Stranger Danger: Exploring the Ecosystem of Ad-based URL Shortening Services

Nick Nikiforakis[†], Federico Maggi[‡], Gianluca Stringhini[*], M. Zubair Rafique[†], Wouter Joosen[†], Christopher Kruegel[*], Frank Piessens[†], Giovanni Vigna[*], Stefano Zanero[‡]

[†]iMinds-DistriNet, KU Leuven, 3001 Leuven, Belgium
firstname.lastname@cs.kuleuven.be

[‡] Politecnico di Milano
{fmaggi, zanero}@elet.polimi.it

[*]UC Santa Barbara
{gianluca,chris,vigna}@cs.ucsb.edu

ABSTRACT

URL shortening services facilitate the need of exchanging long URLs using limited space, by creating compact URL aliases that redirect users to the original URLs when followed. Some of these services show advertisements (ads) to link-clicking users and pay a commission of their advertising earnings to link-shortening users.

In this paper, we investigate the ecosystem of these increasingly popular ad-based URL shortening services. Even though traditional URL shortening services have been thoroughly investigated in previous research, we argue that, due to the monetary incentives and the presence of third-party advertising networks, ad-based URL shortening services and their users are exposed to more hazards than traditional shortening services. By analyzing the services themselves, the advertisers involved, and their users, we uncover a series of issues that are actively exploited by malicious advertisers and endanger the users. Moreover, next to documenting the ongoing abuse, we suggest a series of defense mechanisms that services and users can adopt to protect themselves.

Categories and Subject Descriptors

K.6.5 [**Security and Protection**]: Unauthorized access; H.3.5 [**Online Information Services**]: Web-based services; K.4.4 [**Electronic Commerce**]: Security

Keywords

Short URLs; advertising; malware; iframe; HTML5

1. INTRODUCTION

Many of the most popular sites on the modern web rely heavily on user-generated content. For instance, online social networks depend on their users for sharing links and status updates with their contacts, while news-aggregation sites expect users to submit stories and rate the content submitted by other users. At the same time, however, some services, such as Twitter, have chosen to restrict the verbosity of users by limiting the total number of characters in the text fields of their generated content.

URL shortening services arose to address the problem of publishing long URLs when the available space is limited. A URL shortening service creates compact alias URLs for longer URLs that, when visited, redirect the user to the original long URLs. While, initially, URL shortening services were used primarily for their shortening functionality, nowadays users utilize them even when there are no space limitations, as a way of "beautifying" their links and tracking user clicks [2]. Unfortunately, attackers found URL shortening services equally useful and started shortening links towards malicious pages. By spreading the generated short URLs instead of their original ones, attackers could evade blacklists and filtering systems looking for suspicious patterns in URLs, or simply exploit the fact that users consider short URLs as a (benign) part of the web ecosystem.

One special type of URL shortening services are ad-based URL shortening services, like `adf.ly`. These services, in addition to shortening long URLs, pay the link-shortening users a small commission every time a user clicks on their shortened link. The services generate income by exposing the visitor of short URLs to advertisements before redirecting them to the final destination. Although some of the previous work on generic URL shorteners included some ad-based shorteners in their list of studied services [21], the ad-based URL shortening services were treated in the exact same way as the rest of the traditional URL shorteners, ignoring their peculiarities.

In this paper, we argue that (1) the monetary incentive for link-shortening users and (2) the involvement of third-party advertising networks expose ad-based URL shortening services and their users to many more security and privacy threats than their non-advertising counterparts. To discover and quantify these threats, we perform a three-pronged analysis of ad-based URL shortening services.

First, we compile a list of the top ten ad-based URL shortening services and analyze their technical characteristics and architectural choices, focusing on the ones with security and privacy consequences. We discover that, among others, none

WWW'14, April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2744-2/14/04.
http://dx.doi.org/10.1145/2566486.2567983.

of the services properly isolate the advertising content, meaning that malicious advertisers can escape from their containers, take control of the page, and redirect the user to malicious pages of their choice. Second, we study the advertisers that choose ad-based URL shortening services as their advertising platform. We identify advertisers that perform drive-by downloads, attempt to trick the user into installing malware, escape their isolation containers, and expose minors to inappropriate content. Third, we turn our attention to the users of these services, separating them into producers and consumers, i.e., users who create short links and users who click on them. For producers, we analyze 29,709 short URLs and discover many abuses, including the chaining of multiple ad-based URL shortening services to gain multiple commissions from a single click, and luring the user into clicking on links shortened by ad-based URL shortening services, promising them access to attractive content, but finally not delivering it. To study the consumers, we buy advertising packages from the two most popular services and show that more than 10% of all visitors who were exposed to our ads were running vulnerable and exploitable browsers and browser plugins. Finally, motivated by the magnitude of the discovered abuse, we propose a series of pragmatic defense mechanisms that ad-based URL shortening services and their users can readily adopt.

In summary, we make the following original contributions:

- We present the first study of ad-based URL shortening services, highlighting their increased attack surface over traditional URL shortening services.
- We collect more than 10,000 ads shown to users of these services and describe the discovered attacks and abuses.
- We analyze the users of these services and show that, in principle, cybercriminals could use ad-based URL shortening services as a platform for drive-by downloads.
- We propose the use of modern, existing HTML5 tags and other mechanisms for strengthening the services and their users against attacks.

2. BACKGROUND

In this section, after a brief overview of traditional URL shortening services, we describe how ad-based URL shortening services work and how they differ from traditional ones. More precisely, we highlight how they relay the user from the short URL to the final destination URL, and the monetary incentives that could make their users act differently.

2.1 URL shortening services

The first memorable URL shortening service was TinyURL, which was launched in 2002. Its success attracted competitors and today, there are hundreds of different URL shortening services that occasionally offer extra features, as a way of differentiating themselves from the rest.

As the name of these services suggests, the primary purpose of a URL shortening service is to shorten a long URL. For instance, the link to the PDF file of the Call for Research Papers of the WWW 2014 conference[1] is 74 characters long. When using `bit.ly`, one of the currently most popular URL shortening services, to shorten this long URL, we obtained the link `http://bit.ly/1bdXeib` which is 21 characters long, i.e., almost four times shorter than the original link. In almost all cases, creating short URLs can be achieved either by visiting the websites of the shortening services and submitting the desired destination URL through their web interface or, in some cases, programmatically through the use of APIs.

When a user visits a short URL, her browser is automatically redirected to the destination page, usually through the use of appropriate HTTP status messages (HTTP 301 or 302), or other client-side mechanisms, e.g., JavaScript or HTML meta tags. At the same time, the URL shortening service registers the visit and creates aggregate statistics about the visitors that clicked on each specific short URL, which are usually made available publicly or just to the creator of the short link.

2.1.1 Advantages

We highlight the following advantages that URL shortening services offer.

