Analyzing Security Risks of Ad-Based URL Shortening Services Caused by Users' Behaviors

Naoki Fukushi(✉), Takashi Koide, Daiki Chiba, Hiroki Nakano, and Mitsuaki Akiyama

NTT, Tokyo, Japan
{naoki.fukushi.kz,takashi.koide.fk,hiroki.nakano.ma}@hco.ntt.co.jp,
{daiki.chiba,akiyama}@ieee.org

Abstract. URL shortening services make URLs shorter and simpler. 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 are widely used on the Internet to make URLs shorter and simpler, which enables users to bypass character limits when posting to social media or improve the appearance of URLs themselves. 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

J. Garcia-Alfaro et al. (Eds.): SecureComm 2021, LNICST 399, pp. 3–22, 2021.
https://doi.org/10.1007/978-3-030-90022-9_1

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 [20]. That revealed that ad-based URL shortening services display malicious advertisements that automatically forward users to phishing sites or perform drive-by downloads attacks on users. In addition, we found these services have a new strategy for receiving more advertising revenue which is not mentioned in the previous study. For example, these services display many buttons that confuse users as to which 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 they access short URLs. 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 there is a risk of communicating malicious domain names 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.

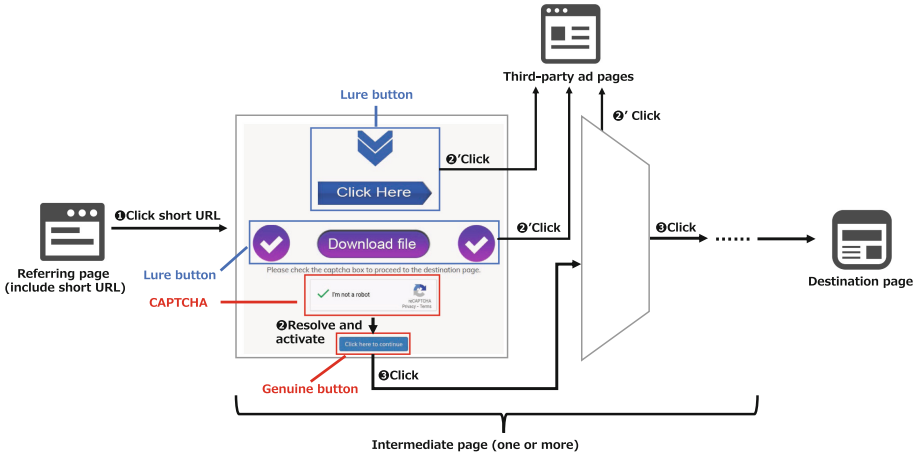

Fig. 1. Overview of structure of ad-based URL shortening services. This figure shows a screenshot of an intermediate page of an ad-based URL shortening service. A user clicks on a short URL included in a referring page and reaches the intermediate page (❶). For this service, the user needs to resolve CAPTCHA to activate the genuine button (❷). If the user mistakenly clicks on the lure buttons, third-party ad pages will be opened (❷'). The number of intermediate pages is one or more and varies with each service. The user clicks on the genuine button and reaches the next intermediate page or destination page (❸).

- We comprehensively investigated the latest ad-based URL shortening services. We are the first to reveal the security risks caused by users' 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 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.

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) (❶). In the case of URL shortening services without ads (e.g., Bitly [3], TinyURL [10]), 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 [20] 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 (❷). 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 (❷'). Note that these ad banners are not unique to ad-based URL shortening services. A previous study [14] reported that popular file-sharing sites display such ad banners as well. Moreover, users may need to go through multiple intermediate pages before reaching destination pages. Therefore, when the user clicks on the genuine button that has been activated, he or she will reach the next intermediate page or destination page (❸). Moreover, users may have to click on the same genuine button several times to reach the next web pages. The actual security risks caused by users' behaviors in such a complex page structure of 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,

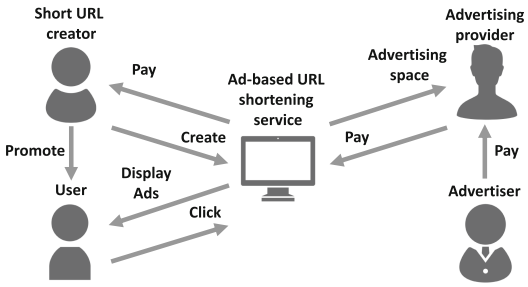

Fig. 2. Stakeholders of ad-based URL shortening services.

