Understanding Security Risks of Ad-based URL Shortening Services Caused by Users’ Behaviors

Naoki Fukushima¹,†,a) Takashi Koide²,b) Daiki Chiba²,c) Hiroki Nakano²,d) Mitsuaki Akiyama¹,c)  

Received: March 8, 2022, Accepted: September 2, 2022

Abstract: URL shortening services enable us to shorten or simplify URLs. Ad-based URL shortening services display advertisements to users who access short URLs and reward short URL creators. However, ad-based URL shortening services have specific security risks that URL shortening services without ads do not, such as displaying malicious advertisements to users. In this study, we reveal previously unknown security risks of these services caused by users’ behaviors. We conducted a comprehensive measurement of ad-based URL shortening services. First, we accessed short URLs of these services, clicked buttons on the web pages, and reached the final destinations of the short URLs. Then, we reveal the security risks posed to users by monitoring and analyzing traffic logs when such short URLs are accessed. We found that all services generated an average of 86.5 web requests to malicious domain names per short URL. We then showed the security risk of unintentionally communicating malicious domain names even when users click only on buttons that correctly move users to their desired destinations. Finally, we discuss countermeasures to mitigate these risks from the perspective of each stakeholder in ad-based URL shortening services.

Keywords: URL shortening service, online advertising

1. Introduction

URL shortening services, which shorten and simplify URLs, are one of the essential services on the Internet. For example, by using URL shortening services, we can improve the appearance of URLs themselves or bypass character limits when posting to social media. URL shortening services that generate income by displaying advertisements to users who access short URLs are called ad-based URL shortening services. When accessing short URLs, users reach web pages where advertisements are displayed, and by clicking on buttons on the web pages, the users can reach their desired destinations. To draw users’ attention to the advertisements, these buttons may become clickable after a few seconds. Ad-based URL shortening services can receive advertising revenue from advertising providers, and short URL creators can be rewarded in accordance with the number of users’ accesses to their short URLs. In this way, ad-based URL shortening services can financially benefit services and short URL creators. A previous study reported security and privacy risks of users of ad-based URL shortening services [2]. That study revealed that ad-based URL shortening services display malicious advertisements that automatically forward users to phishing sites. In addition, we found these services have a new strategy for receiving more advertising revenue that is not mentioned in the previous study. For example, these services display many buttons that confuse users as to which button to click or require users to go through multiple web pages. Most of these buttons are ad banners. These complex structures unnecessarily increase the number of users’ clicks to reach the destinations of the short URLs. To the best of our knowledge, no study has revealed the security risks of ad-based URL shortening services caused by users’ browsing behaviors.

In this study, we thus conduct a measurement of these latest ad-based URL shortening services. We consider two types of users’ behaviors to reveal the security risks comprehensively: one is when users click on only buttons that correctly move them to the destinations, and the other is when users mistakenly click on ad banners that look like these buttons. It is difficult for users and security analysts to distinguish these ad banners and buttons on the basis of visual information since these ad banners often change their appearance each time short URLs are accessed. Therefore, we propose a method that automatically detects buttons on the web pages and supports security analysts when accessing short URLs and implemented it as a Chrome extension. Moreover, to analyze malicious web requests, we add a function to the above extension to monitor traffic logs when accessing short URLs. We first comprehensively select ad-based URL shortening services that are heavily accessed by users. Next, we generate short URLs using a benign URL for each selected ad-based URL shortening service. Then, we conduct experiments to access the short URLs
and reach their final destinations using the extension described above. The results of our experiment showed that all services generated web requests to malicious domain names even though the final destination of the short URLs was benign. The average number of web requests to malicious domain names generated from each service was 86.5. These malicious domain names mainly included registration pages for scam services and web pages that ask users to allow malicious web push notifications. We also found that malicious domain names risk being communicated even if users click on only buttons on the web pages. Furthermore, we revealed that users’ risks of communicating malicious domain names increase when they mistakenly click on ad banners that look like buttons on the web pages. Finally, we discuss countermeasures to mitigate these risks from the perspective of each stakeholder in ad-based URL shortening services.

Our main contributions are as follows.

- We comprehensively investigated the structure of latest ad-based URL shortening services. The number of services we investigated is large enough compared with the previous study [2]. We are the first to reveal the security risks caused by two types of users’ browser behaviors to reach the final destination of the short URLs.
- We proposed a method that uses a common feature among ad-based URL shortening services to automatically detect buttons that correctly move users to destinations and support security analysts.
- We found for the first time that users can be forced to communicate with web pages that contain malicious domain names even if they click on only buttons that correctly move them to the destinations.
- We make recommendations to each stakeholder of ad-based URL shortening services on how users use these services, what kind of advertisements should not be displayed, and how these services should be designed.

This paper is an extended version of our paper published in SecureComm 2021 [1]. Herein, we include a report on new measurement studies that clarify actual users’ behaviors when they reach malicious domain names detected in our experiments. To this end, we use statistical data of real user accesses. The additional elements included in this paper are summarized as follows.

- We revealed what other paths users are taking (incoming channel) when they reach malicious domain names detected in ad-based URL shortening services.
- We analyzed actual users’ behaviors when they reach malicious domain names in terms of how long they spent on web pages of the domain names (visit duration) and whether they left without going to other web pages of the domain names (bounce rate), respectively.
- We analyzed the security risk of users who reach malicious domain names in terms of whether the next web pages they reach via the malicious domain names are also malicious.

2. Background

Ad-based URL shortening services have two characteristics: 1) They require users to go through one or more web pages where advertisements are displayed before reaching the final destinations, and 2) they reward short URL creators in accordance with the number of users’ accesses to their short URLs. In the following sections, we describe each characteristic in detail.