Length Reduction: Reducing the length of a link is desirable in certain circumstances. In some social media applications such as Twitter, users have only a limited number of characters available to type their message. Thus, reducing the size of a shared link provides users with more space for the rest of their message. In printed media, such as a business card, there are physical constraints where short URLs are favored over longer ones. Additionally, in many cases, the user has to manually type the URL off of a product, or a presentation, into her browser. Requiring the user to type less characters or using a mnemonic alias, e.g., `bit.ly/summer2013`, means that there are less chances for typing errors, which could lead to different websites and navigation errors, causing annoyance for the user.

Beautification: It is not uncommon for certain links to include a large number of parameters with special values and control characters. Thus, when sharing link such as `http://example.com/~user1/foo/index.php?a=1&b=2&c=3#secA`, a user may choose to use a shortening service to hide the presence of these parameters from the shared link. Certain services, such as Google Maps, have a built-in URL shortening functionality to ease the creation and sharing of "beautified" short links.

Analytics: Whenever users share a link, they may want to inspect whether the users who received the link, actually visited the page. In many cases, however, the destination URL is not under the control of the link-sharing user, i.e., the user does not have access to web analytics or web-server logs. In these cases, URL shortening services can be used to provide a wrapper around the destination URL, which will record the fact that the user went through the URL shortening service, and thus provide visitor analytics to the link-creating user. For instance, `bit.ly` recently released a tool that allows users to create, manage, share, and track bundles of (short) links in a way similar to browser bookmarks.

Centralized control: Some social networking and microblogging sites wrap all user-produced links with their own

[1] `http://www2014.kr/wp-content/uploads/2013/09/WWW2014_CFP_ResearchTrack.pdf`

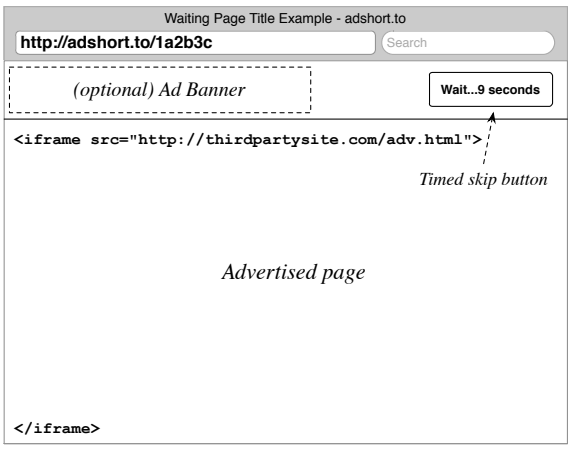

Figure 1: Typical organization of the "Waiting Page", when visiting a link shortened by an ad-based URL shortening service. Notice the large amount of space alloted to the ads.

URL shortening service, which basically acts as a proxy. This allows the service to disable any link at anytime, without the need to hunt down all posts including a specific link, and without the need to take offline the final destination. For instance, if a malicious user shares a link towards a drive-by download site, whenever Twitter detects the threat, it can simply block one single short URL and stop its users from visiting the malicious page, regardless of the connectivity status of the actual malicious page.

2.1.2 Disadvantages

Although URL shortening services are immensely popular, mainly because they are "embedded" in modern online social networks, they are also the cause of several security-, privacy-, and availability-related concerns [16, 21]:

Linkrot: Whenever users share URLs pointing to resources of interest that are outside of their control, e.g., news articles, they unconsciously trust the websites hosting these resources, not to change the content present on those URLs. For instance, if a user, Alice, shares the link of an image of an inspiring scenery with her social network friends, she assumes that when, in the future, her friends visit that URL, the image will still be available, and it will still be the same inspiring scenery. If, however, Alice used a URL shortening service and the shortening service ceases to be operational, her link will *rot*, i.e., it will no longer lead the visiting user to the proper destination, regardless if the original image is still online and available. Moreover, since most URL shortening services are free of charge, there are no strong guarantees that they will be available in the future.

Hijacking: Similar to the issue of linkrot, *hijacking* can occur if an attacker manages to change the destination of short URLs and redirect the visiting users to pages under his control. An example of such a scenario was an attack against the `cli.gs` URL shortening service in 2009, where the attackers compromised the backend servers of the service and managed to change the destination of 2.2 million short URLs. As a result, millions of users were redirected to the attackers' domains [5].

Obfuscation and Maliciousness: In the list of advantages, we mentioned that users may utilize a URL shortening service for beautification purposes. Unfortunately, the same reasoning can be abused by attackers, who use such services in order to hide the final, malicious destination of their URLs, or even to pique a user's interest by crafting a short URL with a special suffix so as to fit the victim's profile (e.g., `tinyurl.com/freemoviepack`). Thus, in the past, short URLs have been found to lead to phishing pages [4], pages hosting malicious, drive-by downloads [18], and shock sites, i.e., sites with content that is offensive and provocative for the vast majority of users.

2.2 Ad-based URL shortening services

Ad-based URL shortening services are services that use advertising and referral programs to encourage users to create and share short links by paying them a small amount of money for every visit to their short URLs. For the user who creates the short link, the process is similar to shortening a link with any other URL shortening service. The key difference is that the link-creating user must have an account with the service, if she wants to get paid for the traffic that she later brings.

2.2.1 Waiting page and redirection

Whenever another user clicks on the link shortened by an ad-based URL shortening service, she lands on the service's "Waiting Page", where she must first watch an advertisement for at least a few seconds before she is allowed to proceed to the final destination of the short URL. Most services follow the page organization presented in Figure 1, where the top-part of the page is controlled by the ad-based URL shortening service and the bottom one presents the advertised content inside an `iframe`. The timed "Continue" button becomes active and clickable only after a predetermined number of seconds. This ensures that the link-following user gets exposed to the ad before continuing to the landing page. During this time span, the landing page's URL is not revealed. Depending on the service, it could be simply obfuscated, or loaded asynchronously from the service's server by a JavaScript routine. Some services also use the top part of the page to show additional advertising banners, maximizing the screen real-estate dedicated to ads.

2.2.2 Advertised page

The `iframe` displaying the ad to the user is under the full control of the advertiser. Barring the use of modern HTML5 tags that limit the functionality available to the page inside an `iframe`, an advertiser is free to run arbitrary JavaScript code, Flash, and Java applications, set cookies on the visitor's browser, and show arbitrary content. Finally, note that the ads appearing when a user follows a short URL are unpredictable, and depend on each service's internal bidding system as well as the available ads. Thus, there is no guarantee that when two users follow the same short URL, that they will be exposed to same advertisement.

