Deceptive Previews: A Study of the Link Preview Trustworthiness in Social Platforms

Giada Stivala
CISPA Helmholtz Center for Information Security
giada.stivala@cispa.saarland

Giancarlo Pellegrino
CISPA Helmholtz Center for Information Security
gpellegrino@cispa.saarland

***Abstract*—Social media has become a primary mean of content and information sharing, thanks to its speed and simplicity. In this scenario, link previews play the important role of giving a meaningful first glance to users, summarizing the content of the shared webpage within their title, description and image. In our work, we analyzed the preview-rendering process, observing how it is possible to misuse it to obtain benign-looking previews for malicious links. Concrete use-case of this research field is phishing and spam spread, considering targeted attacks in addition to large-scale campaigns.**

We designed a set of experiments for 20 social media platforms including social networks and instant messenger applications and found out how most of the platforms follow their own preview design and format, sometimes providing partial information. Four of these platforms allow preview crafting so as to hide the malicious target even to a tech-savvy user, and we found that it is possible to create misleading previews for the remaining 16 platforms when an attacker can register their own domain. We also observe how 18 social media platforms do not employ active nor passive countermeasures against the spread of known malicious links or software, and that existing cross-checks on malicious URLs can be bypassed through client- and server-side redirections. To conclude, we suggest seven recommendations covering the spectrum of our findings, to improve the overall preview-rendering mechanism and increase users' overall trust in social media platforms.

I. Introduction

The way Internet users access online information has changed dramatically. While not so long ago, users relied on search engines to find new online content, nowadays users predominantly follow links distributed over social media platforms such as social networks and instant messaging to discover web pages. For example, about 40% of web traffic in 2017 originated from social networks [2], against the 37% share of Google searches [2]. When sharing a link, instead of showing the raw URL string, social platforms prepare a user-friendly preview often containing an image, a title, and a description extracted from the shared web page. Link previews play an important role to reach and engage Internet users by providing a meaningful overview of the page content and inviting users to click on them to access more information.

Network and Distributed Systems Security (NDSS) Symposium 2020
23-26 February 2020, San Diego, CA, USA
ISBN 1-891562-61-4
https://dx.doi.org/10.14722/ndss.2020.24252
www.ndss-symposium.org

Unfortunately, the popularity of social platforms has attracted the attention of scammers and other malicious users, who use social platforms to distribute malicious links exposing users to a plethora of security risks ranging from online scams and spam to more concerning risks such as exploitation of 0-day vulnerabilities in mobile devices (see, i.e., [3]). The security risks of visiting malicious web pages have been at the center of the attention of the past decades of research activities, focusing on, for example, detection techniques [13], [20], evaluation of defenses (e.g., [22], [30], [21]), studying the attacker behavior (e.g., [16], [5]), and detection of evasion techniques (e.g., [39], [18]). Only recently, the attention has shifted on studying the extent to which these attacks entered and adapted to social platforms. Existing work has studied different aspects such as the pervasiveness of spam campaigns in social networks (e.g., [32], [15]), the infrastructure used by attackers to distribute malicious pages [29], and the accounts spreading malicious content (e.g., [35], [9]). Other lines of works looked at the demographics of the victims (e.g., [27]), showing that individual and communities behavior influence the likelihood to click.

This paper looks at the problem of malicious link distribution by investigating one of the elements used by attackers to draw the attention of the victims, i.e., link previews. Link previews synthesize the content of a web page, and anecdotal evidence suggests that they are a fundamental piece of information used by users to decide whether to click. For example, in 2017, Facebook forbade users to modify the content of link previews during the creation of posts [28] to contain the creation of deceptive link previews to influence user clicks [28]. This paper puts under the microscope the connection that previews create between users and the actual landing pages, with the overarching goal to provide a new interpretation of the reasons why social platforms' users click on malicious links. Our investigation starts by delivering one of the first characterizations of the process of the link preview creation of 20 popular social media platforms. We provide a comprehensive analysis covering three relevant aspects, i.e., the fields composing link previews, the layout of link previews, the platforms' behavior when fetching the web resources of a preview. Once established a behavioral baseline, we probe social platforms with malicious links to determine deviations from our baseline and characterizing—if any—platforms' defense mechanisms. Finally, starting from the observations collected during our investigation, we show how an attacker can create in practice effective malicious web pages that all our 20 platforms display as benign-looking link previews. In particular, in four of them, i.e., Facebook, Xing, Plurk and

Slack, an attacker can stage such attacks by controlling the content of the web pages only. Finally, we show how to bypass existing countermeasures to avoid the detection of malicious URLs.

This paper makes the following key findings. First, we discovered seemingly innocuous behaviors when creating previews that provide a great advantage to an attacker. For example, on Facebook, an attacker can create a benign-looking link preview of a malicious web page, fooling even experienced and skilled Internet users. Similar attacks are effective against other platforms too, such as Xing, Plurk, and Slack. Second, the vast majority of the tested platforms do not implement any countermeasure that prevents sharing malicious URLs. Only two platforms (Twitter and LinkedIn) implement such countermeasures and, sadly, they are improperly implemented, allowing an attacker to bypass them with redirections. Third, the shortcomings identified by our study are not merely technical issues and are not limited to a few social platforms. Instead, we present a systematic problem affecting all platforms in the ways they design and create previews. Our results show 14 distinct link preview layouts, each with several optional fields. Such a large number of variations may fail to help users in establishing trust with the previewed web sites. As a result, users may overlook security-relevant signals and underestimate the security risks of the previewed page, exposing themselves to a plethora of web attacks. Finally, from the analysis of our results, we distilled a list of *seven* recommendations, ranging from short-term solutions to the technical shortcoming, to the creation of a standard encoding for the content of link previews and the rules to create them.

Contributions — To summarize, this paper makes the following contributions:

- We present the first comprehensive study and characterization of the link preview creation process of 20 popular social media platforms, showing which field is shown under what circumstances;
- We present 14 distinct link preview templates and variants across all platforms, indicating the lack of consensus among all platforms;
- We perform a set of controlled experiments to determine the presence of existing countermeasures on social platforms, showing that all except for two platforms do not implement any defense mechanism. Furthermore, we perform additional tests to determine their effectiveness, discovering that the two countermeasures can be easily bypassed via redirections;
- We test the link preview creation in an adversarial setting, showing that four platforms out of 20 can create benign-looking previews for malicious resources, fooling even experienced and skilled users;
- From our results, we distill 7 recommendations towards more robust and trustworthy link previews;

Organization of the Paper — The content of the paper is structured as follows: Section III presents the general behavior of the social media platforms under test when posting a regular link, creating a baseline for following observations and comparisons. In Section IV we repeat the link submission experiments providing the platforms with malicious content, i.e., blacklisted links or known malware. We present considerations on passive and active countermeasures employed by each social media platform. Section V presents the link preview creation under adversarial influence and presents our attacks. Finally, Section VI presents a set of recommendations and technical solutions.

