the probability that an exposed user who accessed malicious pages at least x times in the past, will access a malicious page in the future. Namely, Fig. 3 reports the empirical estimate of $Pr[A(x+1)|A(x)]$, where $A(x)$ denotes the event that a user visited at least x malicious pages. Here too, we vary τ. We find that:



(a) Requests/user/day **(b) Avg. session length (sec)**

Figure 4: Differences between exposed and unexposed users. Exposed users tend to make far more requests, and engage in longer sessions than unexposed users. Kolmogorov-Sminov (KS) tests show that the differences between exposed and unexposed users are statistically significant (p-value <0.01).

Finding 3. *The more users got exposed in the past, the higher the probability they will get exposed again.*

This finding substantiates that, while prior exposure by itself is insufficient (see Finding 2), short-term predictions can nevertheless benefit from long-term inputs, using prior exposure as a feature. Namely, this measurements motivates the need for combining short- and long-term reasoning, which will be our core contribution.

Session-Level Metrics. We next investigate the differences in the level of activity between exposed and unexposed users. Fig. 4(a) (in logarithmic scale on the x-axis) shows that, regardless of the value chosen for τ, unexposed users generally request far less pages per day than exposed users. We observe a similar trend for sessions—unexposed users engage in considerably less sessions than exposed users. Finally, Fig. 4(b) exhibits clear differences in session lengths: unexposed users engage in usually shorter sessions than exposed users. Here too, the result is robust to the value of τ we choose. In short, irrespective of the value τ chosen, we see that:

Finding 4. *Exposed users are more active than unexposed users—they make more HTTP requests, and engage in more, longer, browsing sessions.*

These variables will thus play an important role in a predictive classification model.

Diurnal and Weekly Effects. We next look into the amount of requests by exposed and unexposed users through the day by the hour, as shown in Fig. 5. To meaningfully compare the different classes we perform "feature scaling," i.e., we normalize each value to a [0, 1] range, where 0 represents the minimum number of requests from a given type of user per day, and 1 is the maximum number of requests.

All requests in our corpus occur on the same time zone (JST). Both behaviors are relatively similar and demonstrate time-of-the-day effects: people browse the Internet most during lunch time, and the early morning (midnight-4am) is the quietest time of the day. However, it seems that exposed users tend to request data more evenly throughout the day; in particular, the gap with unexposed users is most pronounced in the evening. Canali et al. found similar differences when analyzing user browsing on traditional operating systems [10]. We conjecture this may be due to a combination of exposed users 1) browsing the Internet more often than unexposed users overall, and 2) potentially browsing riskier websites in the evening than when they are at work.

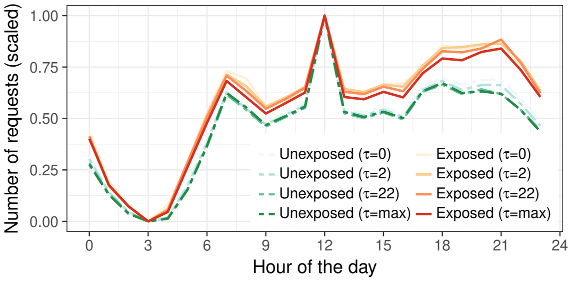


Figure 5: Scaled number of requests by type of user over the day. Due to imbalance in the absolute numbers among classes, we use a [0, 1] feature scaling. Exposed users are more active at night time. KS tests show that the differences between exposed and unexposed users are significant (p-value <0.01).

Finding 5. *Exposed users tend to browse the Internet more frequently at night and outside of working hours.*

Thus, time-of-the-day might be a useful feature to look at for our predictive classifiers. For the sake of brevity, we do not present a plot here, but we found a similar behavior relating to activity levels throughout the week: exposed users tend to be more active on week-ends than their unexposed counterparts. We will thus also consider day-of-the-week as a potential predictive feature.

Webpage Categories. Finally, motivated by prior findings that certain types of webpages present higher risks of infection than others [45, 61, 87], we posit that certain categories of webpages may be more strongly correlated with exposure risks. For example, Fig. 6 shows the proportion of advertising and adult webpages for unexposed and exposed users. Although we might overlook some requests to advertising pages due to our logs ignoring, e.g., JavaScript, we observe that exposed users tend to access advertising pages with a higher frequency than unexposed users. The same is true for adult content. More generally, we find that exposed and unexposed users access webpages of 65 categories out of the 99 at different frequencies (these differences are statistically significant with p-value < 0.05 according to Kolmogorov-Smirnov tests, after Bonferroni correction). Exposed users are more likely to access webpages of 19 categories (e.g., adult, advertising, and video-search webpages), while unexposed users are more likely to access webpages of 46 categories (e.g., education, financial, and news webpages).

Finding 6. *Certain categories of content may be indicative of higher risk exposure.*

We thus might want to use website category as a feature in our predictive classifier.

Take-Aways. As a major take-away from this set of measurements, exposed and unexposed users clearly show differences along a number of metrics. We will use these differences in the remainder of the paper to devise proactive defenses. Another important take-away is that the notion of "exposure" itself is highly dependent on

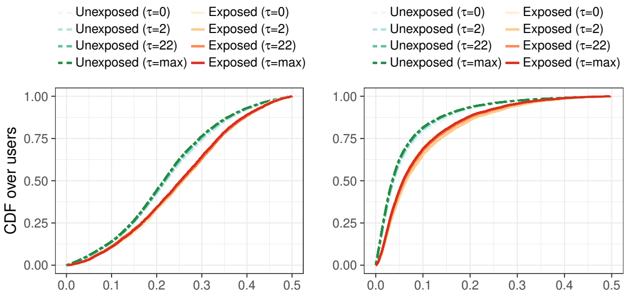


Figure 6: Fraction of page requests of a certain type over all requests. Exposed users tend to request more advertisement (left) and adult (right) pages than unexposed users. The differences between exposed and unexposed users are significant according to KS tests (p-value $<$0.01).

what is considered malicious. We have seen that blacklists such as the GSB database appear to lag a bit behind actual deployment of malicious pages, and have introduced the notion of τ-exposure—the user is exposed to a page that will be labeled as malicious within τ days. Measurements indicate that $\tau = 2$ seems to be a reasonable compromise: larger τ would probably encompass pages that were not actually malicious at the time of browsing, but smaller τ would likely miss dangerous pages.

5 SURVEY RESPONSES AND EXPOSURE RISK

We now turn to determining whether we can accurately predict the risk of a compromise to a given individual. We start from a general perspective: can we establish meaningful insights on users' exposure risk from survey responses? If so, we could use these insights to complement the system-level metrics we collect, and improve performance of our proposed predictive schemes.

In essence, we aim to explore the relationship between a binary dependent variable (whether users are exposed) and a mix of categorical (e.g., whether an anti-virus is installed) and continuous (e.g., proactive awareness score) variables. To this end, we build a logistic regression model (a popular tool for modeling the relationship between various explanatory variables and a binary outcome [74]) to determine the influence of the different variables our survey captures on user exposure risk.