2.2.3 Business model

As is usual for ad-syndicators, ad-based URL shortening services make profit through the difference between the com-

Service	Alexa Ranking	Link Hijacking	Sequential URLs	URL Leaking
adf.ly	83	✓	✓	✗
linkbucks.com	260	✓	✗	✗
adfoc.us	1,421	✓	✓	✓
bc.vc	2,473	✓	✗	✗
ysear.ch	17,571	✓	✓	✓
coinurl.com	20,831	✓	✓	✗
reducelnk.com	79,463	✓	✓	✗
ssl.gs	99,099	✓	✗	✗
zpag.es	136,459	✓	✓	✓
adcrun.ch	263,694	✓	✗	✗

Table 1: List of the ten investigated ad-based URL shortening services and their discovered shortcomings

mission they pay to their link-creating users, and the rates they charge the advertisers. For instance, as of October 2013, `adf.ly`, one of the most popular ad-based URL shortening services, charges $5 for 1,000 ad impressions targeted to users from the US, and offers $3.94 to link-creating users who deliver 1,000 visitors originating from the US.

2.2.4 Key characteristics

Overall, one can see that ad-based URL shortening services differ in two important ways from traditional URL shortening services. First, they provide a monetary incentive to link-shortening users, not simply to create short links but to make sure that they are visited by as many users as possible. We argue that this business model can motivate an equivalent of click-fraud, where some link-shortening users may try to automate visits in order to increase their profits. Second, these services provide advertisers with a privileged position in a browser, i.e., an `iframe`, which can potentially be used to harm the users who visit these shortened links. For instance, whenever attackers desire to spread malware through a drive-by download strategy, they first need to exploit popular websites and add malicious code that will divert traffic towards their browser-exploiting page. In contrast, ad-based URL shortening services give this power directly to attackers nearly for free. In the next sections of this paper, we show how all services are vulnerable to various attacks and demonstrate that attackers are actively exploiting using them for attacking link-clicking users.

3. SECURITY AND PRIVACY PROBLEMS

To discover ad-based URL shortening services we searched for competitors of `adf.ly` on popular search engines. We manually examined each discovered website and checked if it conformed to the definition of ad-based URL shortening services presented in Section 2.2. We were able to find ten such services, which we list in Table 1 ordered by their Alexa ranking. As one can readily observe, almost all services are in the first quarter of Alexa's top 1 million websites [1], which means that millions of users visit them daily.

For each discovered service, we shortened several URLs which we later visited, recording their workings and certain architectural choices that could lead to abuse by attackers. Some of the issues that we discovered, detailed below, are both novel and specific to ad-based URL shortening services, while others are generic to URL shortening services.

3.1 Link Hijacking

As mentioned in Section 2.2, ad-based URL shortening services place advertisements in an `iframe` that spans most of the "Waiting Page" that the user encounters when clicking on a short link. The use of an `iframe` sufficiently separates the advertiser from the including page, since the advertising scripts cannot access the DOM of the parent frame due to the Same-Origin Policy (SOP) [32], a powerful security mechanism enforced by all browsers. The SOP, however, does not stop the attacker from redirecting the entire page to an arbitrary destination. This can be easily done in JavaScript by simply setting the `top.location` variable to the desired destination URL.

This technique is called "frame-busting" and has been associated with sites that tried to protect themselves against clickjacking [27], i.e., an attack based on rendering a victim page in an invisible `iframe` overlaying a malicious page, and convincing the user to interact with the malicious page. Legitimate sites would include (and still do) a simple JavaScript snippet which would detect the fact that they were "framed" and escape the `iframe`, as follows:

```
if (top != self)
   top.location = self.location;
```

In ad-based URL shortening services, however, it is the untrusted party that is framed and can perform the exact same check, escaping the `iframe` and redirecting the entire tab of the user's browser. Thus, an attacker can redirect the victim from the service's "Waiting Page", to browser-exploiting pages, scams and phishing attacks. Interestingly, attackers can utilize their full power to conduct more sophisticated phishing attacks. For instance, since, by default, a site rendered in an `iframe` has full access to JavaScript and plugins, the attacker can fingerprint the user's browser [9] and redirect only specific users to a phishing site, i.e., conduct a spear-phishing attack. Additionally, for the sites that leak the page's short URL to advertisers (described in Section 3.3), an attacker can discover to which site the user will be redirected once she clicks the shortening service's time-activated button, and can thus redirect the user to phishing pages, specific to each destination site.

Finally, because of the time that the user needs to wait before she is allowed to proceed to the landing page, ranging from 5 to 10 seconds for the studied services, it is likely that the user will switch focus to another tab, thus not witnessing the redirection to a phishing page. As argued for in the *tabnabbing* attack [8, 25], this loss of focus can increase the chances that the user will later believe that the phishing page is a legitimate one, and proceed to disclose her credentials.

Even though modern browsers include `iframe`-restricting mechanisms that allow a parent page to severely restrict the power of an attacker, unfortunately, none of the investigated services are currently utilizing them — see Table 1 — thus all are prone to the aforementioned attacks. In Section 6, we describe these mechanisms in detail and show how ad-based URL shortening services could adopt them.

3.2 Sequential URLs

In the first large-scale study of traditional URL shortening services, Antoniades et al. [2] exploited the fact that popular URL shortening services were creating short URLs in a sequential, and thus predictable way, in order to gather

short URLs for their experiments. In a similar manner, six out of the ten studied ad-based URL shortening services also generate sequential short URLs.

The danger posed by sequential short URLs is the fact that they can be discovered by others, without being explicitly handed a link. This becomes particularly problematic when these short URLs point to non-public resources, such as URLs including hidden directories of a website, or including sensitive parameters, such as a Document Identifier for Google Docs, or file identifiers used by one-click-hosting services [22].

For ad-based URL shortening services, we argue that because of the commission that link-shortening users get for clicks on their links, they will prefer to link and distribute content that will appeal to many users, rather than private content that is relevant to only a small number of users. In principle, however, the risk that some users will choose these services for sharing private URLs, not knowing that their content can be discovered by other users, is still present.

3.3 Short URL Leaking

The Referer header is an HTTP header that is automatically added by the browser to outgoing requests, and identifies the URL of the resource from which the current request originated.

In the context of ad-based URL shortening services, a service, if not appropriately designed, can leak the visited short URLs to the advertisers appearing in the service's "Waiting Page", through the browser's Referer header. Three of the ten investigated services are leaking their short URLs to advertisers, leading to security, as well as, privacy consequences for their users. In terms of security, the leaked URL can be combined with the advertiser's ability to escape their `iframe`, and lead, as described earlier, to sophisticated phishing attacks. In terms of privacy, the advertiser gains knowledge about a user's browsing habits that he did not previously possess. That is, an advertiser who desires to build a browsing profile of users, can use the leaked short URL, resolve it to its real long URL, and include that expanded URL in a database of that specific user. Thus, overtime, advertisers can build lengthy browsing profiles of the users of ad-based URL shortening services, even if the destination sites do not directly provide them with visitor data.

Note that most traditional URL shortening services are not prone to this problem, because, as described in Section 2.1, the user is immediately redirected to the destination URL, without the use of a "Waiting Page" that can host third-party content.