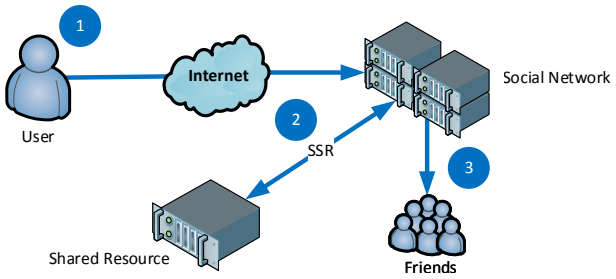


Fig. 1: Sequence of steps when sharing pages on social networks.

II. Background

Before presenting the study of this paper, we introduce building block information. In Section II-A, we start by introducing the general framework used to generate a link preview, and then, in Section II-B, we present the list of social media platforms that we selected for our evaluation. Finally, in Section II-C, we introduce the threat model considered in our analysis.

A. Sharing External Content on Social Media Platforms

Sharing text messages on social platforms, such as social networks, is usually a straightforward process: a user logs into the platform, types the message, and posts it. The message is then stored and delivered to all friends when they update their timeline. When the message contains a URL, the platform retrieves the resources in the shared page to build a link preview. In theory, link previews can be created either by the client-side program (e.g., Javascript) or the server-side programs. However, as URLs often originate from third-party domains, most platforms cannot rely on the client-side programs because the same-origin policy for cross-origin requests (SOP for CORs) prevents the client-side programs from fetching resources from other origins by default. Accordingly, platforms tend to use server-side requests [25] (SSRs).

Figure 1 shows the sequence of steps when sharing URLs on social platforms. The user accesses the social media platform through their browser, or through a mobile app, and then types the URL in the input box to share the URL content with friends or contacts (Step 1 Figure 1). Then, the platform performs a number of SSRs to retrieve the URL and the linked resources, e.g., images (Step 2 Figure 1). Then, the platform processes the collected resources to create a preview for the webpage. The construction of the preview can be aided with a set of additional HTML meta tags specifying suggested content for each field of the preview, such as the page title and page description. Two popular meta tags languages are Open Graph [11] by Facebook and Twitter Cards [37] by Twitter. Table I shows the list of meta tag types that can be

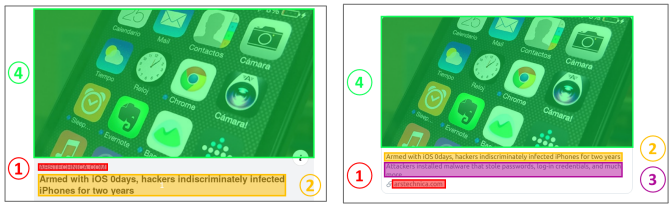
Open Graph	Twitter Cards	Description
`og:title`	`twitter:title`	The title of the article without any branding
`og:description`	`twitter:description`	A brief description of the content.
`og:image`	`twitter:image`	The URL of the image that appears in the preview.
`og:url`	-	The canonical URL for the page, without session variables or user identifying parameters. This URL is used to aggregate likes and shares.

TABLE I: Description of meta tags used to create the link preview of HTML content

```
<meta name="twitter:site" content="①">
<meta name="twitter:title" content="②">
<meta name="twitter:description" content="③">
<meta name="twitter:image" content="④">

<meta property="og:site_name" content="①" />
<meta property="og:title" content="②" />
<meta property="og:description" content="③" />
<meta property="og:image" content="④" />
```

Listing (1) Open Graph and Twitter Cards tags for both Previews



(a) Preview on Facebook (b) Preview on Twitter

Fig. 2: Example of real world use of meta tags.

used to create previews for HTML content. Listing 1 shows an example of meta tag use to create two previews of the same article. Figure 2 shows two screenshots for the resulting previews.

B. Case Studies

We conduct the study of this paper on 20 popular social media platforms, ten of which are social networks, and ten are instant messaging apps. In this section, we present the selection criteria we used.

1) Social Networks: We created an initial list of social networks by combining two sources. First, we manually inspected the Alexa Top 1M domains, retrieved in May 2019, and removed all the websites which do not fall under the *Social Network* category (e.g. `amazon.com`); then, we manually visited the remaining ones until we collected 30 social networks, with no pre-established cutoff on the domain rank value. Then, we merged the 30 social network domains from the Alexa Top 1M domains with additional 30 domains of social networks ranked by the number of users. For this ranking, we used the list maintained by Wikipedia[1], retrieved on July 2019. Then, from these 60 social networks, we removed duplicates obtaining a list of 47 social networks.

We inspected each of the 47 social networks manually, and removed 37 of them for one of the following reasons: (i) social networks that no longer exist, (ii) we were unable to create user accounts[2], (iii) the social network is ranked too low in the Alexa Top 1M, (iv) platforms that do not support link sharing (e.g., Soundcloud), (v) platforms that require Premium subscriptions, (vi) social networks that merged with already discarded ones, and (vii) posting prevented due to bot detection. Table II lists the 10 social networks that we used for the study of this paper.

2) Instant Messaging Apps: We created the list of candidate instant messaging apps by crawling the first 32 apps in order of appearance from the category "Communication" of the Google Play store. To these samples, we added six more apps (i.e., Instagram, Discord, Slack, Kik, Signal, and Snapchat), that we considered popular but not part of the initial list. From these 38 apps, we removed duplicates obtaining a list of 28 instant messaging apps. Then, we inspected each app manually and removed 18 of them for the following reasons: (i) not available in the Apple Store[3], (ii) no instant messaging function, (iii) link previews not supported, and (iv) a low number of downloads. Table II lists the 10 apps we used for the study of this paper.

C. Threat Model

We now present the threat model of this paper. In this paper, we assume the best scenario possible for both the attacker and the victim, i.e., a strong attacker and a tech-savvy user.

Attacker — The attacker of this paper intends to lure their victims into visiting a malicious web page. The specific final attack delivered with the malicious page can vary, based on the motivations of the attacker. For example, an attacker with economic interest may want to steal credit card numbers with a phishing page. In this paper, we also consider highly-motivated powerful attackers such as state-sponsored attackers that can use malicious pages to deliver 0-day exploits to compromise users' device.