For each explanatory variable, logistic regression assigns a coefficient that estimates the change in the log odds-ratio of the dependent variable. In other words, the coefficients estimate the change in $ln\left(\frac{p}{1-p}\right)$, where p is the probability of the user being at risk. For example, if the coefficient of the binary variable indicating whether a user reported to have an anti-virus installed is estimated to be $\beta > 0$, then the odds of visiting malicious sites are e^{β} times higher for users who report to have an anti-virus on their device than those who do not. Alternately, if β is negative, then the odds of being at risk decrease for users who have an anti-virus.

5.1 Experiment Design

We use the three-month worth of HTTP requests we collected, and the survey responses to build the logistic regression. Using $\tau = 2$ we identify which users are exposed from the HTTP requests—exposure will serve as the dependent variable in the regression. The independent variables are constructed from the survey responses. Specifically, we compute the following variables: (1) Gender; (2) Presence of an anti-virus on the user's device; (3) Whether the user downloads apps from unofficial marketplaces; (4) Whether the user proceeds on browsers' warnings; (5) Whether the user reports having suffered from a compromise; and (6) the RSeBIS proactive awareness score (via summing the users' responses to the Likert-scale questions and normalizing to [0, 1] range). Variables (1)–(5) are binary; (6) is continuous.

Table 1: Parameter estimates for the logistic regression model. Shown: log odds-ratios of variables, their odds-ratios, and the p-values for H_0 that the odds-ratios are equal to 1.

Parameter	β	e^{β}	p**-value**
(Intercept)	-1.97	0.14	<0.01
Is female?	-0.50	0.54	<0.01
Proactive awareness	-0.20	0.82	<0.01
Proceeds on browser warning?	0.23	1.26	<0.01
Suffered from compromise?	0.51	1.67	<0.01
Uses anti-virus?	0.92	2.51	<0.01
Uses unofficial app marketplace?	0.16	1.17	<0.01

We exclude age and self-confidence in security knowledge from the regression—exploratory data analysis did not indicate correlation between age and exposure; and we wanted to avoid multicollinearity in the regression due to the correlation between self-confidence in security knowledge the proactive awareness score [73].

We use Python's `statsmodels` package to build the regression model [41]. To select the best model, we begin from all possible interactions between the variables and perform backward model selection by removing interactions until the model's likelihood does not decrease much (particularly, the decrease in the Bayesian Information Criterion becomes lower than two, as standard [74]).

5.2 Experiment Results

The parameter estimates of the model with the best fit are reported in Table 1. While none of the interactions survived model selection, all the main factors in the model were found to have a significant effect on exposure at a significance level $p < 0.01$. Notably, the model estimates that women are 0.54 times as likely as men to visit malicious URLs. Prior work showed that women are less likely to detect deceptive, malicious, webpages [75], but this finding shows that women may be less likely to encounter such webpages in the first place. Participants who have suffered from compromise have an increased odds of exposure to malicious content by 1.67 times. This further corroborates Finding 3 that users with prior exposure have higher probability to be exposed again. Similarly, the parameter estimates show that risky behavior, such as proceeding on browser warnings, and using unofficial marketplaces for mobile apps, increase the odds of exposure to malicious content by at least 1.17 times, on average. This aligns with our expectations. In contrast, and somewhat surprisingly, having an anti-virus is the factor that is most correlated with exposure—users who reported to have an anti-virus were 2.51 times more likely than others to visit malicious URLs. We hypothesize that these users have a false sense of security that leads them to engage in a high-risk behavior—something that was previously observed by others [14]. It may also be the case that the users installed an anti-virus in response to a past exposure. Finally, users who achieved full proactive awareness score on RSeBIS were 0.82 times as likely as those who achieved the lowest score to visit malicious URLs, supporting prior evidence that the scale correlates with certain security behavior [11, 21, 22].

Our analysis provides insights into how self-reported information correlates with actual exposure to malicious content, and how different types of information collected via self-reporting differ in their importance. Yet, the best model we could build using the data we asked about can explain only 5% of the variance (i.e., the $R^2 = 0.05$). Thus, despite being rather diverse and certainly helpful, the self-reported information we considered may be insufficient *by itself* to fully explain exposure to malicious content online. We have to complement it with other features, as shown next.

6 LONG-TERM PREDICTION OF EXPOSURE

Now we move from exploring the relationship between self-reported information and exposure to predicting exposure. In particular, we evaluate predictive models that leverage self-reported answers to surveys, past behavioral observations, or both to predict user exposure over a relatively long time period (specifically, one month). Here, as opposed to the previous section, models that rely solely on survey responses help us estimate how accurately can self-reported data predict exposure. Additionally, models that rely on past behavioral observations of users help set a baseline to evaluate the utility of self-reported data for prediction.

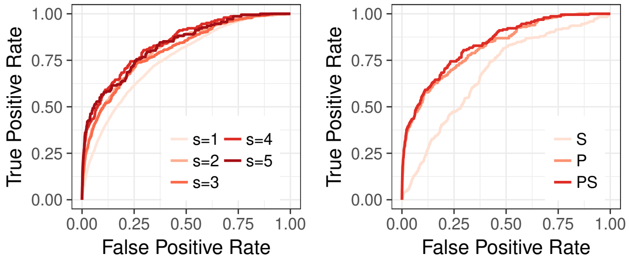
6.1 Classifier Design

Self-Reported Features. From the survey responses, we extract seven features for prediction. Six of the features are the same as described in Sec. 5.1. The seventh denotes users' self-confidence in their security knowledge (computed via summing the responses to the Likert-scale questions then normalizing to the [0, 1] range).

Past-Behavior Features. The main premise when using Past-Behavior Features is that past behavior is indicative of future one (e.g., users who visited malicious domains in the past are likely to visit malicious domains in the future [10]). The features that we develop are motivated by the findings of Sec. 4 regarding the differences between exposed and unexposed users. Some of the features quantify the user's amount of activity per day: average number of sessions and HTTP requests, average number of bytes uploaded and downloaded, and average session length in seconds and number of requests. Two features are used to estimate the amount of past exposure: One feature indicates prior exposure, and another quantifies the fraction of previously exposed sessions. 24 features summarize the level of activity during different hours of the day, and 99 features describe how previous HTTP requests are distributed among the 99 topics of DigitalArts [39]. Finally, one feature reports the fraction of request to domains not in the Alexa top 100,000 websites. The premise behind the last feature is that top websites are unlikely to be malicious or to link to malicious content. Overall, we use 132 Past-Behavior Features.

Table 2: Summary of the classifiers' design.

	Long term (Sec. 6)	Short term (Sec. 7)
Classifier	Random Decision Forest	Conv. neural network
Input	Past-Behavior and/or Self-Reported features	Contextual, and, possibly, Past-Behavior and/or Self-Reported features
Output	Probability estimate of user exposure	Probability estimate of exposure within session
Class balance	18:82	0.1:99.9



(a) Threshold effect (b) Feature set effect

Figure 7: ROC curves for long-term predictions. Left: the effect of the threshold (as a number s of previously exposed sessions) using all (PS) features. Right: the effect of the various feature sets, using a threshold of $s = 4$ sessions.