4. ADVERTISEMENTS

Ads are the lifeblood of ad-based URL shortening services. In this section, we investigate the ads shown to users of these services, looking for possible abuses by the advertisers of each service.

4.1 Drive-by download advertisements

Drive-by download attacks [24, 31] are a big threat in the modern web. In a drive-by download attack, miscreants try to exploit a vulnerability in the browser of the victim, or in one of its plugins and, if successful, instruct the victim's machine to download and install malicious software.

Intuitively, web advertisements are the perfect way for attackers to perform drive-by download attacks: the malicious advertisement will be displayed on many different websites, and a multitude of victims visiting such websites will be infected. For these reasons, attackers are using advertisements to perform their attacks, and researchers have developed techniques to detect and block such malicious advertisements [17, 29].

Ad-exchange	Shorteners	#Malicious URLs
popads.net	adf.ly, adfoc.us, bc.vc	377
adcash.com	adf.ly	231
adsmarket.com	adf.ly, bc.vc	188
directrev.com	adf.ly	146
adbrite.com	adf.ly, adfoc.us	125
clicksor.com	adf.ly	72
yieldads.com	adf.ly	60
doubleclick.net	adf.ly, adfoc.us	31
adnxs.com	adf.ly	27
z5x.net	adf.ly, adfoc.us	17

Table 2: Top 10 ad-exchanges serving drive-by download exploits, as logged by Wepawet, showing that many ad-exchanges serve multiple shorteners. The number of malicious URLs indicates the unique number of ad-based short URLs that pointed to a malicious web page served by that ad-exchange.

To investigate to what extent cybercriminals use the advertisements displayed by ad-based URL shorteners for their drive-by download attacks, we analyzed the historical data from the Wepawet service [7]. Wepawet visits suspicious web pages in an emulated browser, and looks for signs of maliciousness that are typical of a drive-by download.

We analyzed the logs for Wepawet for the period between January 1, 2013 and July 31, 2013. In total, Wepawet logged 892 malicious web pages that were accessed by clicking on an ad-based short URL. Malicious `adf.ly` links were responsible for 80.7% of the total URLs, while the remaining percentage was shared among the other services. Note that Wepawet is a completely passive tool. It records HTTP requests as they happen, and analyzes the JavaScript code of the pages it visits, but it never clicks on links. Since Wepawet does not click on links, the "Continue" button of a service's "Waiting Page" is not clicked, and thus the emulated browser never reaches the destination URL. Therefore, the 892 detected URLs were malicious because of their ads, and not because of the final destination URL.

Note that the number of malicious short URLs detected by Wepawet in this case is a lower bound: in fact, Wepawet can detect that a particular advertisement is malicious only when that advertisement is displayed among the many possible ones. In addition, drive-by downloads are only one among many types of malicious web pages. As discussed later in this section, some advertisements displayed by ad-based URL shorteners often ask the user to install a binary, by leveraging social engineering techniques similar to fake antivirus scams [28]. Since, however, these attacks involve actions by the user, they are not detected by Wepawet.

We then wanted to understand which ad-exchange services provided the malicious advertisements detected by Wepawet. Note that once Wepawet marks a URL as malicious, it does not provide information about which particular page along the chain of redirection served the malicious content. In addition, ad-exchanges typically undergo a complex bidding system to display the most suitable advertisement for their

Service	EU Monitor					US Monitor				
	#Ad Clusters	#Malicious clusters	% Impressions	#Adult clusters	% Impressions	#Ad Clusters	#Malicious clusters	% Impressions	#Adult clusters	% Impressions
adf.ly	76	3	4.61%	1	0.99%	139	0	0.00%	0	0.00%
linkbucks.com	43	1	89.1%	2	0.45%	51	3	90.1%	0	0.00%
adfoc.us	114	6	36.5%	0	0.00%	142	23	52.1%	0	0.00%
bc.vc	25	10	95.1%	0	0.00%	48	2	75.3%	0	0.00%
ysear.ch	117	0	0.00%	0	0.00%	135	0	0.00%	0	0.00%
coinurl.lcom	248	1	0.18%	0	0.00%	232	2	1.08%	0	0.00%
reducelnk.com	329	0	0.00%	2	34.2%	300	19	45.0%	4	15.8%
ssl.gs	1	0	0.00%	0	0.00%	1	0	0.00%	0	0.00%
zpag.es	107	0	0.00%	2	3.5%	149	3	7.16%	0	0.00%
adcrun.ch	76	0	0.00%	0	0.00%	93	0	0.00%	0	0.00%

Table 3: Advertising abuses discovered for each examined service, separated by the geographical location of our monitors

users, and often times they ask other peered advertisement networks to display the advertisement for them [29]. In this case, Wepawet logged a sequence of HTTP redirections that spans multiple ad-exchanges. For simplicity, we counted an ad-exchange that appeared at any point of the redirection chain as serving a malicious URL. The ad-exchange could be a rogue one that is purposefully sending malicious advertisements, or a legitimate one that failed to detect a particular advertisement as malicious. A summary of our results is presented in Table 2, showing that some of the detected ad-exchanges served ads to multiple ad-based URL shorteners, while others were specific to `adf.ly`.

4.2 Malicious and Adult advertisements

In this section, we investigate the presence and workings of malicious and adult advertisements appearing in each of the studied ad-based URL shortening services. Note that, in this case, we denote malicious ads as those that attempt to convince the user to perform some action that will lead to the installation of malware and a possible exfiltration of private data. Thus, these ads are malicious but distinctly different than drive-by downloads. Adult ads are ads that are obviously intended for an adult audience, with varying degrees of nudity. We chose to group malicious and adult ads together because previous research has shown that adult content is tied to increased maliciousness [34], i.e., attackers take advantage of the popularity of adult content as a way of luring users to malicious pages and then exploiting their machines, or exfiltrating their private data.

To gather advertisements, we designed a scraper based on PhantomJS [12], a headless and scriptable browser, and resolved short URLs, once every hour, for each of the ten studied services, for a period of six weeks, starting on August 1, 2013. To account for ads specific to a user's geographical location, we deployed two such monitors, one in a European country and one in the US. For every visited short URL, our scraper took a screenshot of the ad showing in each service's "Waiting Page". We opted for screenshots since the advertisers in the `iframe` were using other `iframe` tags of varying and unpredictable nesting, making the automated URL extraction of the actual advertising-page, an error-prone approach. Conversely, by capturing an image of the ad, we are capturing the essence of what a user would have seen. At the end of our six-week collecting period, we had collected the screenshots of approximately 1,000 ads for each service. These screenshots contained a large variety of ads with different image size, colors, product displays, and model pictures.