2.1 Structure

First, we describe the structure of ad-based URL shortening services. Figure 1 is an overview of the structure of ad-based URL shortening services. In this paper, we refer to the final destination of the short URLs as destination pages. As shown in Fig. 1, a user first accesses a short URL from a referring page (e.g., a web page posted on social media). In the case of URL shortening services without ads (e.g., Bitly [3], TinyURL [4]), once a user clicks on the short URL, he or she will automatically reach a destination page without any further interaction. On the other
hand, in the case of ad-based URL shortening services, users first reach web pages where advertisements are displayed. We refer to these web pages that users go through from accessing short URLs to reaching destination pages as intermediate pages. To reach destination pages, users need to click on buttons on intermediate pages. We refer to these buttons as genuine buttons. Genuine buttons may not be present immediately after the intermediate page loads, or they may not be clickable even if they are present. Therefore, users need to make them clickable, i.e., activate them. A previous study [2] states that genuine buttons are automatically activated a few seconds after users reach intermediate pages. This mechanism enables ad-based URL shortening services to show advertisements for at least a few seconds to users.

Ad-based URL shortening services with new strategies to increase advertising revenue have appeared and are being heavily accessed by users. These services have complex page structures in addition to the above mechanism and have not been reported in previous studies. For example, there are ad-based URL shortening services that require CAPTCHA to be resolved to activate genuine buttons. A CAPTCHA is a type of challenge-response test that is widely used on the web to verify whether a responder is a real human. In the case of the ad-based URL shortening service shown in Fig. 1, the user can activate the genuine button by resolving the CAPTCHA. They are both bordered in red. Also, several ad-based URL shortening services display ad banners that look like genuine buttons on intermediate pages. In this paper, we refer to these buttons as lure buttons. In the case of the ad-based URL shortening service shown in Fig. 1, the lure buttons are bordered in blue. These lure buttons are ad banners that are displayed by advertising providers in the iframe or generated as images on intermediate pages. Users cannot reach destination pages by clicking on these buttons, but third-party ad pages will be opened. Note that these ad banners are not unique to ad-based URL shortening services. Ad-based URL shortening services have not been revealed so far.

2.2 Reward
Ad-based URL shortening services reward short URL creators in accordance with the number of accesses to their short URLs. To describe this mechanism, we show the stakeholders of ad-based URL shortening services in Fig. 2. There are five types of stakeholders: ad-based URL shortening services, short URL creators, users (who access short URLs), advertising providers, and advertisers. Advertising providers display advertisements created by advertisers on intermediate pages, and users go through these web pages before reaching destination pages. When users click on these advertisements on the intermediate pages, ad-based URL shortening services can receive advertisement revenue. Short URL creators can be rewarded with a portion of this advertising revenue.

3. Method
3.1 Design and Technical Points of the Proposed Method
Our goal is to understand the security risks caused by users’ behaviors when they reach the destination pages from ad-based URL shortening services. We consider two types of users’ behaviors. One is when users click on only genuine buttons on the intermediate pages. The other is when users mistakenly recognize and click on lure buttons that resemble genuine buttons. By considering two types of users’ behaviors, we can comprehensively reveal the security risks caused by user’s behaviors. To shed light on such risks, we analyze web requests that occur on the intermediate pages when we access the short URLs. In this experiment, we need to distinguish between the genuine button and misleading lure buttons on the intermediate page. However, the lure buttons are not visually identifiable because they change their appearance every time we visit the intermediate page.

In this paper, we propose a method to support the analysis of ad-based URL shortening services by detecting genuine buttons without visual information. We implemented this method as a Chrome extension to detect a genuine button when we visit the intermediate page of an ad-based short URL. The key point of this method is to take advantage of the common feature among ad-based URL shortening services that the DOM changes when the genuine button is activated. Our method is very different from the method proposed in the previous study [5], which uses visual information to identify trick banners that mislead users. To clarify the security risks in detail, we need to monitor and analyze the traffic logs, such as URL redirections and URLs loaded in the iframe. We also added a function to the extension to automatically monitor and save web requests. In the following sections, we explain two functions: detecting genuine buttons and monitoring web requests.

3.2 Detecting Genuine Buttons and Manual Operations
In this section, we describe the function of detecting genuine buttons. In this section, we describe the function of detecting genuine buttons. This function detects genuine buttons on intermediate pages and shows us their locations. This eliminates the need for security analysts to select the next element to click. Figure 3 shows the three steps of this function: activating a genuine
button, scoring web elements, and detecting a genuine button. Figure 3 also shows the manual operations required to proceed with the steps in gray with a human icon. We perform two types of manual operation, resolving a CAPTCHA and clicking on a genuine button. We repeat each step and manual operation until we reach a destination page.

Step 1: Activating genuine buttons
In step 1, this function activates a genuine button, which is not clickable or visible upon an intermediate page is loaded. There are two conditions for activating genuine buttons: users wait for a few seconds and resolve CAPTCHA. In the former condition, ad-based URL shortening services aim to make users pay attention to online advertisements for a certain period of time. The purposes of the latter are to avoid crawling by bots or bypassing the services by adblockers. First, this function checks the current DOM for the presence of a CAPTCHA. For example, Google reCAPTCHA is determined if there is an element whose `html` tag is `iframe` and its `src` attribute starts with `https://www.google.com/recaptcha/api2`. Second, if CAPTCHA is present, our extension changes its Cascading Style Sheet (CSS) style to highlight it and we manually resolve it. If it is not present, our extension waits for up to 30 seconds before proceeding to step 2, because the genuine button is activated a few seconds after the intermediate page is loaded. Our extension saves two DOMs, one upon loading intermediate pages, and one a few seconds after intermediate pages are loaded. These two DOMs are used in step 3 to detect genuine buttons.

Step 2: Scoring web elements
In step 2, this function selects candidates of genuine buttons from DOMs. To do this, we designed a scoring algorithm to evaluate web elements by focusing on two features of genuine buttons. We implemented this scoring algorithm because without this scoring, considering only DOM differences would select many other elements as candidates, and our proposed method cannot uniquely detect genuine buttons. The first feature is that genuine buttons often contain “btn” or “button” in the `class` and `id` attributes. The second feature is that many genuine buttons contain the terms such as “click” and “continue” in the `value` attribute and their text content to encourage users to click. Thus, we calculated the score as the number of terms shown in Table 1 included in each of the above attributes. The elements to be scored are those with the `a`, `button`, `img`, and `input` tags. We explain the selection process for terms shown in Table 1. We pre-analyzed the DOM of ad-based URL shortening services and collected samples of genuine buttons. Then, using the previous study as a reference, we collected terms that indicate buttons and that encourage users to click. Only elements with scores greater than or equal to 2 will be further analyzed in the next step. The reason the score threshold was set to 2 is that an element with a score of 2 or higher may possess both the aforementioned features. Since this scoring is only used to select candidates for genuine buttons, it is sufficient to use a simple process that focuses on not missing the candidates rather than on accuracy.