Prediction Algorithm. We use Random Decision Forests [37], a popular machine-learning algorithm, to perform predictions. For best performance, we set the number of classification trees to 50, the maximum height to 5, and weigh negative examples by ×0.1 the weight of positive ones. We also experimented with other classification algorithms, but they did not perform as well. Table 2 summarizes the classifier's design.

6.2 Experiment Design

Essentially, the question that we aim to answer is: based on users' self-reported data and/or browsing behavior in a specific month, can we predict exposure in the following month? To answer this question, we use our data to train Random Decision Forests to predict exposure in May ($\tau = 2$) using observations made in April, and evaluate how well these models can predict exposure in June via observations made in May. We perform ten-fold cross-validation rounds in which we use 90% of users for training and 10% of users for testing.

6.3 Experiment Results

Fig. 7 shows the overall classifier performance as a function of two parameters. Using our complete feature set (Self-Reported and Past Behavior), Fig. 7(a) shows the influence of the session threshold s. s is used to denote the number of distinct exposed sessions a user has to endure to be considered as exposed. Intuitively, the higher s, the more conservative the notion of user exposure. In particular, with $s = 1$, a user might have landed on an exposed webpage by accident; with $s = 5$, on the other hand, the user is repeatedly getting exposed.

Unsurprisingly, for long-term predictions, a higher s seems to work better. Fig. 7(b) shows the impact of both feature sets: S denotes the impact of self-reported features, P, that of past-behavior features, and PS the impact of considering both sets. As expected, the whole set (PS) achieves the best results, but more interestingly, the impact of the Self-Reported Features by themselves is very modest. While, in isolation, Self-Reported Features might be useful (e.g., if user monitoring is not an option), the Past-Behavior Features achieve much better performance on their own; and combining them with the Self-Reported Features does not yield a great improvement.

Our long-term prediction system performs slightly worse than Canali et al.'s system [10]. However, we use a far more restricted set of features: our data collection is less intrusive, we only collect text/html content, and most importantly, our features are considerably cheaper to compute (e.g., most can be computed using counters).

Regardless of the accuracy of long-term predictions, they are not a panacea—recall from Sec. 4 that many users are actually only exposed once or twice (Finding 2). This begs for an additional set of classification primitives, that attempt to prevent exposure on much shorter time intervals.

7 WITHIN-SESSION EXPOSURE PREDICTION

We next explore the feasibility of predicting whether a user will engage in risky behavior (namely, visiting a malicious website) *during a single browsing session*. Thus, we aim to use contextual features that describe the session at some stage, to predict whether the user will be exposed at a later stage in the session. Each time the user makes an HTTP request, the contextual features are updated to describe the new state of the session, and another prediction attempt is made. As Sec. 6 and prior work have shown, users' behavioral features (e.g., number of websites visited [50]) can help predict user exposure risk over long periods of time (e.g., three months [10]). We next show how to use similar behavioral features to predict exposure on much shorter time scales.

7.1 Classifier Design

The prediction problem we aim to solve poses two key challenges. First, the dataset is imbalanced—there are about 1,733 benign sessions for each exposed session. Learning from imbalanced datasets is hard, as classifiers which predict the majority class most of the time, albeit useless, would be favored by training algorithms due to their high accuracy [36]. We address this challenge by undersampling the majority class during training. Other methods propose to oversample the minority class (e.g., SMOTE [13] and ADASYN [35]), but we did not find them useful. Second, browsing sessions contain large numbers of HTTP requests. Thus, the prediction algorithm and the feature computation need to be highly efficient to be deployed in real-time. We thus design features that can be efficiently updated after each HTTP request, and rely on a compact neural network for prediction.

Contextual Features. From the observations presented in Sec. 4, we develop a set of session-level features to predict whether a session will become exposed. In essence, the Contextual Features are similar to the Past-Behavior Features from Sec. 6, but are computed over the course of the session, rather than over the entire historical observations we have about the user.

Since exposed sessions tend to have a higher amount of activity, features quantifying magnitude of activity are a good proxy for our predictions. These include session length (in seconds), number of HTTP requests, and amount of bytes transferred during the session.

Malicious domains are less likely to be (directly) linked to from top-domains. So, we use the fraction of HTTP requests to non-top domains (i.e., outside the Alexa top-100,000) as another feature.

Exposed sessions are more likely to occur in the weekends, and late in the day (Finding 5). We thus use a feature to indicate whether the session is taking place during the weekend, and 24 features to indicate the hours in which the session has been active.

Six binary features indicate the operating system and the browser observed in the session (as learned from the user-agent strings in HTTP headers): Two of the features indicate Android and iOS (the two most popular mobile OSes [80]), two features indicate the use of Chrome and Safari (the most popular browsers [77]), and two features are used to indicate other OSes and browsers (e.g., Firefox OS). While our exploratory data analysis and prior work [49] did not indicate stark differences in the risk profile of different systems (e.g., iOS vs. Android), these features could help capture the subtle differences between systems (e.g., how often browser blacklists are updated) that could affect users' exposure.

Last, 99 features describe the topics of domains visited in the session. As prior work has shown, and in line with Finding 6, certain website categories (e.g., online streaming [61]) tend to exhibit more malicious activity than others. Hence, visits to such websites are likely to increase the likelihood of exposure. The features we developed reflect the distribution of domains in the HTTP requests among the 99 Digital Arts categories [39].

All 135 contextual features we use can be computed efficiently throughout sessions via counters and table lookups. Some of the models presented in this section also rely on the Past-Behavior and Self-Reported Features from Sec. 6. For a given session, we compute the Past-Behavior Features using the user's history up to the beginning of that session. Once the session is over, we update the Past-Behavior Features based on the observations made during the session.

Prediction Algorithm. We use convolutional deep neural networks (DNNs) to perform predictions [30]. DNNs can achieve state-of-the-art performance on many different tasks ranging from image classification [7] to speech recognition (e.g., [70, 89]) to playing Go [16]. Besides their high performance, DNNs are practically useful when training with large datasets as their parameters can be learned iteratively using small batches of data.

We train DNNs using data from the early parts of our collection and evaluate them using data from later parts. As the goal is to predict exposure to malicious content before the exposure takes place, HTTP requests in exposed sessions that are made to or after the visits of malicious URLs are disregarded. We create a feature vector for each HTTP request in the training set. We assign a positive label (i.e., 1) to HTTP requests that belong to exposed sessions, and a negative label (i.e., 0) to the remaining requests. During the training process we do not use HTTP requests in exposed sessions that are made more than one minute before the moment of exposure. This apparently improves performance, possibly because requests that are made long before exposure takes place are unlikely to play an influential role in the exposure.