To automatically cluster the screenshots of the ads for each service, we made use of perceptual hashing [19]. Given an image, perceptual hashing generates a distinct (but not unique) hash value that can be used to compare the image with others. The key advantage of using perceptual hashing, is its hash-generating speed and robustness against image scaling, aspect-ratio variation, and small changes in color, such as those due to contrast and brightness. We use the Hamming distance between hash-values as our distance metric. If the distance between two hashes is less than an empirically calculated threshold, we cluster the screenshots together. Overall using perceptual hashing, we achieve a precision of 99.7% and recall of 97.4% (compared against manually generated ground-truth). Once screenshots were clustered, we manually examined each cluster for malicious and adult ads.

Table 3 shows the results of our clustering and labelling, separated by monitor location. The first thing one can notice is that, for both monitors, five or more services exposed users to at least one malicious ad while the user was on the services' "Waiting Page". At the same time, however, it is evident that some services are much more targeted by malicious advertisers than others. For instance, in the case of `linkbucks.com`, malicious ads were shown to users from Europe, 89.1% of the time. The majority of malicious ads, across both monitors, were trying to convince the user to install software by either pretending that her browser or plugins are out of date, or that the user needed a special type of media player (plugin or stand-alone) to stream media content. Figure 2 shows an example of such an ad that warns users about their *supposedly* out-of-date Adobe Flash plugin. Sampled executables from these ads were downloaded and submitted to VirusTotal, where they were detected, by at least one anti-virus engine, as malicious software. Additionally, in Table 3, one can see a trend of showing more malicious ads to visitors from the US (with the exception of the `bc.vc` service) and more adult ads to users from Europe. One reasonable explanation for this phenomenon is that, according to research in pay-per install services, attackers are paid significantly more money for infecting machines in the US in comparison with machines anywhere else in the world [3, 20]. As such, when it comes to users from Europe, it may be more lucrative for malicious advertisers, to show them adult ads, subscribe them to ex-

Figure 2: One of the malicious ads found in the `reducelnk.com` service.

pensive services, and sell their registration data, instead of trying to infect them with malware. It is also worthwhile to note that the adult ads were unrelated to the destination page of the short URL. For instance, for one specific service, we unfortunately witnessed adult ads when following short URLs pointing to domains such as `disney.com`, showing the danger of exposing minors to adult content.

4.3 Link Hijacking

In Section 3, we described the problem of link hijacking in the context of ad-based URL shortening services. While manually labelling ad clusters, as explained earlier, we found ads displayed through four different services that had escaped their `iframe` and had redirected the entire page to a new destination of their choice. These could be visually detected, by the lack of a "Waiting-Page"-specific banner, as described in Section 2.2. By manually analyzing these redirections, we realized that while some of them were accidental, the rest were intended.

For the accidental redirections, we saw that one specific legitimate website was escaping its `iframe` and redirecting the entire page to its proper domain. By investigating their code, we located frame-busting JavaScript, as described in Section 3.1, which was meant to protect the page from unwanted framing. The same code, triggered the redirection when the page was shown, as an ad, by an ad-based URL shortening service, where there was obviously no intention to attack it.

The rest of the `iframe` escapes were intended redirections, malicious in nature. The user was redirected to malicious pages claiming to offer the Google Chrome browser, the latest version of Java, and even the waiting pages of other ad-based URL shortening services, where the user would have to wait again and then be redirected to a destination, different than the one of the original short URL.

4.4 Summary of findings

Overall, it is evident that malicious advertisers are exploiting ad-based URL shortening services, using them to launch drive-by downloads, convince users to install malicious software, or redirect them to an altogether different page. This shows that even when the users creating short links are well-meaning individuals, who try to earn some income from other users clicking on their links, they are unintentionally exposing link-clicking users to a wide range of unpredictable attackers. At the same time, some services seem to be much better than others, in keeping malicious and adult content away from their users.

5. USERS OF THESE SERVICES

In this section, we turn our attention to the users of ad-based URL shortening services. We first investigate the consumers of short links, i.e., the users who click on URLs shortened by these services, and then the producers, i.e., the people who choose these services to shorten their links.

5.1 Consumers

To find out more about the consumers of URLs shortened by ad-based URL shortening services, we purchased advertising impressions from the two top ranking shorteners, i.e., `adf.ly` and `linkbucks.com`. More specifically, for `adf.ly`, we purchased 1,000 impressions to visitors from the US and 5,000 impressions to world-wide traffic. For `linkbucks.com`, we purchased 2,000 impressions to visitors from the UK. At the time of this writing, `adf.ly` charges $5 for 1,000 visitors from the US, while only $1 for 1,000 visitors from all around the world. `linkbucks.com` uses a bidding system, where we bid $3.3 for every 1,000 UK visitors.

Given the maliciousness that we identified in previous sections of the paper, we were mostly interested in how lucrative it would be for an attacker, to use ad-based URL shortening services as a way of infecting machines. To that extent, every time our ad was rendered by a user, we obtained a partial fingerprint of her browsing environment [9, 23], in terms of the user's IP address, the browser's User Agent and the list of plugins, through a small snippet of JavaScript. We also sent to the user a session cookie from our advertising domain, so that we could remove multiple entries from the same user, at the later processing stage. From a total of 8,000 impressions, we only collected 4,300 fingerprints. The worldwide traffic from `adf.ly` (the cheapest available traffic) was the main responsible for this effect, sending us only 28.6% of the expected number of fingerprints. We believe that this is mainly due to many crawlers, scrapers and bots with limited, or altogether missing, JavaScript support, that consumed the majority of our advertising impressions. This result, in itself, points to the possibly limited usefulness of cheap world-wide traffic as a source of human impressions.

By analyzing the 4,300 fingerprints that we received, and removing duplicate entries, we discovered that approximately 50% of the users were running some sort of outdated software, in terms of browsing client, version of Adobe Flash, or version of Java. To get a realistic approximation of the percentage of exploitable software, we compared the outdated software versions with exploits in 50 modern exploit kits listed in [10]. This list of exploits is compiled by known security analysts and research groups, by tracking various exploit packs over a period of more than 3 years. We discovered that, according to the exploits used by the exploit kit authors, approximately 25% of the users running outdated software were vulnerable to at least one exploit. For example, from 978 US users of `adf.ly` for whom we received

Reputation	Safe for children		Trustworthy	
	%Referrers	%Landing	%Referrers	%Landing
Excellent	30.78	42.09	32.78	56.21
Good	9.12	26.45	15.02	18.80
Poor	3.79	14.04	0.58	13.47
Unsatisfying	5.02	6.50	4.12	2.69
Very Poor	51.29	10.92	47.50	8.82

Table 4: A non-negligible amount of the ad-based short URLs observed were found on low-reputation referrers, and were actively pointing to inappropriate content.

their fingerprints, 424 had outdated software and 110 of those were vulnerable to exploits used by exploit kits. More specifically, the 110 users were vulnerable from one to five different vulnerabilities, exploitable by nine different exploit kits, including the popular BlackHole and CoolExploit [10].