Step 3: Detecting genuine buttons
In step 3, this function detects genuine buttons from candidates selected in step 2 by using differences in DOMs. Here, the algorithm we used to compare two DOMs was difflib, which is implemented in Javascript. When a non-existent or non-clickable genuine button becomes active, there should be a difference in the source code of the intermediate page. For example, a genuine button that is not clickable due to the presence of the `disabled` attribute becomes active when this attribute is removed. First, this function compares two source codes (one upon loading the intermediate page and the other after activating genuine buttons) and extracts the changed or newly created elements. Next, this function finds the union of these elements and the candidate elements selected in step 2. The element with the highest score, i.e., the genuine button, is highlighted by changing its CSS style. Finally, we click on the genuine button. These steps can be repeated until the browser reaches the destination page.

### 3.3 Monitoring Web Requests
The second function automatically records web requests from a web browser for the time between accessing short URLs and reaching the destination pages. Specifically, we used the `chrome.webRequest` API to monitor every web request, e.g., a web page is newly loaded in a browser tab, an `iframe` is generated on the web page, a JavaScript file is loaded, and a URL redirection occurs. The types of web requests can be identified from the `ResourceType`. For example, if the `ResourceType` is `main_frame`, the web request is used to load the main content of a web page, and its URL is displayed in the browser’s URL bar. If the `ResourceType` is `sub_frame`, the traffic is requested from an `iframe` element. Also, this function obtains a screenshot of the web page after all resources on the web page are loaded.

### 3.4 Operations We Perform Manually
We use Google Chrome with the extension that implements the proposed method to analyze ad-based URL shortening services. We organize the operations we perform manually to reach the destination pages of short URLs. What we need to do is to resolve CAPTCHA and click on a genuine button that our extension detected on the intermediate page. At that time, we must repeat-
4. Experimental Setup

We conduct a measurement study of ad-based URL shortening services. The purpose of our measurement is to reveal the security risks caused by users’ behaviors. In this section, we describe the experimental setup of our measurement. We first explain the selection criteria for the ad-based URL shortening services to investigate. Next, we conduct a preliminary experiment to evaluate our proposed method described in Section 3.2. Finally, we explain the environment and procedure of our measurement.

4.1 Selecting Ad-based URL Shortening Services to Investigate

In this section, we describe the criteria for selecting the ad-based URL shortening services to investigate. We decided to select ad-based URL shortening services that are online as of January 2021 and accessed by a large number of users. We first used a search engine to search for terms related to ad-based URL shortening services, such as “high paying URL shortener” and “best URL shortening service.” We then selected a total of 35 online ad-based URL shortening services from the web pages indexed as search results. To select services accessed by many users, we used the Tranco List [10] obtained in January 2021, which is a list of one million domain names ranked by popularity. Finally, we selected a total of 30 services that use domain names listed in the Tranco List for our measurement, out of 35 ad-based URL shortening services. In the Tranco rank column of Table 2, we show the Tranco rank lists for the domain names used by each selected ad-based URL shortening service.

We then generated one short URL for each of these 30 ad-based URL shortening services using a URL of a certain benign web page and investigated them in the following experiments. This means that the destination page is always a single URL that we prepared. In the following experiments, this means that the destination frame is always the frame of the supplied URL. In the experiments, we refer to the following frame and the genuine button described in Section 3.2 and its result. We evaluate our proposed method in this section. Finally, we explain whether the button detected by the proposed method is the correct button.

4.2 Evaluating Our Proposed Method

In this section, we describe the preliminary experiment we conducted to evaluate our proposed method of detecting the genuine button described in Section 3.2 and its result. We evaluate whether the button detected by the proposed method is the correct button.
rect genuine button or not. In ad-based URL shortening services, when we click on lure buttons other than the genuine button, we cannot reach the destination page. If we can reach the destination page by clicking on only the button detected by the proposed method, it means that all the detected buttons are the correct genuine buttons. We accessed a total of 30 short URLs of the ad-based URL shortening services we selected in Section 4.1. We then investigated whether we could reach the destination page by clicking on only the buttons detected by the proposed method. Here, these short URLs are created by us, so we can find out for ourselves whether we have reached the destination page or not. In the experiment, we reached the destination page from all short URLs, which means that our method correctly detected all genuine buttons. Our proposed method takes advantage of a common feature among ad-based URL shortening services: the source code change when the genuine button is activated. Therefore, our proposed method will work with versatility even when new ad-based URL shortening services are created in the future.

4.3 Environment and Procedure of Our Measurement

We consider two types of users’ behaviors to reveal the security risks comprehensively: one is when users click on only genuine buttons, and the other is when users mistakenly recognize and click on lure buttons that look like genuine buttons. First, we investigate the security risks when users click on only genuine buttons. We prepare Google Chrome with our original Chrome extension enabled, which detects genuine buttons and monitors traffic, as described in Section 3. We access a total of 30 short URLs with this Google Chrome. Then, we reach the destination page by clicking on only the genuine buttons detected by the proposed method. Also, we analyzed the differences in displayed advertisements caused by users’ regional information. We changed our source IP addresses using a virtual private network (VPN) service. We chose six regions that can be selected by the VPN service: United States (US), Brazil (BR), Japan (JP), Germany (GE), Hong Kong (HK), and United Kingdom (UK). We selected these regions as geographically distant regions with a large number of Internet users. For our selection above, we used Statista’s survey [11] of the number of Internet users by region in 2020.