Additionally, both in training and testing, we disregard the first nine HTTP requests of each session. These requests are needed to create context about the browsing sessions and to bootstrap the Contextual Features. In other words, we perform predictions only for sessions with more than nine requests. Consequently, we only keep sessions with more than nine requests for training and testing (these include ~96% of all exposed sessions). Using Selenium [18] and tPacketCapture [79], we crawled the Alexa top 100 HTTP-only websites on an Android phone, and found that, on average, two website visits correspond to ten HTTP requests. So, our proposed system would begin analyzing a user browsing activity after she has visited two websites, on average.

In each training epoch of the DNN, to address dataset imbalance, we only use a random subset of the negative examples for training, such that the number of positive and negative samples is equal [36]. At test time, we classify a session as exposed only if the feature vector describing the state of the session at a certain stage is classified as positive.

The DNNs we use are sequential, and consist of three convolutional layers, each followed by a Rectified Linear activation, a fully-connected layer, and a *softmax* layer. The convolutions use 5×128 kernels, and are applied with a stride of one. In training, the DNNs' the *cross-entropy* loss is minimized. This is a standard choice of architecture and training strategy [29, 30, 68]. Before selecting the convolutional architecture, we initially tested a neural network architecture consisting of fully-connected layers only, finding it to perform less accurate predictions.

We normalize the DNNs' inputs to a $[0, 1]$ range using statistics learned from the training data, as normalization helps to speed up training and improve performance [17]. We set the learning rate to 5×10^{-5}, the batch size to 128, and the number of epochs to 50. We performed a grid search to set the hyper parameters, optimize depth of the DNNs, and the size of the convolution kernels. We implement the DNNs in Keras [43] (with a TensorFlow backend [1]). Table 2 summarizes the predictor's design.

7.2 Experiment Design

We trained several DNNs to predict whether sessions will become exposed. Since we empirically found that models that are trained using data collected over the range of 15 consecutive days perform well when used for prediction in the five days that follow, we picked different periods of 20 days each (of which the first 15 days were used for training, and the last five for testing) to evaluate our approach. Specifically, we used five different periods of 20 days that were uniformly spread over three months of our data collection, starting from April 5th, 2017. We did not start from April 1st so that we can use the first days of April to observe the behavior of users and initialize the Past-Behavior Features to a meaningful state.

Similarly to Sec. 4.3, we considered a page to be malicious if it had been in GSB, or appeared in GSB within two days or less from the time it had been accessed by a user (i.e., we set $\tau = 2$). This required special care when training the DNNs. Specifically, when training a DNN at time t, one cannot tell if accesses in the interval $t - \tau + 1, \dots, t$ appeared in GSB within the time interval $t + 1, \dots, t + \tau$. Thus, to avoid mislabeled data in our training, we exclude HTTP requests made in within $t - \tau + 1, \dots, t$ from training (in our case, we exclude HTTP requests made in $t - 1$ and t from training).

To speed up the training process and minimize the use of private information, we uniformly sampled the HTTP requests of unexposed sessions during training. Specifically, we used only 5% of HTTP requests of unexposed sessions in training. We avoid similar sampling of HTTP requests belonging to exposed sessions, as their absolute number is relatively small. Overall, to train the DNNs, we used ~3,200,000 requests from unexposed sessions and ~43,000 requests from exposed sessions, per evaluation period For testing, we had an average of 412,005 benign sessions (containing ~21,620,410 requests) and an average of 277 exposed sessions (containing ~19,457 requests), per evaluation period.

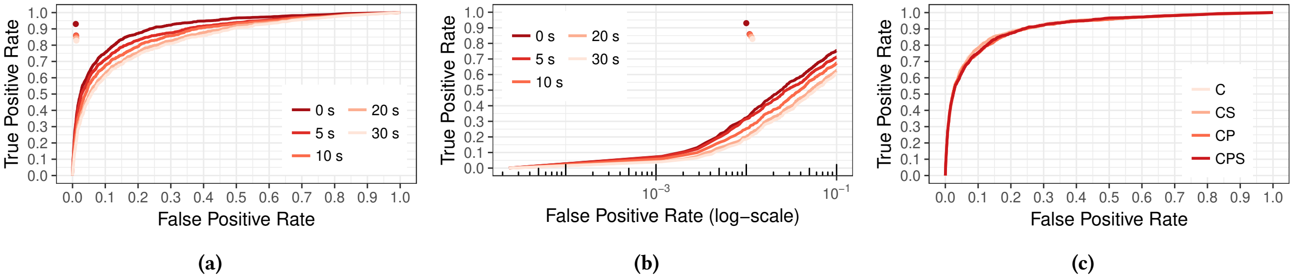
7.3 Experiment Results

We first tested how accurately do the DNNs predict exposure in advance. Specifically, we measured the true-positive rate (TPR; the rate in which exposed sessions were classified as exposed) and the false-positive rate (FPR; the rate in which unexposed sessions were classified as exposed) of the DNNs at different times before exposure. In these experiments we used all the features: Contextual, Past-Behavior, and Self-Reported.

Fig. 8(a) and Fig. 8(b) summarize the results. They show that the DNNs can predict exposure with good accuracy. E.g., if the aim is to predict exposure at any time before it happens, it is possible to achieve 87% TPR with only 20% FPR. If lower FPR is desired, it is possible to achieve a 75% TPR with 10% FPR, or even 32% TPR with 1% FPR. Comparable accuracy can be achieved even when aiming to predict exposure much in advance. E.g., we can obtain a 74% TPR with 20% FPR balance even when predicting exposure 30 seconds before it actually happens. As we discuss in the following section, we believe that such early detection opens a window of opportunity for a diverse set of interventions that can prevent exposure to malicious content.

We next evaluated the influence of the different types of features on the performance of the DNNs. In particular, we evaluated the performance of the DNNs at predicting exposure when combining Contextual Features with both Past-Behavior Features and Self-Reported Features, only Past-Behavior Features, only Self-Reported Features, or none. The results are shown in Fig. 8(c). The accuracy of predictions is roughly similar in all cases—Contextual Features slightly benefit from Past-Behavior Features, but the interaction with Self-Reported Features turns out to be slightly detrimental. Shortly stated, Contextual Features by themselves are sufficient to accurately predict exposure over the short-term. Hence, accurate within-session predictive engines may be developed without the need to collect and store historical behavior information or self-reported user input (which can be expensive and time consuming), but rather via simply using available contextual information describing users' browsing activity.