Using estimates of pay-per-install services [3, 20], an attacker can sell 1,000 malware installations on machines in the US for $180, while paying only $50 for advertising, to receive the needed number of visitors. While an attacker has more costs than just advertising, e.g, the cost of acquiring an exploit kit, and the cost of early detection by an ad-based URL shortening service, the large gap between the production cost of 1,000 infections and the selling price of these infections, is a sign that such a scheme would be profitable.

5.2 Producers

To find out more about the producers of the ad-based URL shortening ecosystem, i.e., the link-shortening users, we analyzed 29,709 distinct short URLs. We obtained this dataset by querying Bing's Search API for the `'http://<service>/*'` string (for each of the top ten services) on a daily basis between Aug 28 and Sep 20, 2013. We then visited the referrers returned by Bing, i.e., the pages hosting short links, with a PhantomJS scraper similar to the one described in Section 4.2, and searched the computed DOM for short URLs of the studied services. We found short URLs on 3,619 distinct referrers, which means about 8 to 10 short URLs per page, on average. Last, we resolved each of the collected short URLs with another scraper, obtaining 19,563 distinct landing pages, making an aliasing ratio of about 1.52 distinct short URLs per distinct landing URL. The aliasing phenomenon was mentioned in [16, 18] as one of the typical characteristics of malicious short URLs.

In order to understand where the ad-based short URLs are typically found, and where they typically point to, we categorized the referring and landing URLs in our dataset using the WOT Reputation API[2] which provided a security-oriented website categorization. When WOT was not able to provide a precise categorization (e.g., "Other" or "Unknown"), we used Trendmicro's Site Safety Center service.[3]

5.2.1 Referrer pages

Table 4 shows that the majority of the short URLs in our measurement were found on low-reputation referrers, i.e., pages labeled as "Poor", "Unsatisfying", and "Very Poor". By examining the distribution of the content categories we found that websites belonging to the "Blogs / Web Communications" category, are the most responsible for hosting links shortened by ad-based URL shortening services (26.19%).

We analyzed the top domains, i.e., the domains belonging to Blogspot, Tumblr, WordPress, and found out that these referrers are actually aggregators of short URLs that attract visitors (and search engines) by promoting free versions of otherwise paid content (e.g., software, music, videos). Indeed, a simple analysis of the URL strings revealed that the most popular words were "download", "premium" and "rip". For instance, we discovered the practice of creating attractive user-generated content, e.g., YouTube videos, that include ad-based short URLs in the video description and comments, which promise free goods if followed.

Manual inspection of a sample of the referrers categorized as "Blogs / Web Communications" revealed that the contained short URLs actually bring the users into an endless circle of ad pages, pop-ups, and other ad-based short URLs. We calculated that 25.83% of the collected short URLs pointed back within the ad-based short URL ecosystem. By performing the same calculation for 1 million short URLs belonging to traditional shortening services, as provided to us by Maggi et al. [18], we discovered that there, the practice of chaining multiple services was much less popular, with only 6.37% of the short links pointing to other shortening services. This large difference can be explained by the fact that the chaining of multiple ad-based URL shortening services increases the prospects of multiple commissions for the link-shortening users. Contrastingly, chaining multiple traditional URL shortening services merely increases the chances of linkrot, with the only plausible benefit being the obfuscation of the original referring page.

5.2.2 Landing pages

Even though many short URLs were discovered in the referrer pages of low reputation sites, Table 4 shows that the majority of links point to high-reputation landing pages. We analyzed the top categories of the landing pages and found out that they were "Good Site" (59.44%), "Other" (13.58%), "Computers/Internet" (13.46%). The top sites are file-hosting services (e.g., `fileserve.com` and `mediafire.com`), Facebook apps, and `tny.cz` (an ad-based pastebin service, often abused to host illegally-obtained content, such as usernames and passwords). These results reveal that ad-based URL shortening services are used as a "wrapper" around other services, in order for the poster of, legal and illegal, attractive content to receive a monetary compensation for their content, even if the final service does not directly provide a referral program.

Interestingly, only 9.95% of the landing pages within file-hosting domains were actually serving content, making the content promoted in the referring page unreachable. This confirms our intuition that ad-based short URLs are actively promoted with appealing content with the purpose of driving traffic through them, yet without actually providing the promised content to link-following users. Regarding the links to Facebook apps, we found that they are instances of the so-called "cheats", which are specially-crafted URLs that allow, for example, to bypass a level or gain free goods in Facebook games.

[2] `http://www.mywot.com/wiki/API`

[3] `http://global.sitesafety.trendmicro.com/`

5.3 Summary of findings

In this section, we showed that a significant fraction of the consumers of ad-based short URLs are vulnerable to exploits due to outdated client software, and estimated that the usage of ad-based short URLs as a platform for the distribution of malware would be a profitable scheme.

Regarding the producers, in addition to the historical analysis in Section 4.1, our results confirmed that ad-based short URLs are abused, arguably more than non-ad-based short URLs, due to the formers' monetary incentives. The vast majority of referring pages are used to attract visitors and lure them into clicking on ad-based short URLs. Although the amount of low-reputation landing pages may seem low (10.92% and 8.82%), our results show that that the WOT classification is actually a very conservative estimate, and that, overall, ad-based shortening services are abused by the producers of short links.

6. DEFENSES

Given the dangers due to rogue advertisers presented in Section 4, in this section, we provide an overview of methods and techniques that ad-based URL shortening services and their users, can use to defend against the aforementioned attacks and abuses.

6.1 Link hijacking

In Section 3 we demonstrated how all investigated services are vulnerable to short URL hijacking, through the redirection of the parent window, initialized by the advertiser-controlled `iframe`. While this "write-access" of the parent window's `location` property is the current default behavior of browsers, the `iframe` markup of HTML5 supports certain security-enhancing attributes which allow the parent window to modify this behavior.

More specifically, HTML5 specifies a "sandbox" attribute that can be added as part of an `iframe` [33], e.g.,

```
<iframe sandbox src="http://www.untrustedads.com">
</iframe>
```

The "sandbox" attribute subjects the loaded page to multiple severe restrictions, including the following:

- Disabling of JavaScript
- Non-loading of plugins
- Disabling the `iframe`'s ability to navigate its parent window
- Disabling the submission of forms
- Disabling pop-ups
- Unique `iframe`-specific origin, regardless of the domain where the content is loaded from

Each of these restrictions can be lifted by the parent page, through explicit white-listing, specified in the sandbox attribute of the `iframe`. While some of these restrictions, like the disabling of JavaScript, may be too strong for much of the advertised content, there are others that should always be there, such as the disabling of the parent navigation, which solves the problem of link hijacking.