The attacker uses social media platforms to distribute the link to the malicious page. For example, in the case of social networks, the attacker can register one or more accounts to direct the campaign. The attacker can also use stolen credentials to spread malicious links over a platform, including instant messaging systems. Their goal is to post malicious links

[1] See, https://en.wikipedia.org/wiki/List_of_social_networking_websites

[2] The main reason was the language barrier. Then, even when using automated translation and help from a native speaker (Chinese), we were deemed to be a robot or a non-trusted user, and denied access to the platform. We would speculate this occurred because our mobile phone numbers were not Chinese or because of the geo-location of our IPs.

[3] We ignored apps that are not in the Apple Store because of our testing setting (See Section III). We used one iPhone device and one Android device: one for the user sharing a link, and the other for the user clicking on the link preview.

Social Network	Alexa
Facebook	3
Twitter	11
VK	15
LinkedIn	23
Pinterest	67
Tumblr	75
Medium	113
Xing	1.294
Plurk	1.341
MeWe	5.142
App	**Downloads**
Instagram	1.000.000.000+
Messenger	1.000.000.000+
Skype	1.000.000.000+
Snapchat	1.000.000.000+
WhatsApp	1.000.000.000+
Line	500.000.000+
Viber	500.000.000+
KakaoTalk	100.000.000+
Telegram	100.000.000+
Slack	10.000.000+

TABLE II: List of platforms

	Visible Features					User Actions		
Name	Site title	Site descr.	Image	Host	Shared URL	Mouse over	Add. Info	**Priority**
Facebook	●	●	●	●	○	DP	●	O, H, T
Twitter	◐	◐	◐	◐	○	SU	○	T, O, H
VK	●	○	●	●	○	DP	○	O||T, H
LinkedIn	●	○	●	●	○	URL	○	O, H, T
Pinterest	○	○	●	●	○	URL	○	O||H, T
Tumblr	●	◐	◐	●	○	DP	○	O, H, T
Medium	●	●	●	●	○	URL	○	O, H, T
Xing	●	◐	◐	●	○	DP	○	O, H, T
Plurk	●	○	◐	○	○	URL	○	O, T, H
MeWe	●	●	◐	●	○	URL	○	O, H, T
Instagram	●	◐	◐	○	●	-	○	O||T||H
Messenger	●	●	●	●	●	-	●	O, H, T
Snapchat	●	○	●	●	○	-	○	O, T||H
WhatsApp	●	◐	◐	●	●	-	○	O, H, T
Skype	●	◐	◐	●	●	-	○	T, O, H
Line	●	●	●	○	●	-	○	O||T, H
Viber	●	○	●	●	○	-	○	T, O, H
KakaoTalk	●	◐	●	●	●	-	○	O, H, T
Telegram	◐	◐	◐	◐	●	-	○	O||T, H
Slack	◐	◐	◐	◐	●	-	○	O, T, H

TABLE III: Characterization of the link preview creation. For the visible features, we use "●" when we observed a field in all of our experiments, "○" when we never observed a field, and "◐" when the presence of the field depends on the context, e.g., meta tags or user edits. We use "DP" for dereferal page, "SU" for shortened URL, and URL for the shared URL. For the priority, we use "O" for Open Graph, "T" for Twitter Cards, and "H" for standard HTML tags.

while, on the one hand, being undetected by possible active or passive detection systems put in place by the hosting platform and, on the other hand, misleading the users, who make use of the link preview to decide whether to click. To this end, the attacker creates a mismatch between the malicious content in the page and its benign-looking link preview, by including in the attacker's code specific meta tags.

Victim — The victim of these attacks can be a specific individual or small group of individuals (i.e., targeted attack), or as many users as possible, indiscriminately. For the analysis of this paper, we consider skilled and experienced social network users—a category of users who is less prone to click on malicious URLs [27], [8], [17], [38].

III. Characterizing Link Preview Creation

In essence, link previews synthesize a web page, creating the expectation on what the user would see when clicking on the preview. The analysis of this section intends to shed some light on the ways social media platforms create link previews. This analysis reviews the content of previews of a set of test web pages, and identifies precisely the fields that are displayed and under which circumstances. After presenting a comprehensive overview of link preview creation, our analysis studies the network traffic to retrieve the resources to build the link preview, looking for distinctive features that can be used to discover social media platforms' requests. Finally, our analysis investigates the extent to which the coherence between previews and web pages content holds.

Experimental Setup: For the analysis of this section, we prepared a set of controlled experiments. Our experiments involve a user submitting links of test web pages we control, and another user observing the created link preview. Accordingly, we registered two user accounts for each platform. Facebook is the only platform offering test accounts, which are users separated from regular users.

We conducted our experiments on social networks using Firefox (version 69.0 for Ubuntu 18.04), Chrome (77.0.3865.75 for Ubuntu 18.04) and Brave Software (0.68.132 based on Chromium 76.0.3809.132 for Ubuntu 18.04) browsers. For IMs, we purchased two mobile phone SIM cards and used two different mobile phones for our experiments, i.e., an iPhone 5S (OS version 12.4.1) and an Android Pixel device (OS version Android 9).

To serve our test pages, we set up an Internet-facing web server serving resources over different subdomains. We used one subdomain for each social media platform and each experiment, achieving a high degree of isolation among the experiments on one platform and across all platforms. Also, we configured our web server to deliver test pages only when accessed via one of the unique subdomains and not through our public IP address, reducing the noise caused by bots of search engines and rogue web scans. All web pages of our experiments contain a unique page title, text paragraphs, and one image. Depending on the specific test, web pages can contain Open Graph and Twitter Cards meta tags in different combinations. We detail the content of meta tags in the corresponding subsection below. Finally, we logged the main fields of the HTTP requests incoming to the server, for further analysis.

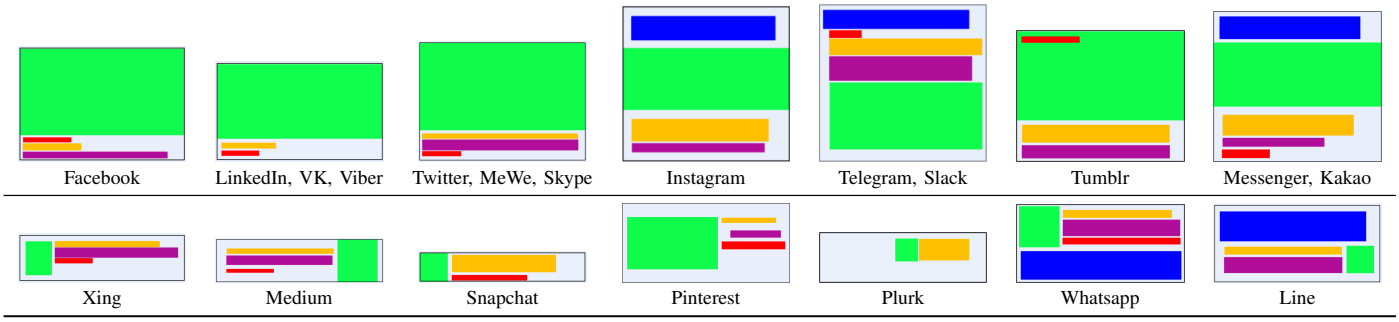


TABLE IV: Color-coded link preview layouts grouped by visual similarity, i.e., same field order and position. Color coding: Red is for the domain name, green for the image, yellow for the site title, purple for the site description, and blue for the URL.