False Alarms and Baseline Rate. There is a pronounced imbalance between exposed and unexposed sessions. For example, consider the operating point TPR=56% and FPR=3% in Fig. 8(a),



(a)

(b)

(c)

Figure 8: ROC curves for short-term, within-session predictions. The curves are averaged between the five evaluation periods. (a) Performance at different times before exposure (all features are used). (b) Is a zoomed-in version of (a). In (a) and (b), the dots correspond to example operating points under the more inclusive ground truth (considering pages *eventually* flagged as malicious), over the five timing horizons considered. (c) Performance when using different types of feature combinations (*C*: Contextual Features; *P*: Past-Behavior Features, and *S*: Self-Reported Features).

which can be achieved when using all features, and predicting exposure at any time before it happens. While the performance is significantly better than a random predictor (where FPR=3% yields TPR=3%), the low baseline exposure rate is a problem. The ratio between exposed and unexposed sessions is roughly 0.1:99.9 (see Table 2). At our chosen operating point, the system would make 56 true detections (out of 100) and ~3,000 false detections for every 100,000 sessions. In other words, our system would primarily issue false alarms, which could majorly impede deployment of active defenses—for instance, terminating risky sessions could make the network unusable, and issuing warnings would likely result in users ignoring them.

Fortunately, there is evidence that many of these suspected false positives are actually true positives. Indeed, we found that many of the URLs that were not flagged by GSB during the time of data collection were *eventually* (i.e., weeks or months later) flagged as potentially malicious by either GSB or other lists that VirusTotal [15] uses. More precisely, checking these services roughly one year after our field measurements, we discovered ~71% of the 3,000 "false" positives observed at the TPR=56%, FPR=3% operating point were eventually flagged as malicious, and thus were probably true positives.[1] Hence, the system may actually be achieving roughly 2,186 true positives and 870 false positives for every 100,000 sessions. (This also hints that our system performs much better than relying on blacklists with respect to false negatives.) In contrast, only 0.13% of the true negatives (i.e., ~130 sessions per 100,000) may actually be false negatives. Overall, this corresponds to TPR=93% and FPR=1% with a ratio of roughly 2.4:97.6 between exposed and unexposed sessions. Figures 8(a) and 8(b) show several operating points under the more inclusive ground truth—they all hover around 90% TPR for 1% FPR.

On a machine equipped with a Xeon X5875 CPU (3.07GHz), and 128GB of memory, our system can classify ~1,300 feature vectors per second. Thus, assuming that one webpage visit usually accounts for five HTTP requests on average (as we found), then our system can serve about 260 page visits per second. We believe this can be further optimized (e.g., via GPU computation).

8 DISCUSSION

We now discuss the implications of our findings in terms of possible interventions that can be enabled, the utility of self-reported data for designing interventions, and several extensions to our work.

Reproducibility is a thorny issue in our case. We cannot release measurements for user-privacy reasons, but hope to help others reimplement our system and reproduce our results by open sourcing the code used for computing features, training the neural networks, and evaluating them.[2]

Possible Interventions. Our proposed short-term prediction approach may enable human-centered defenses. For instance, users about to be exposed may be alerted (e.g., via nudging [3], or warnings [26]) about possible risks much before the exposure. Giving users sufficient time to consider their actions may improve their decision making [55]. In certain networks (e.g., government/defense), exposure to malicious content may be intolerable. Operators of such networks may terminate the browsing sessions of users about to visit malicious pages.

Deploying firewalls or intrusion detection systems (e.g., [60]) at the scale of a nationwide cellular network may be hard due to increased expenses and high latencies that might affect users' experience. A session-based exposure-prediction system like ours can help prioritize traffic that should pass through expensive inspection, thus enabling elastic defenses (e.g., [25]). Using deep processing, it may also be possible to prevent the installation of third-party apps in risky contexts. Additionally, the session-based prediction system may be used to collect potentially malicious pages for scanning to update blacklists, as it often predicts exposure to malicious pages before the pages appear on blacklists (as shown in Sec. 7).

We note that in some cases the FPR may be too high for effectively deploying the system in production. Indeed, when usability is the main concern (e.g., in a general purpose network), certain interventions, such as terminating connections or warning users, may lead to annoyance or habituation [84] when the FPR is high. In such cases, we suggest to tune the system to decrease the FPR (thus also decreasing the TPR). Alternately, one may resort to more conservative interventions, such as identifying potentially malicious pages for scanning or preventing the installation of 3rd-party apps.

[1] Similarly to prior work [5], we conservatively considered a URL to be malicious only if it was flagged by two or more of the lists that are used by VirusTotal.

[2] https://github.com/mahmoods01/exposure-prediction

We expect that future enhancements of the system would make it effective in a wider range of settings.

Similarly to short-term prediction, long-term prediction can enable certain types of human-centered interventions, such as increased training and education (e.g., [44]) for users predicted to engage is long-term risky behavior.

Utility of Self-Reported Data. We developed a better understanding of the utility of self-reported data for explaining and predicting security behavior and incidents thus contributing to a growing body of literature [21, 22, 86]. For example, users who reported using anti-virus are much more likely than others to get exposed, echoing previous findings [14]. While self-reported data helped forecast long-term exposure risk, sole reliance on self-reported data resulted in only moderately accurate predictions. To build accurate predictive models based on self-reported data, one probably needs to complement self-reported data with behavioral measurements of metrics similar to those described in Sec. 4.

Potential Extensions and Limitations. Naturally, improving the performance of our session-based apparatus would improve its practicality. Using additional, computationally more expensive, features (e.g., domain-name reputation [4], or redirection-graph features [78]) that were shown to be useful for detecting malicious domains may improve the performance. Machine-learning algorithms other than the ones we explored in this work, such as recurrent neural networks which achieve state-of-the-art results on learning from sequences (e.g., in machine translation [6]), may also help.

Our session-based prediction approach could be extended to networks other than cellular networks (e.g., residential or enterprise networks). Such networks introduce new challenges (e.g., due to different user behavior or higher device diversity), but similar prediction performance should be achievable.

While users should not have strong incentives to evade our predictive models, attackers may want to "poison" the training data [38]. Such attackers however need the ability to interact at the scale of the (entire) exposed population of the service provider, and thus to control a large number of devices on the provider network. On smaller networks, this is an important issue to address.

To trust the predictive engine, users and network operators may want to know why a session or a user are predicted to be exposed; recent research on explaining machine-learning models [20, 66] might be helpful.

Last, privacy concerns mandate to minimize the amount of user data required for training the predictive models. Our (strong) subsampling of HTTP requests from unexposed sessions helps, but recent techniques might reduce user data collection even further [46].

9 CONCLUSION

We developed a system to predict whether users will be exposed to malicious pages while browsing the web, and evaluated it using three-months worth of HTTP traffic generated, in 2017, by >20,000 users of a Japanese cellular provider. We found that the system can quite accurately predict exposure seconds before it occurs, thus potentially enabling several proactive defenses. Our measurements further motivated the need for such a system, as we discovered blacklists such as Google Safe Browsing were, in some instances, noticeably lagging (days or even weeks) behind malicious page accesses. In fact, our system manages to detect malicious pages *before* they are included in blacklists, and is thus a good complement to reactive defenses. Additionally, we collected self-reported demographic and security-related data from the same users and evaluated their utility at predicting exposure. We found that models that solely rely on self-reported data may help forecast exposure to malicious pages over long time periods, but are also remarkably less accurate than those including behavioral data.