One possible way of taking full advantage of this sandboxing technology, is for ad-based URL shortening services to map these `iframe` restrictions to variable advertising rates, where the rates increase together with the risk. For instance, an advertiser who simply wants to display some static content will happily agree to a fully sandboxed `iframe`, especially if it means a reduction of his advertising rates. Moreover, the resources that the companies use to detect abuse, like anti-malware and anti-phishing scanners, can then be safely migrated to increase the thoroughness and frequency of content-checking for advertisers who require more functionality. In some cases, it may even be prudent to charge more for allowing very risky functionality, like enabling the loading of plugins whose vulnerabilities are commonly abused by exploit authors to install malware on the machines of users.

From the point of view of the users, browser extensions such as AdBlocker and NoScript, could also be used to limit the danger of these services. Note, however, that since the services' profitability depends on the display of ads, usage of such tools may be against the services' terms of use.

6.2 Short URL Leaking

The leakage of a short URL through the Referer header, as explained in Section 3, can lead to various security and privacy issues. This problem is unfortunately common among services which transfer sensitive parameters in their URLs, such as session identifiers, file identifiers, or data resulting from a user-filled form submitted using the GET, instead of the POST, method [15].

Ad-based URL shortening services can solve the problem of referrer leakage in various ways. The services that are currently not vulnerable to this attack use an indirect redirection within the service itself. More specifically, instead of setting the source of the advertising `iframe` directly to the URL of a third-party advertiser or advertising network, they set it to a generic page, internal to the service. This page will, in turn, request the advertising URL and thus reveal, through the Referer header, the URL of the generic page, rather than the URL containing the short-link identifier.

Another, more generic, approach would be to change the architecture of the page, so that instead of passing the id specific to each short URL through a GET parameter, it would be passed as a fragment identifier. For instance, a specific short URL of the `adf.ly` service could be `http://adf.ly/#Vtcwh` instead of `http://adf.ly/Vtcwh`. Fragment identifiers are not sent out through the Referer header (thus eliminating short URL leakage) and can be read and consumed by the service itself through JavaScript, as proposed by Close [6].

Users of most modern browsers can also protect themselves, by disabling the sending of the Referer header, although this may interfere with some CSRF countermeasures which rely on the presence of this header [35].

6.3 Adult ads

For the issue of minors exposed to ads of adult content, one solution would be to ban this type of ads altogether, like `adf.ly` does, or to display them only when the shortened URL leads to a destination website that is also of an adult nature. The ad-based URL shortening service `linkbucks.com` follows such an approach, where it requests advertisers to explicitly mark their adult-content ads as "over 18." These ads are then only shown in combination with short URLs which their creators have also marked as "over 18."

While this classification depends solely on each user, and thus, as demonstrated in Section 4.2, does not always work, the service warns that users who misclassify their content may have their accounts banned and their earnings lost. Additionally, lists of adult websites and a scanning of ads for tell-tale signs of adult content can be used to supplement the classification process. These two techniques can also, in principle, be used at the client side by users to stop the display of adult content.

7. RELATED WORK

Traditional, non-ad-based URL shortening services have been thoroughly investigated in previous research.

The first study on short URLs was done by Antoniades et al. [2], who collected approximately 8.5 million distinct short URLs by periodically crawling for bit.ly URLs on Twitter and by brute-forcing the key space of ow.ly. Although these services are the most popular, part of the statistics analyzed were actually calculated from the data offered by bit.ly, and thus may represent a service-centric view of the overall usage. Maggi et al. tracked the use of short URLs on the web by monitoring more than 7,000 real web users for a period of two years [18] and provided insights on where short URLs are found, on the type of content that they point to, and on the security threats associated with them.

Rodrigues et al. [26] analyzed a dataset collected on Twitter between 2006 and 2009. In their work, the authors show that short URLs accounted for 75% of all URLs posted on Twitter in 2009, and that TinyURL and ow.ly were the top services. Chhabra et al. [4] analyzed the number of phishing scams that were posted on Twitter, and that were hidden behind short URLs. They discovered that most phishing on Twitter aims at stealing social network credentials rather than other services. Klien and Strohmaier in [14], performed a geographical analysis of the short URLs from the qr.cx service. Kandylas et al. [13] show that the quality of the landing pages pointed by bit.ly URLs extracted from Twitter is either very high (popular, reputable pages) or very low (spam and malicious pages).

Neumann et al. [21] showed that many URL shortening services that are popular on Twitter have privacy implications both in terms of information disclosure and security. Our analysis of ad-based URL shortening services shows that the studied services still have all the problems of traditional URL shorteners, as outlined by previous work, as well as multiple additional issues, specific to them.

Previous work also showed evidence of malicious content hiding behind short URLs. Stringhini et al. [30] show that most spam campaigns targeting social networking sites leverage short URLs. Gao et al. [11] analyzed the activity of about 3.5 million users on Facebook, and detected a large number of malicious posts with embedded short URLs. Our study of ad-based URL shortening services shows the real danger of users being exposed to malicious content through a short URL, even if the original shortened URL was benign.

A number of systems studied the redirection chains that lead to malicious web pages. WarningBird [16] analyzes the redirection chains generated by visiting short URLs on Twitter, aggregates URLs that point to the same final page, and determines if that final page is malicious or not. Similarly, SpiderWeb looks at the *redirection graphs* that lead to final web pages [31]. Unlike WarningBird, SpiderWeb looks at URL that are posted on any web page, and not only on Twitter. MadTracer [17] is a system that detects malicious ads by inspecting the redirection chains that visitors follow. Our work is complementary, in that it studies the problem of malicious ads from a different perspective, by investigating the threats facilitated by ad-based URL shortening services.

8. CONCLUSION

In this paper, we explored the ecosystem of ad-based URL shortening services from a security and privacy perspective. Because of certain unique attributes of these services, such as the monetary incentive for clicks on shortened URLs, and the involvement of third-party advertising networks, we showed that these services attract abuse that is not typical of traditional URL shortening services. We described how all the investigated services were vulnerable, among others, to the hijacking of short URLs, and demonstrated that many are actively exploited in order to infect users with malware and exfiltrate private data.

Our work highlights the dangers of URLs shortened by this type of services, not necessarily because of a malicious service or malicious link-shortening users, but rather because of the unpredictable, and often malicious, advertisers. To diminish this danger, we proposed a series of straightforward countermeasures, which we hope that ad-based URL shortening services and their users will consider and adopt.

Acknowledgments: We want to thank the anonymous reviewers for the valuable comments. For KU Leuven, this research was performed with the financial support of the Prevention against Crime Programme of the European Union (B-CCENTRE), the Research Fund KU Leuven, and the EU FP7 projects NESSoS, WebSand, and STREWS. For Politecnico di Milano, this research was supported by the FP7 project SysSec funded by the EU Commission under grant agreement no 257007. For UCSB, this work was supported by the Office of Naval Research (ONR) under grant N000140911042, the Army Research Office (ARO) under grant W911NF0910553, and Secure Business Austria.

9. REFERENCES

[1] Alexa Top 1 Million websites. http://www.alexa.com.