A. Displayed Information

Link previews intend to summarize the content of the embedded links, by showing a site name, an image, and a brief description of the web page's content, typically. These fields originate from the web page's HTML code, either from the standard HTML tags or from ad-hoc meta tags such as Open Graph or Twitter Cards markups. The goal of this section is learning the exact information shown to a user across different social media platforms, and tracing back the content of each preview field to the web page.

To that end, we defined a set of controlled experiments by posting links to resources hosted on our web server, and observing the resulting link preview. As the link preview could show data originating from both standard HTML tags and meta tags, we created web pages with Open Graph or Twitter meta tags, both meta tags at the same time, and no meta tags. When creating our test pages, we used unique values (i.e., titles, descriptions, and images) for each of the meta tags to allow us to identify the exact source of the data values used by the preview creation. Also, we intend to study the ways the link preview may change for pages delivered with redirections. Accordingly, we repeated our experiments using server-side and client-side redirections. Table III summarizes the result of our analysis.

1) Visible Features: We start our analysis by pinpointing the exact fields that social media platforms include in link previews and their location. Table III, columns "Visible Features", lists the displayed fields that we observed. We say that a link preview field is visible ("●") when the field is present in all previews created during our experiments. We say that a link preview field is not visible ("○") when the field is not present in any link preview of our experiments. Finally, we say that a field *may* be visible (symbol "◐") when at least one link preview shows that field. Table IV shows the position of each field per platform.

a) Inconsistent Use and Position of Fields: All platforms include a different combination of the following fields: title of the web page, description of the web page, an image, the domain name, and the shared URL. We observed that there is no regular usage of these fields and that there is no field that is always displayed. The ones that are presented by most of the platforms are the site title (16 over 20 platforms) and the hostname (14 platforms). Then, interestingly, the image field is not shown all the times, and 11 platforms out of 20 may fail in showing an image when, for example, the linked web page does not include the meta tag for images.

When looking at the shared URL field, we observed a noticeable difference between social networks and instant messaging platforms. As opposed to IMs, none of the social networks shows the shared URL in the preview. However, we need to clarify that IMs do not have a dedicated field for the URL. Instead, by default, IMs show the URL in the textbox of the user's message.

Finally, the content of link previews varies with the presence of meta tags. Across all platforms, a total of 25 fields are not present in the link preview when the linked web page does not include any meta tag, i.e., the web page contains only standard HTML tags. Such behavior may be caused by shortcomings of HTML parser, or more probably, by intentional decisions of the developers due to the cost of processing a large number of web pages.

b) Heterogeneous Link Preview Templates: When visiting Table III per platform, one can observe that only nine platforms out of 20 (Facebook, VK, LinkedIn, Pinterest, Medium, Messenger, Snapchat, Line and Viber) create link previews with a consistent number and types of fields, regardless the presence of meta tags. However, when looking at the variety of fields shown across these nine platforms, we observe four different sets of fields indicating that there may not be an accepted consensus on which fields constitute a link preview. For example, the previews created by Facebook and Medium include all fields except for the shared URL, which is instead present in Messenger. VK, LinkedIn, Snapchat and Viber show only site title, image, and hostname, whereas both Pinterest and Line show a different subset of fields each (Pinterest's title and description have to be user-provided at posting time).

Then, the preview created by the remaining eleven platforms varies with the presence of meta tags. Interestingly, the absence of these fields is not consistent within the same platform. Only three platforms (Twitter, Telegram, and Slack) fail to build a preview of pages containing only standard HTML tags. The previews of the other eight platforms incoherently display fields. For example, Instagram shows only the title and the shared URL of pages with only HTML tags.

Finally, when looking at the visual position of field in the preview, we identified 14 distinct template layouts. Table IV lists the layouts we observed, grouping layouts by same order

of fields and position.

2) Priority: The second part of our analysis studies the behavior of the platforms when processing web pages with multiple meta tags and without meta tags. The goal of this analysis is to learn the importance assigned to each field. Table III, columns "Priority", summarizes our findings. We use the letter `H` for standard HTML tags, the letter `O` for the Open Graph meta tags, and the letter `T` for Twitter Cards. The three letters are ordered from left to right by priority. When we cannot establish a clear priority, e.g., the preview contains a mix of tags, we use the symbol "`|`". We cross a letter when the type of tag is never used for the preview.

Our analysis reveals that, with few exceptions, the content of link previews originates predominantly from the meta tags, even when they differ from the content of the page. For example, concerning the hostname field, Facebook, Messenger and WhatsApp show the domain name of the URL of the `og:url` meta tag even when it differs from the URL hosting the resource. We observed similar behavior with Xing, Telegram, and Slack, that show the content of the `og:site_name` meta tag in the host field. A few platforms, i.e., Pinterest and VK, directly prompt the user for text for the preview when the platforms fail at rendering the link preview.

Finally, we observe that Open Graph is, by far, the most used markup language for link previews. Open Graph is also the first one displayed by all platforms except for three, i.e., Twitter, Skype and Viber. While Twitter Cards seems to be rarely used by social networks, it has a bigger userbase among IMs, where only two platforms (Messenger and KakaoTalk) do not seem to support it.

3) User Actions: The third analysis of this section involves fields that a user can inspect only upon an action. We identified two of such fields (see, Table III, columns "User Actions").

The first field is the URL shown when the user moves the mouse over the link preview. Typically, when moving the mouse over an anchor tag, the browser shows in the status bar the hyperlink. Social networks respect such an expected behavior; however, 50% of the social networks do not show the original URL in the status bar but prefer showing either a shortened URL ("SU") or a dereferer page ("DP"). By *dereferer page* we indicate a social network-specific proxy interposed between the user and the shared web page, e.g. as a click aggregator.