Next, we investigate the security risks when users click on lure buttons. We manually distinguished such lure buttons from ad banners displayed on the intermediate pages. Lure buttons contain terms that encourages users to click, such as “DOWNLOAD” or “CLICK HERE.” These buttons also contain video play buttons or downward arrows that remind users to download. We conduct an experiment to click on all the lure buttons on the intermediate page and then click on the genuine button to reach the destination page. We experimented with one short URL for each service using multiple environments, instead of experimenting with multiple URLs using the same environment. This is because the previous study [2] has shown that the advertisements displayed on the intermediate pages depend on the environment and behavioral history of users who accessed short URLs, not the destination pages. We also found that the structures from the intermediate page to the destination page of each ad-based URL shortening service such as the number of intermediate pages were always the same regardless of the destination pages. Specifically, we generated five different short URLs per service and confirmed that the aforementioned structure did not change depending on original URLs. Note that the experiments with one short URL for each service do not allow us to fully observe all possible advertisements that may be displayed. See Section 6.1 for more on this limitation.

5. Measurement Results

In this section, we describe the results of our measurement of the ad-based URL shortening services. In Section 5.1, we first describe the results of our experiment to reach the destination pages by clicking on only the genuine buttons. In Section 5.2, we next describe the results of our experiment to reach the destination pages when we clicked on the lure buttons in addition to the genuine buttons. In Section 5.3, we finally describe the results of our experiment to access the short URLs with Adblocker enabled. In Section 5.4, we describe the results of our analysis of paths to malicious domain names detected in ad-based URL shortening services (incoming channel), actual users’ behaviors when reaching malicious domain names, and users’ next destinations. We leverage results of analyzing real user traffic.

5.1 Clicking on Only Genuine Buttons

In this section, we describe the results of our experiment to access the short URLs and reach the destination pages by clicking on only genuine buttons. We conducted our experiment a total of 180 times (the number of combinations of 30 ad-based URL shortening services and 6 source IP addresses varying regional information by using the VPN service described in Section 4.3). We fixed our browser language settings to en-US and conduct each experiment at the same time in January, 2021. With this experiment, we reveal the structures to the destination pages of the ad-based URL shortening services, the security risks caused by users’ clicks on the genuine buttons and its regional characteristics.

5.1.1 Structure of Services

In this section, we reveal the latest structures to the destination pages of the ad-based URL shortening services. We investigated these structures in terms of the number of intermediate pages users need to go through before reaching the destination pages. We examined this number for each ad-based URL shortening service. We show the results in the Intermediate pages column of Table 2. As shown in Table 2, our measurement of 30 ad-based URL shortening services showed that 5 services had 1 intermediate page, 19 services had 2 intermediate pages, 5 services had 3 intermediate pages, and 1 service had 4 intermediate pages. The number of intermediate pages is equal to the number of genuine buttons we clicked. Therefore, this result indicates that users need to select and click on the genuine buttons at least twice in more than 80% of the ad-based URL shortening services we investigated. Such a structure of ad-based URL shortening services requiring users to go through multiple intermediate pages has not been mentioned in previous studies.

Moreover, of the 62 genuine buttons (the total value of the Intermediate pages column) we clicked in the 30 ad-based URL
shortening services, only 7 were active immediately after the intermediate pages loaded, and the remaining 55 required users to wait for some time or resolve CAPTCHA to activate. For each ad-based URL shortening service, we indicate “✓” in the C column of Table 2 when CAPTCHA resolution was required to activate the genuine button. The number of ad-based URL shortening services that required CAPTCHA resolution was 24 out of 30, and the number of CAPTCHA resolutions required to reach the destination page was 1 for all of these services. Also, ad-based URL shortening services investigated in the previous study [2], indicated by “[2]” in the Novel column of Table 2, did not require CAPTCHA resolution to activate the genuine buttons. These results indicate that the structure to the destination pages of the latest ad-based URL shortening services has changed compared with those investigated in previous studies.

5.1.2 Security Risks Posed to Users

In this section, we reveal the security risks when users click on only genuine buttons. We extracted a total of 2,003 unique domain names from the web requests monitored during our experiments. The above 2,003 domain names exclude the domain names of both the short URLs and the destination page. To determine if these domain names were malicious, we used VirusTotal [12]. By scanning domain names with VirusTotal, we can retrieve URLs containing domain names identified as malicious in the past, along with the number of vendors that identified the URLs as malicious. We decided to consider domain names with at least one URL that has been identified as malicious by at least one vendor in the past to be malicious domain names. Using VirusTotal API to scan 2,003 domain names, we found a total of 688 malicious domain names. As explained in Section 3.3, web requests can be classified by the ResourceType, such as main_frame (loaded in the main content), sub_frame (loaded in <iframe>), and script (loaded in <script>). Table 2 shows the number of malicious domain names communicated by each ad-based URL shortening service, categorized by the ResourceType. The Total (unique) column shows the number of unique malicious domain names including all ResourceType. The sum of each ResourceType column does not match the value in the Total (unique) column because some malicious domain names use multiple request types. We found that all services generated an average of 86.5 (= 2594 ÷ 30) web requests to malicious domain names. Table 2 shows that 24 ad-based URL shortening services had malicious domain names loaded in the main_frame, i.e., included in the main content displayed in the browser tab. The average number of those domain names was 8.9 for these 24 ad-based URL shortening services, and the maximum number was 24 for Service #25.

Here, we divide the malicious domain names loaded in the main_frame into two types. One was loaded during URL redirection, and the other was loaded after the redirection. We checked the screenshots of web pages that contain the latter domain names identified as malicious by VirusTotal, i.e., finally reached malicious web pages. As described in Section 3.3, these screenshots were taken after all web requests were completed. The total number of unique malicious domain names of the finally reached web pages was 48. The web pages containing these malicious domain names included registration pages for scam services that attempted to steal personal and credit card information by tricking users into creating accounts and adult advertising pages. These web pages also included web pages asking users for permission for web push notifications, which was shown to deliver many malicious advertisements via web push notifications in the previous study [13]. We investigated why these malicious web pages were loaded as main contents loaded in the browser tab.

We found that transparent advertisements covered genuine buttons and CAPTCHA and that the onclick attributes of the genuine buttons were set to window.open, which is used to display third-party ad pages. In particular, we investigated the presence of transparent advertisements covering only genuine buttons for the top 10 ad-based URL shortening services on the Tranco List out of the 30 services. We found that out of a total of 19 genuine buttons for the 10 services, 5 of them displayed transparent advertisements covering only genuine buttons. In other words, even if users click on genuine buttons or resolve CAPTCHA, which are the minimum required operations to reach the destination page, the users will reach malicious web pages. Users will be then at risk of having their personal information stolen by registering scam services or receiving malicious web push notifications. When users are directed to multiple intermediate pages, the number of their clicks on the genuine buttons increases. Consequently, the risk of reaching malicious web pages also increases.