[2] Antoniades, D., Athanasopoulos, E., Polakis, I., Ioannidis, S., Karagiannis, T., Kontaxis, G., and Markatos, E. P. we.b: The web of short URLs. In *Proceedings of the 20th international World Wide Web Conference (WWW '11)* (2011).

[3] Caballero, J., Grier, C., Kreibich, C., and Paxson, V. Measuring Pay-per-Install: The Commoditization of Malware Distribution. In *Proceedings of the the 20th USENIX Security Symposium* (San Francisco, CA, August 2011).

[4] Chhabra, S., Aggarwal, A., Benevenuto, F., and Kumaraguru, P. Phi.sh/$oCiaL: the phishing landscape through short URLs. In *CEAS '11* (2011).

[5] Cligs Got Hacked - Restoration from Backup Started. http://web.archive.org/web/20090628221947/http://blog.cli.gs/news/cligs-got-hacked-restoration-from-backup-started.

[6] Close, T. Web-key: Mashing with Permission. In *Proceedings of the 17th Internation World Wide Web Conference* (2008).

[7] Cova, M., Kruegel, C., and Vigna, G. Detection and analysis of drive-by-download attacks and malicious javascript code. In *Proceedings of the 19th International World Wide Web Conference (WWW '10)* (2010).

[8] De Ryck, P., Nikiforakis, N., Desmet, L., and Joosen, W. Tabshots: Client-side detection of tabnabbing attacks. In *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security (AsiaCCS)* (2013).

[9] Eckersley, P. How Unique Is Your Browser? In *Proceedings of the 10th Privacy Enhancing Technologies Symposium (PETS)* (2010).

[10] Contagio Malware Dump - An Overview of Exploit Packs (Update 19.1) April 2013. `http://contagiodump.blogspot.be/2010/06/overview-of-exploit-packs-update.html`.

[11] Gao, H., Hu, J., Wilson, C., Li, Z., Chen, Y., and Zhao, B. Y. Detecting and characterizing social spam campaigns. In *Proceedings of the 10th Internet Measurement Conference (IMC '10)* (2010).

[12] Hidayat, A. PhantomJS: Headless WebKit with JavaScript API. `http://phantomjs.org/`.

[13] Kandylas, V., and Dasdan, A. The utility of tweeted URLs for web search. In *Proceedings of the 19th International World Wide Web Conference (WWW '10)* (New York, NY, USA, 2010), ACM.

[14] Klien, F., and Strohmaier, M. Short links under attack: geographical analysis of spam in a URL shortener network. In *HT '12* (2012).

[15] Krishnamurthy, B., and Wills, C. E. On the leakage of personally identifiable information via online social networks. In *Proceedings of the 2nd ACM workshop on Online social networks* (New York, NY, USA, 2009), WOSN '09, ACM, pp. 7–12.

[16] Lee, S., and Kim, J. WarningBird: Detecting Suspicious URLs in Twitter Stream. In *Proceedings of the Network and Distributed System Security Symposium (NDSS '12)* (2012).

[17] Li, Z., Zhang, K., Xie, Y., Yu, F., and Wang, X. Knowing Your Enemy: Understanding and Detecting Malicious Web Advertising. In *Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS '12)* (2012).

[18] Maggi, F., Frossi, A., Zanero, S., Stringhini, G., Stone-Gross, B., Kruegel, C., and Vigna, G. Two years of short urls internet measurement: security threats and countermeasures. In *Proceedings of the 22nd international conference on World Wide Web (WWW '13)* (2013).

[19] Monga, V., and Evans, B. L. Perceptual image hashing via feature points: performance evaluation and tradeoffs. *IEEE Transactions on Image Processing 15*, 11 (2006).

[20] Naone, E. MIT Technology Review: Get Paid to Install Malware. `http://www.technologyreview.com/view/417354/get-paid-to-install-malware/`.

[21] Neumann, A., Barnickel, J., and Meyer, U. Security and Privacy Implications of URL Shortening Services. In *Web 2.0 Security and Privacy Workshop (W2SP '11)* (2011).

[22] Nikiforakis, N., Balduzzi, M., Van Acker, S., Joosen, W., and Balzarotti, D. Exposing the lack of privacy in file hosting services. In *Proceedings of the 4th USENIX conference on Large-scale Exploits and Emergent Threats (LEET '11)* (2011).

[23] Nikiforakis, N., Kapravelos, A., Joosen, W., Kruegel, C., Piessens, F., and Vigna, G. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *Proceedings of the 34th IEEE Symposium on Security and Privacy* (2013).

[24] Provos, N., Mavrommatis, P., Rajab, M., and Monrose, F. All Your Iframes Point to Us. In *Proceedings of USENIX Security Symposium* (2008).

[25] Raskin, A. Tabnabbing: A new type of phishing attack. `http://www.azarask.in/blog/post/a-new-type-of-phishing-attack/`.

[26] Rodrigues, T., Benevenuto, F., Cha, M., Gummadi, K., and Almeida, V. On word-of-mouth based discovery of the web. In *Proceedings of the 11th Internet Measurement Conference (IMC '11)* (2011).

[27] Rydstedt, G., Bursztein, E., Boneh, D., and Jackson, C. Busting frame busting: a study of clickjacking vulnerabilities at popular sites. In *Web 2.0 Security and Privacy Workshop* (2010), W2SP '10.

[28] Stone-Gross, B., Abman, R., Kemmerer, R., Kruegel, C., Steigerwald, D., and Vigna, G. The Underground Economy of Fake Antivirus Software. In *Workshop on the Economics of Information Security (WEIS)* (2011).

[29] Stone-Gross, B., Stevens, R., Zarras, A., Kemmerer, R., Kruegel, C., and Vigna, G. Understanding fraudulent activities in online ad exchanges. In *Proceedings of the 11th Internet Measurement Conference (IMC'11)* (2011).

[30] Stringhini, G., Kruegel, C., and Vigna, G. Detecting spammers on social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC '10)* (2010).

[31] Stringhini, G., Kruegel, C., and Vigna, G. Shady Paths: Leveraging Surfing Crowds to Detect Malicious Web Pages. In *Proceedings of the 20th ACM Conference on Computer and Communications Security (CCS '13)* (2013).

[32] W3C: Same Origin Policy - Web Security. `http://www.w3.org/Security/wiki/Same_Origin_Policy`.

[33] West, M. Play safely in sandboxed IFrames. `http://www.html5rocks.com/en/tutorials/security/sandboxed-iframes/`.

[34] Wondracek, G., Holz, T., Platzer, C., Kirda, E., and Kruegel, C. Is the Internet for Porn? An Insight Into the Online Adult Industry. In *Ninth Workshop on the Economies of Information Security (WEIS)* (2010).

[35] Zeller, W., and Felten, E. W. Cross-site request forgeries: Exploitation and prevention. Tech. rep., 2008.