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 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 HTML source code changes when the genuine button is activated. Our method is very different from the method proposed in the previous study [14], which uses visual information to identify trick banners that mislead users. To clarify the security risks in detail, we need to monitor

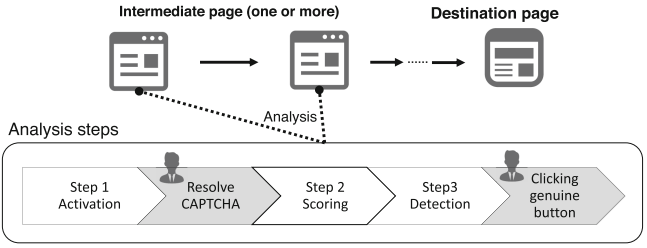

Fig. 3. Overview of steps of detecting genuine buttons and manual operations.

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. This function detects genuine bebuttons 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 after 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 [20]. The purposes of the latter are to avoid crawling by bots or bypassing the services by adblockers. First, this function checks the current HTML source code 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 s before proceeding to step 2, because the genuine button is activated a few seconds after the intermediate page is loaded. Our extension saves HTML source codes after loading intermediate pages, as they 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 HTML source codes. To do this, we designed a scoring

Table 1. List of terms for scoring.

Category	Terms
button	"button", "btn"
click	"click", "continue", "get", "go", "here", "next", "skip", "start", "submit"

algorithm to evaluate web elements by focusing on two features of 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 criteria for terms shown in Table 1. We pre-analyzed the HTML source code of ad-based URL shortening services and collected samples of genuine buttons. Then, using the previous study [16] as a reference, we collected terms that indicate buttons and that encourage users to click. Only elements with scores greater than 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 HTML source codes. 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 `disable` attribute becomes active when this attribute is removed. First, this function compares two source codes (one after 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 [4] 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 repeatedly click the detected genuine button until the transition of web pages or content rewrite occurs. This is because, as explained in Sect. 2, many ad-based URL shortening services require multiple clicks on the same genuine button to go to the next web page. Exceptionally, if an advertisement covers the entire intermediate page and the genuine button is not clickable, we click a close button of the advertisement to dismiss it, because such an advertisement is obviously an obstacle to reaching the destination page.

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 Sect. 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 shortening URL services from the web pages indexed as search results. To select services accessed by many users, we used the Tranco List [22] 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 List ranks for the domain names used by each selected ad-based URL shortening service. We then generated one short

Table 2. Summary of measurement results of ad-based URL shortening services. We abbreviate CAPTCHA as C and Adblock Plus as ABP in the column names.

Service	Sect. 4.1		Sect. 5.1							Sect. 5.2		Sect. 5.3	
	Tranco rank	Novel	Intermediate pages	C	Malicious domain names					Lure buttons	Malicious domains	Ad-block	
					main_frame	sub_frame	script	Others	Total (unique)				
#1	< 2k	✓	1		0	4	17	20	31	0	0		
#2	< 3k	[20]	1		1	8	18	20	32	0	0		
#3	< 6k	✓	3	✓	18	5	56	74	116	0	0	✓	✓
#4	< 9k	✓	3	✓	16	19	60	109	152	0	0	✓	✓
#5	< 13k	[20]	1		0	0	5	2	6	0	0		
#6	< 27k	✓	1	✓	6	7	27	32	53	0	0		
#7	< 33k	✓	3	✓	0	7	33	59	78	0	0	✓	✓
#8	< 38k	✓	2	✓	7	67	75	104	190	0	0	✓	✓
#9	< 39k	✓	2	✓	7	62	70	126	200	0	0	✓	✓
#10	< 43k	✓	2	✓	11	10	50	92	131	2	5		✓
#11	< 46k	[20]	2		6	2	26	39	58	1	6		
#12	< 48k	✓	2	✓	6	19	37	66	106	2	7	✓	✓
#13	< 73k	✓	4	✓	9	7	55	76	112	2	4	✓	✓
#14	< 80k	✓	2	✓	7	11	52	72	113	2	5	✓	✓
#15	< 89k	✓	2	✓	7	10	53	75	112	2	4	✓	✓
#16	< 100k	✓	2	✓	13	14	45	63	104	2	2	✓	✓
#17	< 113k	✓	2	✓	0	1	7	8	12	0	0		
#18	< 133k	✓	2	✓	9	11	35	52	84	0	0	✓	✓
#19	< 182k	✓	2	✓	0	4	12	20	26	8	4		
#20	< 189k	✓	2	✓	6	4	25	28	46	2	2	✓	✓
#21	< 217k	✓	2	✓	5	19	56	81	125	1	7	✓	✓
#22	< 224k	✓	2	✓	6	16	65	111	149	0	0	✓	✓
#23	< 257k	✓	2	✓	0	1	9	12	18	2	1	✓	✓
#24	< 292k	✓	2		7	7	32	34	62	0	0		
#25	< 326k	✓	2	✓	24	10	56	69	109	5	2	✓	✓
#26	< 368k	✓	2	✓	5	1	25	20	37	4	4		
#27	< 426k	✓	3	✓	14	11	49	74	117	6	6	✓	
#28	< 430k	✓	1	✓	5	6	30	32	56	0	0	✓	
#29	< 527k	✓	2		6	5	31	34	61	0	0		
#30	< 865k	✓	3	✓	13	13	47	63	98	3	4	✓	✓