The second field is specific only to Facebook and Messenger. Within the link preview, both platforms show an additional UI button—called *"Context Button"*[4]—to display a dialog box with additional information about the domain name of the `og:url` tag. Such additional information, when available, includes (i) content from Wikipedia, (ii) domain name registration date from the WHOIS database, (iii) a link to the Facebook page associated to the domain name, (iv) the number of times that link was shared, and (v) a map showing the locations on earth of users who shared the link.

4) Page redirections: The final analysis of this section studies the link preview generation when pages are delivered with redirections. For that, we repeated the previous experiments by concealing the URL of the final page with a redirections. We implemented both server-side redirections with 303 and 307 status codes, and client-side redirections either via HTML tags or via JavaScript code. The results of our analysis are not in Table III, and we report them in this section briefly. All platforms correctly handle server-side redirections. Facebook is the only platform supporting client-side redirections (both HTML and JavaScript ones). Overall, the link preview does not differ significantly from the previews created when posting direct links.

[4]See, https://www.facebook.com/help/publisher/1004556093058199

	User Agents			IPs		
Name	# UAs	Org.	Bot	# ASN	Res.	Prov.
Facebook	2	1	1	1	0	1
VK	1	0	1	1	0	1
Twitter	1	0	1	1	0	1
LinkedIn	1	0	1	1	0	1
Tumblr	1	0	1	1	0	1
Pinterest	2	1	1	1	0	1
Xing	3	0	3	2	0	2
MeWe	1	0	1	1	0	1
Plurk	1	0	1	1	0	1
Medium	5	0	5	2	0	2
Instagram	12	9	3	1	0	1
Messenger	6	3	3	1	0	1
Skype	2	1	1	1	0	1
Snapchat	3	0	3	2	1	1
WhatsApp	2	0	2	1	1	0
Line	3	2	1	2	0	2
Viber	1	0	1	1	1	0
KakaoTalk	2	1	1	2	0	2
Telegram	1	0	1	1	0	1
Slack	3	0	3	2	0	2

TABLE V: Analysis of access logs considering IP and User-Agent for each social media platform

B. Network Signatures

After analyzing the displayed information, we look for unique signatures in the incoming HTTP requests. Our goal is to identify distinguishing features that can be used by the owner of a web page to determine when the incoming request originates from a social media platform. For this analysis, we process the entries in our server log files to identify such signatures.

In general, when sharing URLs to our pages on social networks, we should expect that other users may click on the link previews, introducing spurious entries in our logs. To avoid the presence of user activities, we limited the visibility of our posts whenever a platform supports such a feature. Only two platforms do not support access restrictions, i.e., Medium and Plurk; however, upon manual inspection, we verified our logs did not contain any user activity but only requests from both platforms. Finally, we point out that the same concern does not apply for IMs as messages are visible only to the recipient, that, in our setting, is another user under our control.

From our log files, we parsed all entries and extracted the user-agent strings and the IPs. We compared user-agent strings against known strings for browsers, and we looked for substrings that can be used to identify a platform uniquely. An example of these substrings is

"facebookexternalhit" for Facebook or "vkShare; +http://vk.com/dev/Share" for VK. When the user agent contains such unique strings, we classify the entry as bot. When the user-agent string matches one of the known user-agent strings of browsers, we classify the entry as organic. Then, starting from the collected IPs, we resolved the autonomous system numbers (ASNs) and searched the AS name strings for unique substrings. For example, Facebook's request originate from AS 32934, whose name is "Facebook, Inc.". However, not all platforms manage an autonomous system, but they may be relying on third-party providers. For example, Pinterest's requests originate from AS 14618, whose name is "Amazon AES". When the autonomous system name matches the name of a platform or a known network provider, we classify the entry as a service provider.

Table V summarizes the results of our analysis. All the 20 social media platforms under test use at least one user agent string linked to the name of the company or the service itself, allowing for immediate traffic filtering. Of these, 13 platforms use only one user-agent header, and seven platforms (Xing, Medium, Instagram, Messenger, Snapchat, Whatsapp and Slack) use multiple ones. Seven platforms (Facebook, Pinterest, Instagram, Messenger, Skype, Line, and KakaoTalk) request web pages using user-agent strings that are indistinguishable from browsers, posing a potential problem for the identification. However, the analysis of the IPs and the ASes provides a stronger signal than user-agents. As a matter of fact, all platforms perform HTTP requests from IPs of either one or two autonomous systems that can be linked to the platforms. Three instant messaging apps (Whatsapp, Snapchat, Viber) request resources directly from the user's phone, slightly increasing the difficulty in distinguishing if the visitor is organic or not, as the AS usually is from a residential area; nonetheless, all three of them include the app name in the user-agent string, so we can categorize the respective entries as bots.

C. Link Preview Coherence

The final analysis of this section investigates the coherence between the link preview and the web page. In particular, we are interested in studying the ways social media platforms keep up to date the link previews in which a page changes over time. To this end, we generated new, unique URLs, one for each platform, and posted them. Then, we developed a bot controlling a pool of web browsers which is visiting periodically (every 30m) the platforms' pages showing the preview, over a period of 14 days. As IMs messages are expected to be short lived, we did not consider them for these experiments.

The analysis of our logs revealed that eight out of 10 social networks request the page at least once on the submission date, and never again. Twitter and Pinterest are two exceptions, requesting the web page multiple times across a period of 14 days. For what concerns the associated resources, seven social networks requested them only once at submission time, and never again. The remaining three platforms, i.e., Facebook, Twitter and LinkedIn, request the link preview images more regularly.

D. Takeaway

The analysis of this section shed some light on three key aspects of social media platforms when creating a link preview. To summarize, this section makes the following findings:

- Social media platforms rely unconditionally on meta tags for rendering previews, especially on the Open Graph markup language. When meta tags are not present, link previews display fields in an inconsistent manner, exposing users to a great variety of heterogeneous link preview templates. As a result of all this, we speculate that users are misled into taking the wrong security decision. Also, the heterogeneity of templates and inconsistent use of fields may fail in building a secure mental model of link preview outlooks.
- Platforms' requests contain distinguishable signatures that can be used by web sites owners to determine when a request originates from social media platforms. This is a required feature to enable cloaking attacks.
- The temporal analysis reveals that platforms tend to fetch the resources for the link preview very rarely over a period of 14 days. A longer time window may show a different behavior, however, it should be noted that 14 days is sufficient for a successful malicious campaign.

IV. Malicious Content and User Awareness

Section III studied the behavior of social media platforms when sharing links to benign web content. However, as observed by prior work, adversaries can also share malicious content on social media platforms such as phishing pages (see, e.g., [29], [32]). Anecdotal evidence suggests that social media platforms, social networks in particular, may have deployed defenses to counter the spread of malicious content in their systems. For example, Twitter claims to match shared links against a database of potentially harmful URLs [36] and to additionally use their shortening service to interpose informative safeguarding pages in between https://t.co links and their malicious targets. Facebook reports the employment of dedicated teams and tools against spam on the platform [12], as well as anti-virus measures in the file upload and download processes [10].