10 ACKNOWLEDGMENTS

We thank our anonymous reviewers and our shepherd, Tudor Dumitraş, for feedback that greatly improved the manuscript. Mahmood Sharif was partially supported through MURI grant W911NF-17-1-0370, and by CyLab at Carnegie Mellon University via a CyLab Presidential Fellowship.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *Proc. OSDI.*

[2] Maarten Aertsen, Maciej Korczyński, Giovane Moura, Samaneh Tajalizadehkhoob, and Jan van den Berg. 2017. No domain left behind: is Let's Encrypt democratizing encryption?. In *Proc. ANRW.*

[3] Hazim Almuhimedi, Florian Schaub, Norman Sadeh, Idris Adjerid, Alessandro Acquisti, Joshua Gluck, Lorrie Faith Cranor, and Yuvraj Agarwal. 2015. Your location has been shared 5,398 times!: A field study on mobile app privacy nudging. In *Proc. CHI.*

[4] Manos Antonakakis, Roberto Perdisci, David Dagon, Wenke Lee, and Nick Feamster. 2010. Building a Dynamic Reputation System for DNS. In *Proc. USENIX Security.*

[5] Daniel Arp, Michael Spreitzenbarth, Malte Hubner, Hugo Gascon, Konrad Rieck, and CERT Siemens. 2014. DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket. In *Proc. NDSS.*

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. ICLR.*

[7] Rodrigo Benenson. 2017. What is the class of this image? Discover the current state of the art in objects classification. https://goo.gl/3wucZd. (2017). Online; accessed 2 May 2018.

[8] Leyla Bilge and Tudor Dumitras. 2012. Before We Knew It: An Empirical Study of Zero-day Attacks in the Real World. In *Proc. CCS.*

[9] Leyla Bilge, Yufei Han, and Matteo Dell'Amico. 2017. RiskTeller: Predicting the Risk of Cyber Incidents. In *Proc. CCS.*

[10] Davide Canali, Leyla Bilge, and Davide Balzarotti. 2014. On the effectiveness of risk prediction based on users browsing behavior. In *Proc. AsiaCCS.*

[11] Casey Canfield, Alex Davis, Baruch Fischhoff, Alain Forget, Sarah Pearman, and Jeremy Thomas. 2017. Replication: Challenges in Using Data Logs to Validate Phishing Detection Ability Metrics. In *Proc. SOUPS.*

[12] Yannick Carlinet, Ludovic Mé, Hervé Debar, and Yvon Gourhant. 2008. Analysis of computer infection risk factors based on customer network usage. In *Proc. SECURWARE.*

[13] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[14] Nicolas Christin, Serge Egelman, Timothy Vidas, and Jens Grossklags. 2011. It's all about the Benjamins: An empirical study on incentivizing users to ignore security advice. In *Proc. FC.*

[15] Chronicle. 2004–. VirusTotal. https://www.virustotal.com/. (2004–). Online; accessed 8 May 2018.

[16] Christopher Clark and Amos Storkey. 2015. Training Deep Convolutional Neural Networks to Play Go. In *Proc. ICML.*

[17] Adam Coates, Andrew Ng, and Honglak Lee. 2011. An analysis of single-layer networks in unsupervised feature learning. In *Proc. AISTATS.*

[18] Selenium contributors. 2004–. SeleniumHQ Browser Automation. http://www.seleniumhq.org/. (2004–). Online; accessed 2 May 2018.

[19] Aldo Cortesi, Maximilian Hils, Thomas Kriechbaumer, and contributors. 2010–. mitmproxy: A free and open source interactive HTTPS proxy. (2010–). https://mitmproxy.org/ Online; accessed 2 May 2018.

[20] Anupam Datta, Shayak Sen, and Yair Zick. 2016. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *Proc. IEEE S&P*.

[21] Serge Egelman, Marian Harbach, and Eyal Peer. 2016. Behavior Ever Follows Intention?: A Validation of the Security Behavior Intentions Scale (SeBIS). In *Proc. CHI*.

[22] Serge Egelman and Eyal Peer. 2015. The myth of the average user: Improving privacy and security systems through individualization. In *Proc. NSPW*.

[23] Serge Egelman and Eyal Peer. 2015. Predicting privacy and security attitudes. *ACM SIGCAS Computers and Society* 45, 1 (2015), 22–28.

[24] Serge Egelman and Eyal Peer. 2015. Scaling the Security Wall: Developing a Security Behavior Intentions Scale (SeBIS). In *Proc. CHI*.

[25] Seyed Kaveh Fayaz, Yoshiaki Tobioka, Vyas Sekar, and Michael Bailey. 2015. Bohatei: Flexible and Elastic DDoS Defense. In *Proc. USENIX Security*.

[26] Adrienne Porter Felt, Alex Ainslie, Robert W Reeder, Sunny Consolvo, Somas Thyagaraja, Alan Bettes, Helen Harris, and Jeff Grimes. 2015. Improving SSL warnings: Comprehension and adherence. In *Proc. CHI*.

[27] Adrienne Porter Felt, Matthew Finifter, Erika Chin, Steve Hanna, and David Wagner. 2011. A survey of mobile malware in the wild. In *Proc. SPSM Workshop*.

[28] Alain Forget, Sarah Pearman, Jeremy Thomas, Alessandro Acquisti, Nicolas Christin, Lorrie Faith Cranor, Serge Egelman, Marian Harbach, and Rahul Telang. 2016. Do or do not, there is no try: user engagement may not improve security outcomes. In *Proc. SOUPS*.

[29] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proc. AISTATS*.

[30] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[31] Google. 2008–. Google Safe Browsing. https://safebrowsing.google.com/. (2008–). Online; accessed 5 August 2018.

[32] Guofei Gu, Roberto Perdisci, Junjie Zhang, and Wenke Lee. 2008. BotMiner: Clustering Analysis of Network Traffic for Protocol-and Structure-Independent Botnet Detection. In *Proc. USENIX Security*.

[33] Guofei Gu, Junjie Zhang, and Wenke Lee. 2008. BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic. In *Proc. NDSS*.

[34] Shuang Hao, Alex Kantchelian, Brad Miller, Vern Paxson, and Nick Feamster. 2016. PREDATOR: proactive recognition and elimination of domain abuse at time-of-registration. In *Proc. CCS*.

[35] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *Proc. IJCNN*.

[36] Haibo He and Edwardo A Garcia. 2009. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering* 21, 9 (2009), 1263–1284.

[37] Tin Kam Ho. 1995. Random decision forests. In *ICDAR*.

[38] Ling Huang, Anthony D Joseph, Blaine Nelson, Benjamin IP Rubinstein, and JD Tygar. 2011. Adversarial machine learning. In *Proc. AISec*.

[39] Digital Arts Inc. 2009–. i-FILTER (Consumer). http://www.daj.jp/en/products/if_consumers/. (2009–). Online; accessed 2 May 2018.

[40] Luca Invernizzi, Stanislav Miskovic, Ruben Torres, Christopher Kruegel, Sabyasachi Saha, Giovanni Vigna, Sung-Ju Lee, and Marco Mellia. 2014. Nazca: Detecting Malware Distribution in Large-Scale Networks. In *Proc. NDSS*.