5.1.3 Region Characteristics

We conducted our experiments from IP addresses in six regions by using a VPN service. By doing this, we could consider that displayed advertisements vary by regions of users accessing short URLs. In this section, we conduct a region-by-region analysis of the 688 unique malicious domain names found in the section above. Table 3 shows the number of unique malicious domain names detected in a total of 30 ad-based URL shortening services per region. We found that an average of 410.5 malicious domain names were detected per region. The highest number was 442 in Brazil, and the lowest was 378 in Japan. For a total of 688 of these unique malicious domain names, we investigated the number of regions in which each domain name was detected. Table 4 shows the results of this investigation. The largest number of regions in which each domain name was detected was 6 and the second

### Table 3

<table>
<thead>
<tr>
<th>Region</th>
<th># of malicious domain names</th>
</tr>
</thead>
<tbody>
<tr>
<td>BR</td>
<td>442</td>
</tr>
<tr>
<td>HK</td>
<td>419</td>
</tr>
<tr>
<td>US</td>
<td>415</td>
</tr>
<tr>
<td>GE</td>
<td>407</td>
</tr>
<tr>
<td>UK</td>
<td>402</td>
</tr>
<tr>
<td>JP</td>
<td>378</td>
</tr>
</tbody>
</table>

### Table 4

<table>
<thead>
<tr>
<th># of regions</th>
<th># of malicious domain names</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 region</td>
<td>225</td>
</tr>
<tr>
<td>2 regions</td>
<td>74</td>
</tr>
<tr>
<td>3 regions</td>
<td>50</td>
</tr>
<tr>
<td>4 regions</td>
<td>32</td>
</tr>
<tr>
<td>5 regions</td>
<td>30</td>
</tr>
<tr>
<td>6 regions</td>
<td>277</td>
</tr>
<tr>
<td>Total</td>
<td>688</td>
</tr>
</tbody>
</table>

© 2022 Information Processing Society of Japan
largest was 1, accounting for about 73% (≈ (277 + 225) ÷ 688) of the total. This result shows that there are malicious domain names that are specific to users’ regions and independent of users’ regions. We focused on the 225 malicious domain names that were detected in 1 region. Table 5 shows the number of malicious domain names detected in each region only. The regions with the highest and lowest numbers of malicious domain names detected only in those regions were Brazil, with 81, and Japan, with 12. Also, we investigated the region-by-region access status of these malicious domain names. We used SimilarWeb [14], which passively observes traffic of hundreds of millions of global devices and covers over 220 regions. We extracted the top 5 regions sending web requests to the 225 malicious domain names. The (2) # of included in top 5 regions column of Table 5 shows the number of domain names whose regions were included in the top 5 regions. The (3) # of missing data column shows the number of domain names for which we could not acquire SimilarWeb’s data. As shown in the (2) # of included in top 5 regions column, for a total of 95 malicious domain names, the regions where we detected them were included in the top 5 regions. This percentage is 47% (≈ 95 ÷ (225 ÷ 23)) for a total of 6 regions, indicating that about half of the malicious domain names were accessed by many users in the regions in which we detected these malicious domain names. Moreover, we found that in some cases, the regions where these 225 malicious domain names were detected coincided with the regions of the ccTLD of these malicious domain names. For example, 9 of the 30 malicious domain names detected in GE only were acquired under .de. A simple consideration would be that there should be more access from users in the region with the largest number of Internet users. We believe that the correlation between ccTLDs and the regions with the highest number of user accesses is sufficient to lead to the discussion that users in those regions reached more malicious domain names targeting users in those regions. In summary, malicious domain names of web requests generated by users’ clicks on intermediate pages may depend on the users’ regions.

5.2 Clicking on Genuine Buttons and Lure Buttons

In Section 5.1, we revealed the security risks when clicking on only the genuine buttons. In this section, we reveal the security risks when clicking the lure buttons. We manually selected ad banners as lure buttons that users might misidentify as genuine buttons. As explained in Section 4.3, lure buttons are ad banners that contain terms that encourage users to click, such as “DOWNLOAD” or “CLICK HERE,” video play buttons, or downward arrows that remind users to download. Also, lure buttons exclude advertisements of products or services. In February 2021, we conducted an experiment to access 30 short URLs generated one-by-one from all the 30 ad-based URL shortening services. We accessed these short URLs from IP addresses in the US and reach the destination pages. In this experiment, when the lure buttons were displayed on the intermediate pages, we clicked all of them before clicking on the genuine buttons. The third-party ad pages we reached by clicking lure buttons were basically loaded in a new browser window. Exceptionally, when one lure button was clicked for one of the 30 services, a third-party ad page was loaded in the current browser tab as well as in a new browser tab. In this case, we accessed the short URL of the service again, and the second time we reached the destination page without clicking on the lure button above. Also, if we went through multiple intermediate pages before reaching the destination pages, we checked if the lure buttons were displayed on each intermediate page. We analyzed web requests that occurred when clicking on the lure buttons and web requests that occurred when clicking on the genuine buttons.

The Lure buttons column of Table 2 shows the number of lure buttons displayed in each ad-based URL shortening service, and the Malicious domains column shows the number of malicious domain names included in the URLs loaded in the main_frame when clicking on the lure buttons. As explained in Section 3.3, URLs loaded in the main_frame are URLs of the main contents loaded in the browser tab. In Table 2, the number of lure buttons does not necessarily correspond to the number of malicious domain names communicated. This is because clicking on different lure buttons caused web requests to the same domain names, while clicking on a lure button several times caused web requests to different domain names. As shown in the lure column of Table 2, 15 of the 30 ad-based URL shortening services displayed lure buttons, and the average number of the lure buttons displayed by these 15 services was 2.9, with the largest number being 8 for Service #19. Moreover, the Malicious domains column of Table 2 shows the number of malicious domain names included in the URLs loaded in the main_frame when we clicked the lure buttons. The average number of these malicious domain names was 4.2 in 15 services, and the largest number of these malicious domain names was 7 in Services #12 and #21. As shown in Section 5.1.2, even if users click only genuine buttons, communications to malicious domain names may occur. Moreover, advertisers display lure buttons to encourage users to click them and receive advertising revenue. When users mistakenly click these lure buttons believing that they are genuine buttons, the risk of users communicating with malicious domain names will increase.