The second analysis of this paper studies the presence and effectiveness of possible deployed countermeasures when sharing malicious URLs. Also, our analysis reviews the created link previews to evaluate to what extent users may be aware of the risk of clicking on previews of malicious links. In this section, we leverage on the knowledge acquired during the observations of Section III, which we will use as a behavioral baseline to compare social media platforms behavior when dealing with malicious content. Our focus is not built on the attacker's perspective, rather on the observation of existing active or passive countermeasures preventing the distribution of malicious content; the most fitting scenario is the one of malware and phishing spread prevention.

Experimental Setup: The experiments of this section involve sharing links to two types of malicious content to check for the presence of different countermeasures. First,

Test	Sharing Type: Resource	Observ.	Social Networks: Facebook	Twitter	Vk	LinkedIn	Pinterest	Tumblr	Medium	Xing	Plurk	MeWe	Instant Messengers: Instagram	Messenger	Snapchat	WhatsApp	Skype	Line	Viber	KakaoTalk	Telegram	Slack
Direct	Virut/EICAR	Posted	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-
		Preview	●	○	◑	○	◑	◑	○	●	○	◑	-	-	-	-	-	-	-	-	-	-
	Blacklisted URL	Posted	●	×	●	●	●	●	-	●	-	●	●	●	●	●	●	●	●	●	●	●
		Preview	●	○	●	×	●	◑	-	●	-	●	●	●	●	●	●	●	●	●	○	○
Client Red.	Virut/EICAR	Posted	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	
		Preview	●	●	●	○	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	
	Blacklisted URL	Posted	●	●	●	●	●	●	-	●	-	●	●	●	●	●	●	●	●	●	●	●
		Preview	●	●	●	●	●	●	-	●	-	●	●	●	●	●	●	●	●	●	●	●
Server Red.	Virut/EICAR	Posted	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-
		Preview	●	○	◑	○	◑	◑	○	●	○	◑	-	-	-	-	-	-	-	-	-	-
	Blacklisted URL	Posted	●	×	●	●	●	●	-	●	-	●	●	●	●	●	●	●	●	●	●	●
		Preview	●	○	●	○	◑	◑	-	●	-	◑	●	●	●	●	●	●	●	●	○	○

TABLE VI: Test results when sharing a malware and a blacklisted URL.

we want to test platforms against the presence of URL filtering mechanisms. For example, a social network may check whether the shared URL is flagged as malicious by existing URL blacklists, e.g., Google SafeBrowsing [14]. Accordingly, we searched for URLs on PhishTank [23] and verified that the URLs are also blacklisted by Google SafeBrowsing [14]. We used a total of three different blacklisted URLs across platforms, all with the same characteristics, due to their short uptime before being deactivated. Second, we want to check whether platforms proactively scan the content of web pages for malicious content. To this end, we created unique links to our server to download the trojan Win32.Virut. For IMs, we did not perform such an experiment as downloading mobile apps through a browser is not a major attack vector.

When running our tests, we also monitored the exact point where we can observe the effects of any countermeasures. In our analysis, we considered two points: when posting the URL, and when creating the link preview. Table VI shows the result of our analysis.

A. URL Posting

The first aspect that we monitored during the execution of our experiments is whether the platform accepts malicious URLs. Only Twitter detected the blacklisted URL as malicious and prevented posting altogether. Also, Twitter showed a warning message: *This request looks like it might be automated. To protect our users from spam and other malicious activity, we can't complete this action right now. Please try again later*. All other platforms did not show any error or warning messages and created a URL preview instead.

B. Preview Creation

Social media platforms can detect malicious URLs in later stages of the URL processing pipeline, e.g., when fetching the resources. However, our analysis revealed that the vast majority of platforms do not seem to implement any security check.

a) Malware: When sharing the malware program, all platforms correctly retrieved the binary from our server. However, as the binary program does not contain HTML code, platforms tend to render a bare-minimum link preview (i.e., Facebook, Xing), possibly prompting the user to provide more information (i.e., VK, Pinterest, Tumblr, and MeWe) or render no preview at all (i.e., Twitter, LinkedIn, Medium, and Plurk). Also, all platforms did not show any error message or warning, and, clicking on the link preview results in downloading the malware program.

b) Blacklisted URL: When sharing a blacklisted URL, only one platform, i.e., LinkedIn, detected the malicious URL after posting. Here, LinkedIn modified the text of the link to point to a redirector page (`linkedin.com/redir/phishing-page?url=`URL). When a user clicks on the preview, Linkedin shows an informative page explaining that the site was blacklisted according to Google Safe Browsing, thus blocking access to the target URL. In spite of repeated attempts, the user account was not deactivated.

Sixteen social media platforms over 18 treated the blacklisted links as regular links: their bots visit the page and render a preview based on the specified meta tags (if any) or fall back to parsing HTML, when possible. Eight social media platforms (Facebook, VK, MeWe for SNs and Messenger, Snapchat, Line, Viber, KakaoTalk for IMs) created a rich preview with no distinguishable difference from a regular innocuous link. The remaining eight platforms either showed partial information (page title and host, but no image and no description) or did not render a preview at all, due to their implementation.

C. Takeaway

The analysis of this section intends to investigate the presence of possible mechanisms to prevent the distribution of malicious URLs on social media platforms. To summarize, our analysis makes the following findings:

- In general, our experiments could not find evidence of widespread use of countermeasures to prevent the

distribution of malicious content at submission time.

- All platforms—except for Twitter and LinkedIn—do not show specific warnings or error messages to the users, indicating potential danger when clicking on the previews. Also, link previews for blacklisted URLs can contain the same semantic elements that are typical of previews of benign web pages, i.e., title, description, a picture, and the domain name.
- Two out of 20 social media platforms perform security checks on the posted URL. For example, LinkedIn uses the Google Safe Browsing API to detect malicious URLs. While Twitter forbids posting blacklisted URLs, LinkedIn accepts the URLs, but it replaces the URL in the preview with a link to an own warning page.
- Twitter and LinkedIn are the only two platforms implementing a form of defense. However, we could bypass these defenses by using server- and client-side redirections.