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 *Novel* column of Table 2, we put "✓" for the services that we investigated for the first time and "[20]" for the services that were investigated in the previous study [20]. Of the 30 ad-based URL shortening services we selected for this study, only 3 were investigated in the previous study [20]. Moreover, of the 10 services investigated in the previous study, the remaining 7 were offline as of January 2021.

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 Sect. 3.2 and its result. We evaluate whether the button detected by the proposed method is the correct 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 Sect. 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 Sect. 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 [6] 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 [20] 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.

5 Measurement Results

In this section, we describe the results of our measurement of the ad-based URL shortening services. In Sect. 5.1, we first describe the results of our experiment to reach the destination pages by clicking on only the genuine buttons. In Sect. 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 Sect. 5.3, we finally describe the results of our experiment to access the short URLs with Adblocker enabled.

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 Sect. 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.

Structure of Services. In this section, we reveal the latest structures to the destination pages of the ad-based URL shortening services. We investigate 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. This is a mechanism to increase the time spent on intermediate pages of users and increase their advertisements click rate. Such a structure of ad-based URL shortening services requiring users to go through multiple intermediate pages have 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 pages was 1 for all of these services. Also, ad-based URL shortening services investigated in the previous study [20], indicated by "[20]" 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.

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 [11]. 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 have 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 Sect. 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 ($\approx 2594 \div 30$) 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 Sect. 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

Table 3. Number of malicious domain names detected in each region.

Region	# of malicious domain names
BR	442
HK	419
US	415
GE	407
UK	402
JP	378

Table 4. Number of regions in which each malicious domain name was detected.

# of regions	# of malicious domain names
1 region	225
2 regions	74
3 regions	50
4 regions	32
5 regions	30
6 regions	277
Total	688

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

Region	(1) # of malicious domain names	(2) # of included in top 5 regions	(3) # of missing data	Percentage (2) ÷ {(1) − (3)}
BR	81	30	4	39%
US	38	29	4	85%
UK	33	18	1	56%
HK	31	3	6	12%
GE	30	13	7	57%
JP	12	2	1	18%
Total	225	95	23	47%

the previous study [23]. 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 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 increase.

Region Characteristics. We conducted our experiments from IP addresses in six regions by using a VPN service. By doing this, we could consider that dis-

played 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 largest was 1, accounting for about 73% ($\approx (277 + 225) \div 688$) 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 [9], 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% $\{\approx 95 \div (225 - 23)\}$ 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`. 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 Sect. 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 Sect. 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 Sect. 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 Sect. 5.1, 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 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 [25] is a common tactic for site owners against adblockers. If this is used by ad-based URL shortening services, adblockers are not very useful in reducing the security risk for the user. In this section, we analyze whether adblockers effectively prevent users from reaching malicious domain names. We

accessed short URLs of the 30 ad-based URL shortening services using a web browser with adblockers and investigated whether we can reach the destination pages. We chose two of the leading adblockers (AdBlock [1] and Adblock Plus [2]) 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.

6 Discussion

In this section, we first describe the limitation of the proposed method and our implementation. 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 and implementing the proposed method to analyze ad-based URL shortening services.

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 HTML source code 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. 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 [8], Puppeteer [7]) and human-powered CAPTCHA solving services [18], the proposed method can automate analysis of ad-based URL shortening services containing CAPTCHA. However, due to the previous study [18] 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.

6.2 Recommendations

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

Users of Ad-Based URL Shortening Services. The measurement result of Sect. 5.1 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, Sect. 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 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 Sect. 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 [5] 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 [19] proposed an approach to detect malicious short URLs using the content of tweets and properties of accounts. Cao et al. [13] 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. [24] 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 [12,17,21]. 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. [20] 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. [14] 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 [15,16]. 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

In this study, we revealed the security risks of ad-based URL shortening services 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 malicious web push notifications. We hope that our findings will become a foothold for considering how ad-based URL shortening services should be designed and used.