### Table 5 Breakdown of malicious domain names detected in one region and results of analysis using SimilarWeb’s data.

<table>
<thead>
<tr>
<th>Region</th>
<th>(1) # of malicious domain names</th>
<th>(2) # of included in top 5 regions</th>
<th>(3) # of missing data</th>
<th>Percentage (2) ÷ (1) − (3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BR</td>
<td>81</td>
<td>30</td>
<td>4</td>
<td>39%</td>
</tr>
<tr>
<td>US</td>
<td>38</td>
<td>29</td>
<td>4</td>
<td>85%</td>
</tr>
<tr>
<td>UK</td>
<td>33</td>
<td>18</td>
<td>1</td>
<td>56%</td>
</tr>
<tr>
<td>HK</td>
<td>31</td>
<td>3</td>
<td>6</td>
<td>12%</td>
</tr>
<tr>
<td>GE</td>
<td>30</td>
<td>13</td>
<td>7</td>
<td>57%</td>
</tr>
<tr>
<td>JP</td>
<td>12</td>
<td>2</td>
<td>1</td>
<td>18%</td>
</tr>
<tr>
<td>Total</td>
<td>225</td>
<td>95</td>
<td>23</td>
<td>47%</td>
</tr>
</tbody>
</table>
more than when users only click genuine buttons.

5.3 Anti-Adblocking

Our experiments above have shown that there is the risk of visiting malicious domain names due to user clicks intended to reach the destination pages. For users to reduce such risks, they may install adblockers on their browsers to filter out communications to advertising domain names. On the other hand, anti-adblocking [15] is a common tactic for site owners against adblockers. If this is used by ad-based URL shortening services, adblocking [15] is a common tactic for site owners against ad-blockers. This is when users access malicious domain names due to user clicks intended to reach the destination pages. We chose two of the leading adblockers (AdBlock [16] and Adblock Plus [17]) and used them with the default filtering list. Table 2 shows the ad-based URL shortening services for which the anti-adblocking feature worked when the two adblockers were enabled. We were unable to reach the destination pages in 19 services with AdBlock and 18 cases with Adblock Plus. These services with anti-adblocking features showed warning messages asking to disable adblockers. Then, the genuine button was hidden on the intermediate page. This result shows that it is difficult to use ad-based URL shortening services and adblockers at the same time to reduce security risks related to malicious advertisements.

5.4 Actual Users’ Behaviors

In the previous sections, we analyzed security risks posed to users by actually performing browser operations that users who access ad-based short URLs perform to reach destination pages. As a result of our analysis, we found that users unintentionally reach malicious domain names. In this section, on the basis of our analysis of real traffic of real users, we reveal other paths users are taking to reach malicious domain names detected in ad-based URL shortening services (hereinafter referred to as “incoming channels”), users’ actual behaviors when they reach malicious domain names, and users’ next destinations. Figure 4 shows the overview of browser operations performed by users who access ad-based short URLs. After accessing short URLs, when users click on ad banners (that contain lure buttons) or genuine buttons on intermediate pages, they reach third-party ad pages that could be malicious. Users may reach other malicious web pages by performing browser operations on third-party ad pages. Incoming channels to third-party ad pages other than intermediate pages of ad-based URL shortening services may include search engine results, emails, web pages, etc. As shown in Section 5.1.2, we found a total of 688 malicious domain names when we clicked on only genuine buttons. In this section, we analyze incoming channels to these malicious domain names, actual users’ browser behaviors when they reach these malicious domain names, and users’ next destinations, which are shown in red in Fig. 4. To this end, we leverage statistical data provided by SimilarWeb [14] (measured over three months from December 2020 to February 2021), which we also used for our analysis of regional characteristics in Section 5.1.3. This SimilarWeb data enables us to analyze actual users’ browser behaviors before and after reaching malicious domain names on the basis of data from real users’ traffic.

5.4.1 Incoming Channel

In this section, we analyze incoming channels for 583 of the 688 malicious domain names for which statistics existed in SimilarWeb. Domain names that are not sufficiently accessed by users are not included in our analysis, as SimilarWeb does not provide statistical data for them. Specifically, for each malicious domain name, we analyze the percentage of traffic volume accounted for by the seven types of incoming channels defined by SimilarWeb (Direct, Referral, Display ad, Organic search, Social media, Paid search). Direct is when users enter URLs directly or access URLs from bookmarks. Referral is when users access URLs from other web pages. Display ad is when users access URLs from advertisements on web pages. Organic search is when users access URLs from search engine results. Social media is when users access URLs from social media such as Facebook. Mail is when users access URLs from hyperlinks on emails. Paid search is when users access URLs from keyword advertisements displayed on search engine results. Note that Mail only targets access from web mails. Access from email client software or other applications is included in Direct. We reveal from which incoming channel users reached malicious domain names detected in ad-based URL shortening services. Figure 5 shows the percentage of traffic that each incoming channel accounts for in the quartiles. From Fig. 5, we can see that Direct accounts for the largest percentage of traffic, with a median value of about 80%. In other words, users directly accessed many malicious domain names detected in ad-based URL shortening services. On the other hand, Referral and Display ad, which
correspond to cases where intermediate pages of ad-based URL shortening services and the advertisements displayed on them are the incoming channels, account for a smaller percentage of traffic than Direct, with a median value of less than 20 percent for both. Also, Organic search, Social media, Mail, and Paid search accounted for a smaller percentage of traffic, with all having median values of less than 2%. These results suggest that incoming channels of malicious domain names we detected are not limited to ad-based URL shortening services. In other words, many events that trigger users to access malicious domain names are not browser events. For example, cyber-attacks measured as Direct include phishing attacks through the sharing of phishing links and attacks in which users are forced to communicate to malware download sites through the execution of macros embedded in Microsoft Word file.