V. Attacks

So far, we studied the behaviors of social media platforms when processing both benign and malicious webpages, and we learned the various ways platforms could create link previews and validate URLs. This section will take a look at the link preview creation from an adversarial point of view. Here, we consider an attacker who intends to lure one or more users to visit a malicious webpage that is distributed over social media platforms. To do so, the attacker needs to hide their malicious intent by using, ideally, a benign-looking link preview. At the same time, as platforms may be validating URLs against blacklists, the attacker needs to avoid the detection of malicious URLs. In this section, we consider both problems. First, in Section V-A, we present a set of shortcomings of social media platforms that allow attackers with different capabilities to craft arbitrary link previews, regardless of the actual content or purpose of the shared page. Then, in Section V-B, we show how an attacker can bypass URL validation countermeasures.

We summarize our attacks in Table VII. Overall, our results show that all platforms are vulnerable to our attacks—except for two (Plurk and Medium) that we did not test with malicious URLs as they cannot limit the visibility of posts. Four platforms, i.e., Facebook, Xing, Plurk, and Slack, can be attacked by attackers who control the content of a webpage only. The remaining platforms are vulnerable to attackers who can also register domain names for the server distributing malicious pages.

A. Adversarial Analysis of the Link Previews Creation

The goal consists in creating a malicious web page whose preview, when shared on social media platforms, is similar to the preview of a benign webpage, requiring an attacker to be able to replace the content of each field with ones of their choice. In this section, we study the extent to which an attacker can arbitrarily influence the link preview creation considering two attackers with different capabilities, i.e., a first one that controls the content of a web page and an another one that can also register domain names. Table VII shows the results of our analysis.

	Crafted Fields					Bypass		Attacker	
Name	Site title	Site descr.	Image	Host	Shared URL	Client Red	Server Red.	Blacklisted URL	Capability
Facebook	◆	◆	◆	◆	-	-	-	✓	Page cnt.
Twitter	◆	◆	◆	◇	-	✓	-	✓	Domain
VK	◆	-	◆	◇	-	-	-	✓	Domain
LinkedIn	◆	-	◆	◇	-	-	✓	✓	Domain
Pinterest	-	-	◆	◇	-	-	-	✓	Domain
Tumblr	◆	◆	◆	◇	-	-	-	✓	Domain
Medium	◆	◆	◆	◇	-	-	-	-	Domain
Xing	◆	◆	◆	◆	-	-	-	✓	Page cnt.
Plurk	◆	-	◆	-	-	-	-	-	Page cnt.
MeWe	◆	◆	◆	◇	-	-	-	✓	Domain
Instagram	◆	◆	◆	-	◇	-	-	✓	Domain
Messenger	◆	◆	◆	◆	◇	-	-	✓	Domain
Snapchat	◆	-	◆	◇	-	-	-	✓	Domain
WhatsApp	◆	◆	◆	◆	◇	-	-	✓	Domain
Skype	◆	◆	◆	◇	-	-	-	✓	Domain
Line	◆	◆	◆	-	◇	-	-	✓	Domain
Viber	◆	-	◆	◇	-	-	-	✓	Domain
KakaoTalk	◆	◆	◆	◇	◇	-	-	✓	Domain
Telegram	◆	◆	◆	◆	◇	-	-	✓	Domain
Slack	◆	◆	◆	◆	◆	-	-	✓	Page cnt.

TABLE VII: Summary of the evaluation of our attacks. We use "◆" when the attacker can change a field via HTML tags. We use "◇" when the attacker can replace the value of a field via the domain name of the malicious URL. We use "✓" when a bypass technique and attack succeeded. Finally, we use "-" when the field is not present or when we did not test the platform.

1) Crafting Fields: We evaluate the replacement of the preview fields considering two types of attacker models. The first one is a person that can create malicious web pages and upload them on a web server. This setting intends to model the common scenario where the attacker exploits vulnerabilities in existing servers or web applications to upload malicious content such as phishing pages. Since this attacker controls the web page content, they can modify the title, the description, and the images with ones of their choice. Here, the attacker can store the selected values in the meta tags or the standard HTML tags. In Table VII, we mark these field with "◆". However, such an attacker may not be able to alter the content of the domain name and the shared URL.

The second type of attacker possesses the capabilities of the previous attacker and extends them with the ability to register domain names. This scenario intends to model the typical attacker that registers fraudulent domain names to support their malicious activities. Being able to register domain names extends the abilities of the previous attacker as it allows for crafting the domain name and shared URL too.

In the remainder, we present our analysis, discussing in detail what an attacker could do to change the content of these two fields. We grouped our results in five distinct classes based on the observed behaviors:

a) Link Previews without Domain Name: One platform, i.e., Plurk, does not include any information regarding the

landing page URL, i.e., neither the domain name nor the original URL. In this case, the creation of a crafted link preview is straightforward. An example of preview for Plurk is Figure 3b.

Instagram and Line do not show the domain name either. However, we point out that they show the original URL. In our experiments, we could not find a way to remove or replace the string of the shared URL from the preview. Accordingly, it can be changed only by an attacker who has full control of the URL string.

b) Replacing Domain Name using `og:url`*:* In Facebook, we observed that when the URL of the shared webpage mismatches the `og:url` meta tag, the preview fields title, image, description and host are retrieved from the webpage hosted at the URL specified in the `og:url` meta tag rather than in the shared one. Nonetheless, the final landing page remains the URL of the shared web page. In this case, the attacker can assign to the tag `og:url` a URL of a benign resource, resulting in a preview that is entirely indistinguishable from a benign preview. Figure 3c shows such a benign-looking preview. The Messenger app shows the same behavior, but the attacker cannot remove the shared URL from the message text; due to the mismatch between the shared URL and the preview, we say that this attack is possible only for an attacker that can register domain names.

WhatsApp replaces the content of the host field only, showing the URL specified in the `og:url` meta tag. Also in this case, the shared URL cannot be removed from the message text, requiring the attacker to register a new domain name for this purpose.