[41] Jonathan Taylor Josef Perktold, Skipper Seabold. 2012–. statsmodels. http://www.statsmodels.org/. (2012–). Online; accessed 2 May 2018.

[42] Chanhyun Kang, Noseong Park, B Aditya Prakash, Edoardo Serra, and VS Subrahmanian. 2016. Ensemble models for data-driven prediction of malware infections. In *Proc. WSDM*.

[43] Keras. 2017. Keras: The Python Deep Learning library. https://keras.io. (2017). Online; accessed 2 May 2018.

[44] Ponnurangam Kumaraguru, Steve Sheng, Alessandro Acquisti, Lorrie Faith Cranor, and Jason Hong. 2010. Teaching Johnny not to fall for phish. *ACM Transactions on Internet Technology (TOIT)* 10, 2 (2010), 7.

[45] Fanny Lalonde Levesque, Jude Nsiempba, José M Fernandez, Sonia Chiasson, and Anil Somayaji. 2013. A clinical study of risk factors related to malware infections. In *Proc. CCS*.

[46] Mathias Lecuyer, Riley Spahn, Roxana Geambasu, Tzu-Kuo Huang, and Siddhartha Sen. 2017. Pyramid: Enhancing Selectivity in Big Data Protection with Count Featurization. In *Proc. IEEE S&P*.

[47] Martin Lee. 2012. Who's next? Identifying risks factors for subjects of targeted attacks. In *Proc. Virus Bull. Conf.*

[48] Nektarios Leontiadis and Nicolas Christin. 2014. Empirically measuring WHOIS misuse. In *Proc. ESORICS*.

[49] Charles Lever, Manos Antonakakis, Bradley Reaves, Patrick Traynor, and Wenke Lee. 2013. The Core of the Matter: Analyzing Malicious Traffic in Cellular Carriers. In *Proc. NDSS*.

[50] Fanny Lalonde Lévesque, José M Fernandez, and Anil Somayaji. 2014. Risk prediction of malware victimization based on user behavior. In *Proc. MALWARE*.

[51] Yang Liu, Armin Sarabi, Jing Zhang, Parinaz Naghizadeh, Manish Karir, Michael Bailey, and Mingyan Liu. 2015. Cloudy with a Chance of Breach: Forecasting Cyber Security Incidents. In *Proc. USENIX Security*.

[52] Gregor Maier, Anja Feldmann, Vern Paxson, Robin Sommer, and Matthias Vallentin. 2011. An assessment of overt malicious activity manifest in residential networks. In *Proc. DIMVA*.

[53] K.L. Manfreda, M. Bosnjak, J. Berzelak, I. Haas, and V. Vehovar. 2008. Web Surveys versus Other Survey Modes: A Meta-Analysis Comparing Response Rates. *International Journal of Market Research* 50 (2008), 79–104.

[54] Michelle L Mazurek, Saranga Komanduri, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Patrick Gage Kelley, Richard Shay, and Blase Ur. 2013. Measuring password guessability for an entire university. In *Proc. CCS*.

[55] Katherine L Milkman, Dolly Chugh, and Max H Bazerman. 2009. How can decision making be improved? *Perspectives on psychological science* 4, 4 (2009), 379–383.

[56] Terry Nelms, Roberto Perdisci, and Mustaque Ahamad. 2013. ExecScent: Mining for New C&C Domains in Live Networks with Adaptive Control Protocol Templates. In *Proc. USENIX Security*.

[57] Terry Nelms, Roberto Perdisci, Manos Antonakakis, and Mustaque Ahamad. 2015. WebWitness: Investigating, Categorizing, and Mitigating Malware Download Paths. In *Proc. USENIX Security*.

[58] Kaan Onarlioglu, Utku Ozan Yilmaz, Engin Kirda, and Davide Balzarotti. 2012. Insights into User Behavior in Dealing with Internet Attacks. In *Proc. NDSS*.

[59] Michael Ovelgönne, Tudor Dumitras, B Aditya Prakash, VS Subrahmanian, and Benjamin Wang. 2017. Understanding the Relationship between Human Behavior and Susceptibility to Cyber Attacks: A Data-Driven Approach. *ACM Transactions on Intelligent Systems and Technology (TIST)* 8, 4 (2017), 51.

[60] Vern Paxson. 1999. Bro: a system for detecting network intruders in real-time. *Computer networks* 31, 23 (1999), 2435–2463.

[61] M Zubair Rafique, Tom Van Goethem, Wouter Joosen, Christophe Huygens, and Nick Nikiforakis. 2016. It's free for a reason: Exploring the ecosystem of free live streaming services. In *Proc. NDSS*.

[62] Arun Raghuramu, Parth H Pathak, Hui Zang, Jinyoung Han, Chang Liu, and Chen-Nee Chuah. 2016. Uncovering the footprints of malicious traffic in wireless/mobile networks. *Computer Communications* 95 (2016), 95–107.

[63] Moheeb Abu Rajab, Lucas Ballard, Noé Lutz, Panayiotis Mavrommatis, and Niels Provos. 2013. CAMP: Content-Agnostic Malware Protection. In *Proc. NDSS*.

[64] Elissa M Redmiles, Sean Kross, and Michelle L Mazurek. 2017. Where is the Digital Divide?: A Survey of Security, Privacy, and Socioeconomics. In *Proc. CHI*.

[65] Elissa M Redmiles, Amelia R Malone, and Michelle L Mazurek. 2016. I Think They're Trying to Tell Me Something: Advice Sources and Selection for Digital Security. In *Proc. IEEE S&P*.

[66] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should I trust you?: Explaining the predictions of any classifier. In *Proc. KDD*.

[67] Konrad Rieck, Tammo Krueger, and Andreas Dewald. 2010. Cujo: efficient detection and prevention of drive-by-download attacks. In *Proc. ACSAC*.

[68] Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom Van Goethem, and Wouter Joosen. 2018. Automated Website Fingerprinting through Deep Learning. In *Proc. NDSS*.

[69] Carl Sabottke, Octavian Suciu, and Tudor Dumitras. 2015. Vulnerability Disclosure in the Age of Social Media: Exploiting Twitter for Predicting Real-World Exploits. In *Proc. USENIX Security*.

[70] George Saon, Hong-Kwang J Kuo, Steven Rennie, and Michael Picheny. 2016. The IBM 2015 English conversational telephone speech recognition system. *arXiv preprint arXiv:1505.05899* (2016).

[71] Armin Sarabi, Parinaz Naghizadeh, Yang Liu, and Mingyan Liu. 2016. Risky business: Fine-grained data breach prediction using business profiles. *Journal of Cybersecurity* 2, 1 (2016), 15–28.

[72] Armin Sarabi, Ziyun Zhu, Chaowei Xiao, Mingyan Liu, and Tudor Dumitraş. 2017. Patch Me If You Can: A Study on the Effects of Individual User Behavior on the End-Host Vulnerability State. In *International Conference on Passive and Active Network Measurement*. Springer, 113–125.