5.4.2 Visit Duration and Bounce Rate

In this section, we reveal actual users’ behaviors when they reach malicious domain names by using SimilarWeb statistics. Specifically, we analyze how long users spent on web pages including target domain names, and whether they left without going to other web pages of the domain names, in terms of visit duration and bounce rate, respectively. By comparing the visit duration and bounce rate for malicious domain names and domain names of generic ad pages, respectively, we test the hypothesis that “users are unintentionally directed to malicious domain names even though they only click on genuine buttons”. Of the 688 malicious domain names, we analyzed users’ visit duration and bounce rate for all 517 and 588 malicious domain names for which statistics existed in SimilarWeb. As a result, we found that the average visit duration was 71.7 seconds, and the average bounce rate was 69.5%. We also examined the average visit duration and the average bounce rate for generic ad pages. For comparison, by using SimilarWeb statistics, we used the top 100 most popular sites on Tranco List [10] obtained in February 2022 and selected a total of 530 domain names of ad pages with a high volume of traffic reached via advertisements displayed on those sites. Of the 530 ad domain names, we analyzed users’ visit duration and bounce rate for all 521 and 524 domain names for which statistics existed in SimilarWeb. As a result, we found that the average visit duration was 216.8 seconds, and the average bounce rate was 54.1%. The shorter visit duration and higher bounce rate of malicious domain names compared with generic ad pages suggest that users may have unintentionally reached these malicious domain names that they were not interested in.

5.4.3 Outgoing Traffic

In this section, we investigate the next web pages users reach after reaching malicious domain names. Here, the next web pages are defined as web pages users reach via referral links from current web pages. We reveal the risk of users being directed to other malicious domain names from malicious domain names reached from ad-based URL shortening services. Specifically, for each of the 688 malicious domain names, we collected the top 10 domain names that users visit next when statistics exist in SimilarWeb. We then scanned each domain name using VirusTotal [12] to determine if the domain names were malicious. As in Section 5.1.2, we decided to consider domain names with at least one URL that have been identified as malicious by at least one vendor in the past to be malicious domain names. We collected a total of 994 domain names with high traffic that users visited after reaching each malicious domain name by using SimilarWeb. We scanned these domain names with VirusTotal and found that more than half (504) of them were malicious. In other words, malicious domain names reached from ad-based URL shortening service were starting points of another cyber-attack or multi-step cyber-attacks that start from landing pages and require users’ browser interactions on each web page in turn [18]. This result underscores the significant security risk of reaching malicious domain names from ad-based URL shortening services.

6. Discussion

In this section, we first describe the limitation of the proposed method, our implementation, and our experiments. Then, we make recommendations for each stakeholder of ad-based URL shortening services. Finally, we explain the ethical considerations of this study.

6.1 Limitations

The following sections explain the limitations regarding using the proposed method to detect genuine buttons, implementing the proposed method to analyze ad-based URL shortening services, and our experiments.

Detecting genuine buttons First, we describe the limitation of the proposed method’s function for detecting genuine buttons. This function detects genuine buttons that have both characteristics inherent to the button elements and characteristics that encourage users to click on them. To evade detection by our method, ad-based URL shortening services can change genuine buttons to elements that do not have such characteristics or add many dummy elements with such characteristics. However, creating such an element would make it difficult for users to identify the correct genuine button, thus lowering the reputations of the services. Also, our method detects genuine buttons on the basis of the difference in the DOM when the buttons become active on the intermediate pages. If ad-based URL shortening services disable the activation mechanism of genuine buttons, i.e., activating genuine buttons by resolving CAPTCHA or making users wait for a few seconds, our method cannot detect the buttons effectively. In this case, the services cannot keep users on the intermediate pages to show the advertisements. In addition, they cannot prevent crawling by bots or bypassing by browser extensions. As mentioned above, although there are techniques to evade our method, none is effective and realistic. Therefore, even if new ad-based URL shortening services are created in the future, our method will be able to detect genuine buttons with versatility.

Implementation Second, we explain two limitations in terms of our implementation to reach the destination pages from short URLs. The first limitation is that our method requires analysts to perform manual operations, such as clicking detected genuine buttons and resolving CAPTCHA. By using common browser automation tools (e.g., Selenium [19], Puppeteer [20]) and human-powered CAPTCHA solving services [21], the proposed method can automate analysis of ad-based URL shortening services con-
taining CAPTCHA. However, due to the previous study [21] showing that CAPTCHA solving services are involved in low-wage work, we conducted a manual analysis in this paper. The second limitation is that we analyzed ad-based URL shortening services using Google Chrome for desktop computers but not mobile phones. Accessing the services from a mobile environment may change advertisements displayed on the intermediate pages and domain names reached from them. We should analyze these services from mobile environments to comprehensively understand the security risks in the future.

**Experiments** Finally, we explain the limitations regarding our experiments in terms of the experimental setting, displayed advertisements, and the data. For the experimental setting, we did not change the behavioral history when accessing short URLs. Changing the behavioral history may change advertisements displayed on intermediate pages. However, it is impractical to repeat the experiment of accessing short URLs by changing the behavioral history of countless possible patterns.

For the displayed advertisements, we were not able to observe all possible advertisements displayed because the online advertisement is not a mechanism that can be controlled by the client side. We conducted two separate experiments, one in which only the genuine button was clicked, and the other in which the lure button was clicked in addition to the genuine button, while changing the source IP address using a VPN service in our experiment. Thus, we accessed the same short URL many times and observed the changes of displayed advertisements, however, there is no guarantee that those are all the advertisements that could be displayed.

For the data, although the SimilarWeb’s traffic data we used is based on observations of real users’ behavior, it is not exhaustive. Ideally, user experiments should be conducted to directly observe and analyze actual user traffic. However, even if this were done, it would be difficult to directly observe the incoming channels to malicious advertisements displayed on intermediate pages, and the scale of observation would be limited. Thus, we adopted the SimilarWeb’s data as a feasible and realistic approach in our study.

**6.2 Recommendations**

We make recommendations to each stakeholder of ad-based URL shortening services on the basis of the findings in Section 5.