References

1. AdBlock (2021). https://getadblock.com/
2. AdblockPlus (2021). https://adblockplus.org/
3. Bitly (2021). https://bitly.com/
4. chrome.webRequest (2021). https://developer.chrome.com/docs/extensions/reference/webRequest/#type-ResourceType
5. Misrepresentation (2021). https://support.google.com/adspolicy/answer/6020955?hl=en
6. Number of internet users in selected countries in 2020 (2021). https://www.statista.com/statistics/271411/number-of-internet-users-in-selected-countries/
7. Puppeteer (2021). https://pptr.dev/
8. Selenium (2021). https://www.selenium.dev/
9. SimilarWeb (2021). https://www.similarweb.com/
10. TinyURL (2021). https://tinyurl.com/app
11. VirusTotal (2021). https://www.virustotal.com/
12. Albakry, S., Vaniea, K., Wolters, M.K.: What is this url's destination? empirical evaluation of users' URL reading. In: Bernhaupt, R., et al. (eds.) CHI 2020: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 25–30 April, 2020, pp. 1–12. ACM (2020)
13. Cao, C., Caverlee, J., Lee, K., Ge, H., Chung, J.: Organic or organized? exploring URL sharing behavior. In: Bailey, J., et al. (eds.) Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, 19–23 October, 2015, pp. 513–522. ACM (2015)
14. Duman, S., Onarlioglu, K., Ulusoy, A.O., Robertson, W.K., Kirda, E.: Trueclick: automatically distinguishing trick banners from genuine download links. In: Jr., C.N.P., Hahn, A., Butler, K.R.B., Sherr, M. (eds.) Proceedings of the 30th Annual Computer Security Applications Conference, ACSAC 2014, New Orleans, LA, USA, 8–12 December, 2014, pp. 456–465. ACM (2014)
15. Kharraz, A., Robertson, W.K., Kirda, E.: Surveylance: automatically detecting online survey scams. In: 2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21–23 May 2018, San Francisco, California, USA, pp. 70–86. IEEE Computer Society (2018)

16. Koide, T., Chiba, D., Akiyama, M.: To get lost is to learn the way: Automatically collecting multi-step social engineering attacks on the web. In: Sun, H., Shieh, S., Gu, G., Ateniese, G. (eds.) ASIA CCS 2020: The 15th ACM Asia Conference on Computer and Communications Security, Taipei, Taiwan, 5–9 October 2020, pp. 394–408. ACM (2020)
17. Le-Khac, N.A., Kechadi, T.: Security threats of url shortening: a users perspective. J. Adv. Comput. Networks **3**, 213–219 (2015)
18. Motoyama, M., Levchenko, K., Kanich, C., McCoy, D., Voelker, G.M., Savage, S.: Re: Captchas-understanding captcha-solving services in an economic context. In: 19th USENIX Security Symposium, Washington, DC, USA, 11–13 August, 2010, Proceedings. pp. 435–462. USENIX Association (2010)
19. Nepali, R.K., Wang, Y.: You look suspicious!!: leveraging visible attributes to classify malicious short urls on twitter. In: Bui, T.X., Jr., R.H.S. (eds.) 49th Hawaii International Conference on System Sciences, HICSS 2016, Koloa, HI, USA, 5–8 January, 2016, pp. 2648–2655. IEEE Computer Society (2016)
20. Nikiforakis, N., et al.: Stranger danger: exploring the ecosystem of ad-based URL shortening services. In: Chung, C., Broder, A.Z., Shim, K., Suel, T. (eds.) 23rd International World Wide Web Conference, WWW 2014, Seoul, Republic of Korea, 7–11 April, 2014, pp. 51–62. ACM (2014)
21. Onarlioglu, K., Yilmaz, U.O., Kirda, E., Balzarotti, D.: Insights into user behavior in dealing with internet attacks. In: 19th Annual Network and Distributed System Security Symposium, NDSS 2012, San Diego, California, USA, 5–8 February, 2012. The Internet Society (2012)
22. Pochat, V.L., van Goethem, T., Tajalizadehkhoob, S., Korczynski, M., Joosen, W.: Tranco: a research-oriented top sites ranking hardened against manipulation. In: 26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, 24–27 February, 2019. The Internet Society (2019)
23. Subramani, K., Yuan, X., Setayeshfar, O., Vadrevu, P., Lee, K.H., Perdisci, R.: When push comes to ads: measuring the rise of (malicious) push advertising. In: IMC '20: ACM Internet Measurement Conference, Virtual Event, USA, October 27–29, 2020, pp. 724–737. ACM (2020)
24. Yousaf, S., Iqbal, U., Farooqi, S., Ahmad, R., Shafiq, M.Z., Zaffar, F.: Malware slums: measurement and analysis of malware on traffic exchanges. In: 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2016, Toulouse, France, 28 June–1 July, 2016, pp. 572–582. IEEE Computer Society (2016)
25. Zhu, S., Hu, X., Qian, Z., Shafiq, Z., Yin, H.: Measuring and disrupting anti-adblockers using differential execution analysis. In: 25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, 18–21 February, 2018. The Internet Society (2018)