[73] Yukiko Sawaya, Mahmood Sharif, Nicolas Christin, Ayumu Kubota, Akhiro Nakarai, and Akira Yamada. 2017. Self-Confidence Trumps Knowledge: A Cross-Cultural Study of Security Behavior. In *Proc. CHI*.

[74] Howard J Seltman. 2012. *Experimental design and analysis*. Pittsburgh: Carnegie Mellon University.

[75] Steve Sheng, Mandy Holbrook, Ponnurangam Kumaraguru, Lorrie Faith Cranor, and Julie Downs. 2010. Who falls for phish?: a demographic analysis of phishing susceptibility and effectiveness of interventions. In *Proc. CHI*.

[76] Kyle Soska and Nicolas Christin. 2014. Automatically Detecting Vulnerable Websites Before They Turn Malicious. In *Proc. Usenix Security*.

[77] StatCounter. 2017. Browser market share Worldwide (July 2017). https://goo.gl/7KgmuK. (2017). Online; accessed 2 May 2018.

[78] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. 2013. Shady paths: Leveraging surfing crowds to detect malicious web pages. In *Proc. CCS*.

[79] Taosoftware. 2012–. tPacketCapture. https://goo.gl/d9AcQB. (2012–). Online; accessed 2 May 2018.

[80] The Verge. 2017. 99.6 percent of new smartphones run Android or iOS. https://goo.gl/yMCGpW. (2017). Online; accessed 2 May 2018.

[81] Olivier Thonnard, Leyla Bilge, Anand Kashyap, and Martin Lee. 2015. Are you at risk? Profiling organizations and individuals subject to targeted attacks. In *Proc. FC*.

[82] Hien Thi Thu Truong, Eemil Lagerspetz, Petteri Nurmi, Adam J Oliner, Sasu Tarkoma, N Asokan, and Sourav Bhattacharya. 2014. The company you keep: Mobile malware infection rates and inexpensive risk indicators. In *Proc. WWW*.

[83] Phani Vadrevu, Babak Rahbarinia, Roberto Perdisci, Kang Li, and Manos Antonakakis. 2013. Measuring and detecting malware downloads in live network traffic. In *Proc. ESORICS*.

[84] Anthony Vance, Brock Kirwan, Daniel Bjorn, Jeffrey Jenkins, and Bonnie Brinton Anderson. 2017. What Do We Really Know About How Habituation to Warnings Occurs Over Time?: A Longitudinal fMRI Study of Habituation and Polymorphic Warnings. In *Proc. CHI*.

[85] Gang Wang, Tristan Konolige, Christo Wilson, Xiao Wang, Haitao Zheng, and Ben Y. Zhao. 2013. You Are How You Click: Clickstream Analysis for Sybil Detection. In *Proc. USENIX Security*.

[86] Rick Wash, Emilee Rader, and Chris Fennell. 2017. Can People Self-Report Security Accurately?: Agreement Between Self-Report and Behavioral Measures. In *Proc. CHI*.

[87] Gilbert Wondracek, Thorsten Holz, Christian Platzer, Engin Kirda, and Christopher Kruegel. 2010. Is the Internet for Porn? An Insight Into the Online Adult Industry. In *Proc. WEIS*.

[88] Guowu Xie, Marios Iliofotou, Thomas Karagiannis, Michalis Faloutsos, and Yaohui Jin. 2013. Resurf: Reconstructing web-surfing activity from network traffic. In *Proc. IFIP Networking*.

[89] Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig. 2016. Achieving human parity in conversational speech recognition. *arXiv preprint arXiv:1610.05256* (2016).

[90] Ting-Fang Yen, Victor Heorhiadi, Alina Oprea, Michael K Reiter, and Ari Juels. 2014. An epidemiological study of malware encounters in a large enterprise. In *Proc. CCS*.

[91] Junjie Zhang, Christian Seifert, Jack W Stokes, and Wenke Lee. 2011. Arrow: Generating signatures to detect drive-by downloads. In *Proc. WWW*.

[92] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proc. NIPS*.

[93] Yajin Zhou and Xuxian Jiang. 2012. Dissecting android malware: Characterization and evolution. In *Proc. IEEE S&P*.

[94] Yajin Zhou, Zhi Wang, Wu Zhou, and Xuxian Jiang. 2012. Hey, you, get off of my market: detecting malicious apps in official and alternative android markets. In *Proc. NDSS*.

A USER SURVEY

Below are the questions that were asked of our 23,419 survey participants (beyond demographic questions, and consent). These questions are English translations of the original Japanese questionnaire.

(1) Have you ever experienced the following attacks or events while using a mobile device?
  (a) Virus infection
  (b) Anti-virus alert
  (c) Private information theft
  (d) Compromised user ID, password, location, or picture
  (e) Forced subscription to unwanted paid service
  (f) Browser alert
  (g) Other issues
  Possible answers for each sub-question:
  (a) *No*
  (b) *I don't know*
  (c) *Within the past 3 months*
  (d) *Within the past year*
  (e) *Over a year ago*

(2) Have you installed an anti-virus on your smartphone?
  (a) Yes
  (b) No
  (c) I don't know

(3) Which market are you using for app download?
  (a) Google Play, Apple Store
  (b) au Market (KDDI's affiliate)
  (c) Other website
  Possible answers for each sub-question:
  (a) *Never*
  (b) *Sometimes*
  (c) *Often*
  (d) *Always*

(4) How do you respond to a warning page in your browser?
  (a) I proceed to the next page without further thinking
  (b) I proceed to the next page if the webpage is important to me
  (c) I proceed to the next page if the webpage is popular/famous enough
  (d) I do not proceed if the browser shows a warning
  (e) Other
  (f) I have never seen a warning page

The next group of questions (borrowed from RSeBIS, [73]) takes, as answers, a 5-point Likert scale ranging from "Completely disagree" to "Perfectly agree."

(1) When someone sends me a link, I open it only after verifying where it goes.
(2) I know what website I'm visiting by looking at the URL bar, rather than by the website's look and feel.
(3) I verify that information will be sent securely (e.g., SSL, "https://," a lock icon) before I submit it to websites.
(4) When browsing websites, I mouse-over links to see where they go, before clicking them.
(5) If I discover a security problem, I fix or report it rather than assuming somebody else will.

Finally, the last group of questions on self-confidence also takes, as answers, a 5-point Likert scale ranging from "Completely disagree" to "Perfectly agree."

(1) I know about countermeasures for keeping the data on my device from being exploited.
(2) I know about countermeasures to protect myself from monetary loss when using the Internet.
(3) I know about countermeasures to prevent my IDs or Passwords being stolen.
(4) I know about countermeasures to prevent my devices from being compromised.
(5) I know about countermeasures to protect me from being deceived by fake websites.
(6) I know about countermeasures to prevent my data from being stolen during web browsing.