**Users of ad-based URL shortening services** The measurement result of Section 5.1.2 showed that if a user only clicks on genuine buttons to reach the destination page without clicking on any of the displayed advertisements or lure buttons, he or she may reach malicious domain names. Also, Section 5.3 revealed that about 60% of the ad-based URL shortening services implemented the anti-adblocking function, indicating that it is difficult for users to use the adblockers and these services at the same time. Even if users are guided to short URLs of ad-based URL shortening services by attractive content, they should be aware of the above-mentioned risks and not access them carelessly.

**Ad-based URL shortening services** Section 5.1.1 revealed the increasingly complex of ad-based URL shortening services’ structure. These services enhance the chances of user clicks by preparing multiple intermediate pages and placing many lure buttons, which in turn increases the risk of reaching malicious domain names. We argue these services should not create such web pages that trick users into making mistakes. For example, these services should have only one intermediate page. In that case, the number of genuine buttons that users need to click on will be one. Reducing the number of opportunities for users to select and click genuine buttons could reduce the risk of communicating malicious domain names that result in social engineering attacks described in Section 5.1.

**Advertising providers** Section 5.2 revealed that half of the 30 ad-based URL shortening services we analyzed displayed at least one lure button. Advertising providers should not display misleading ad banners to users. Users should be able to easily distinguish between ad banners and genuine buttons. The advertising policy published by Google [22] states that buttons that are difficult for users to recognize as advertisements should not be displayed.

**6.3 Ethical Considerations**

In our experiment, we created accounts for ad-based URL shortening services and generate short URLs. We never received any compensation from any ad-based URL shortening services. We also tried to minimize clicks to avoid ad fraud when we click on ad banners displayed on intermediate pages. To avoid contributing to the spread of malware due to accessing malicious domain names, our experiments were conducted in a virtual environment, and the environment was refreshed frequently.

**7. Related Work**

We summarize previous studies that identified the risks of URL shortening services and analyzed web-based attacks caused by the browsing behaviors of users.

**URL shortening services** URL shortening services are often used in social media such as Twitter because of the character limit for posts. Unfortunately, attackers have taken advantage of the ability to hide destinations to share many malicious URLs. Nepali and Wang [23] proposed an approach to detect malicious short URLs using the content of tweets and properties of accounts. Cao et al. [24] proposed an approach to distinguish organized and organic users on Twitter and found that URL shortening services were used to launch spam campaigns from strategically organized accounts. In addition to the abuse of URL shortening services on social media, Yousaf et al. [25] reported that short URLs are used in the redirection chain generated by traffic exchanges, making it difficult to detect malicious sites. Also, previous studies revealed the users’ perspective on security threats of URL shortening services [26], [27], [28]. Most studies mentioned above analyzed the risk of reaching the malicious destination pages from short URLs but not the risk of being forwarded to malicious sites other than the destination pages. Nikiforakis et al. [2] found that ad-based URL shortening services can cause drive-by download attacks or redirect users to phishing sites when they access intermediate pages from short URLs. However, this study did not consider the security risks resulting from the browser operations of users who reached intermediate pages. We revealed the risks involved
in proceeding to the next web page and discovered a new psychological strategy with a complex page structure to guide users’ behavior.

**Social engineering attacks** Previous studies proposed approaches to analyze web-based social engineering attacks caused by users’ browsing behaviors. Duman et al. [5] implemented TrueClick, a tool to detect fake ad banners that potentially lead to malicious web pages and malware. Although this tool finds buttons that do not link users’ intended pages, ours locates the correct buttons for the users to proceed to the next pages. Also, systems have been proposed to collect malicious web pages by automatically manipulating web browsers [7], [29]. These studies and ours share a common focus on deceptive buttons on web pages. On the other hand, our study is different in that we focus on ad-based URL shortening services and analyze the communications caused by lure buttons that users may misidentify and click, and we reveal the risk of communication with unintended malicious domain names.

8. Conclusion

We analyzed structures of the latest ad-based URL shortening services that encourage users’ clicks and revealed security risks caused by users’ behaviors. We found that even if a user clicks only the genuine button to reach the destination page, there is a risk of reaching malicious domain names that include registration pages for scam services or web pages that ask users to allow services that encourage users’ clicks and revealed security risks.

References


Takashi Koide received his B.S., M.S., and Ph.D. degrees in Informatics from Yokohama National University in 2014, 2016, and 2021. He is currently a researcher at NTT Security (Japan) KK, Tokyo, Japan. His research interests include network and Web security. He won the Research Award from the IEICE Technical Committee on Information and Communication System Security in 2018 and IPSJ Outstanding Paper Award in 2022.

Daiki Chiba received his B.E., M.E., and Ph.D. degrees in computer science from Waseda University, in 2011, 2013, and 2017, respectively. Since 2013, he has been with Nippon Telegraph and Telephone Corporation (NTT), where he has been engaged in research on cybersecurity through data analysis. He is currently a Senior Engineer at NTT Security (Japan) KK, Tokyo, Japan. He is a member of the IEEE and IEICE.

Hiroki Nakano received his B.S. and M.S. degrees in computer science from Yokohama National University, Japan, in 2016 and 2018. Since joining Nippon Telegraph and Telephone Corporation (NTT) in 2018, he has been engaged in research and development on cybersecurity. He is currently an engineer at NTT Security (Japan) KK, Tokyo, Japan, and a Ph.D. candidate at Yokohama National University, Japan.

Mitsuaki Akiyama received his M.E. and Ph.D. degrees in information science from Nara Institute of Science and Technology in 2007 and 2013. Since joining Nippon Telegraph and Telephone Corporation (NTT) in 2007, he has been engaged in research and development on cybersecurity. He is currently a Senior Distinguished Researcher at NTT Social Informatics Laboratories. He received Cybersecurity Encouragement Award of the Minister for Internal Affairs and Communications in 2020, ISOC NDSS 2020 Distinguished Paper Award in 2020, and IPSJ/IEEE Computer Society Young Computer Researcher Award in 2022. His research interests include cybersecurity measurement, offensive security, and usable security and privacy. He is a senior member of IPSJ and a member of IEEE and IEICE.