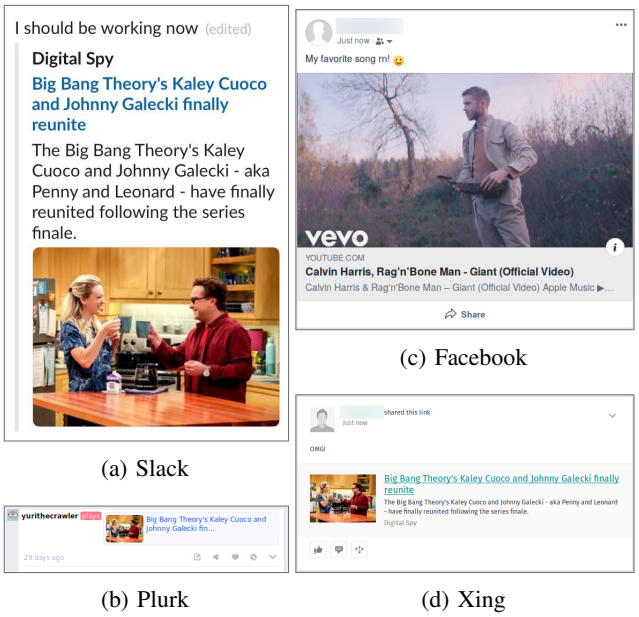
c) Removing Shared URLs in IMs: One IM platform, i.e., Slack, permits the editing of the content of sent messages. We verified that a user could edit the URL string of a message too, after the creation of the preview, effectively eliminating this field from the rendered preview. The platforms Snapchat, Skype and Viber remove the URL from the message text after posting, although we observe that they include the domain name in the preview, which is extracted directly from the shared URL. We could not find a way to replace the domain name with an arbitrary string. Therefore, this attack may not be successful for an attacker controlling the webpage content only.

d) Replacing Domain using `og:site_name`*:* During our experiments, we discovered that three platforms, i.e., Xing, Telegram and Slack, replace the domain name with the content of the `og:site_name` meta tag.

As mentioned before, Slack allows removing the shared URL from the message text after posting the link. Accordingly, an attacker can generate a preview that looks like a benign one only by controlling the HTML code of the page. Figure 3a shows an example of such a link preview.

Xing does not include the original URL; therefore, controlling the web page content is sufficient to craft a URL preview where the domain name is replaced with the site name. Figure 3d shows such a link preview.

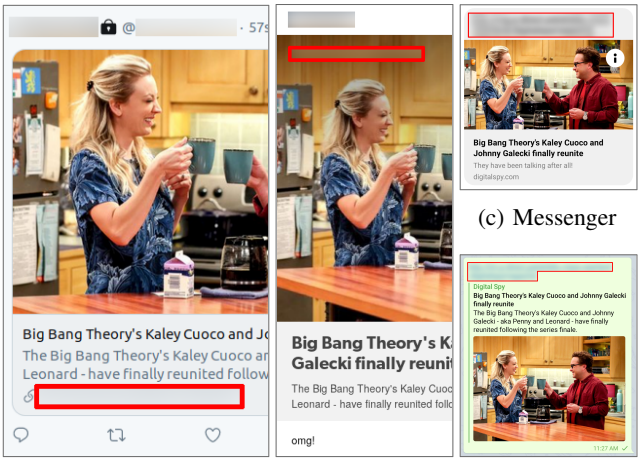
Then, replacing the domain name of Telegram's preview with the `og:site_name` meta tag may not be sufficient as



(a) Slack (c) Facebook

(b) Plurk (d) Xing

Fig. 3: Examples of maliciously-crafted previews by an attacker who controls the content of a webpage. In all these examples, the shared page does not include the text in description, nor the displayed image.



(a) Twitter (b) Tumblr (c) Messenger (d) Telegram

Fig. 4: Examples of crafted previews that always show the domain name. The red box shows the position of the domain name.

Telegram includes the shared URL that we could not remove. Accordingly, the creation of a Telegram's preview is more suitable for an attacker that can register domain names.

2) Attacks: To summarize, our analysis shows that it is possible to create an attack against each platform. Our attacks can create entirely indistinguishable link previews against four platforms, i.e., Facebook, Xing, Plurk, and Slack, by changing only the content of the malicious web page. In three cases, the attacker needs to exploit seemingly innocuous behaviors of

the platforms to achieve their goal. For example, on Facebook, the attacker can replace the domain name with the domain of `og:url` meta tag, whereas for Xing and Slack, the attacker can replace the domain name by using the `og:site_name` tag. As Slack includes the shared URL too, the attacker can also remove the original URL from the preview after its creation. We point out that, in all these four cases, even when the attacker replaces or hides the domain names and the shared URLs, the landing pages, i.e., the malicious pages, of the link preview remain unchanged. When the attacker controls the domain name, then the remaining platforms can be targeted as well. Figure 4 show examples of partially crafted link previews. The areas in red contain either the domain name or the original URL. Finally, the evaluation for two platforms, i.e., Medium and Plurk, was limited to the generation of the previews. On these two platforms, we did not share any malicious URLs as they cannot restrict the visibility of the shared content.

B. Bypassing Countermeasures

When sharing malicious content, social media platforms may detect the maliciousness of the shared web page. As shown in Section IV, only two platforms can detect when a URL is known to be distributing malware by using, e.g., Google Safe Browsing [14]. In this section, we focus on these two platforms and show that, despite the efforts of validating URLs, it is possible to bypass these controls by creating ad-hoc web pages. In this page, we consider two approaches that are based on the findings of Sections III-A4 and III-B.

1) Redirections: During our experiments of Section III-A4, we observed that all platforms except for one (Facebook) do not support HTTP redirections. As a result, those platforms may not be able to determine the next URL in the redirection chain, and accordingly, they should fail in verifying whether the URL is malicious. We tested our hypothesis and confirmed that client-side redirections could effectively bypass both Twitter and LinkedIn URL validation. The evaluation with redirections is summarized in Table VI.

However, interestingly, we also found out that it is possible to bypass the URL filtering of LinkedIn with a server-side redirection, i.e., 30x response. Here, we suspect that LinkedIn does not validate the `Location` header of the HTTP response sent by the redirector.

2) Link Cloaking: As a final step, an attacker may resort to cloaking attacks. The analysis Section III-B showed that the source IP and the user agent strings of the social media platforms are unique, and an attacker can leverage on these features to change the behavior of the servers selectively. For example, when the incoming request matches one of the known signatures, the server will deliver the benign web page for link preview creation. Otherwise, the server delivers the malicious web page.

VI. Discussion and Recommendations

In this section, we discuss our results and distill a set of recommendations for social media platforms towards the creation of more reliable link previews.

A. Variety of Layouts and Processing Rules Can Lead to Underestimate the Risk

Our results show a great variety of layouts used by the platforms under evaluation. We distinguished 14 distinct templates for link previews. Also, we observed that the same platform could create many variants of the same template, for example, by removing or replacing fields.

The variety that we observed suggests that there is no general consensus on (i) which fields constitute a link preview, (ii) under which circumstances fields are displayed, and (iii) the processing rules and priority. The lack of consensus can have a dramatic impact on the way users evaluate the trustworthiness of a preview. As users can be exposed to different layouts, they may neglect the importance of a field, underestimating the overall security risks of a link.

(R1) Standardize Content and Construction Rules of Link Previews: Our first recommendation is to define and agree on the content of link previews, and the exact rules to construct them.

B. Distrustful Scenario

The scenario in which social platforms operate is characterized by distrust. On the one hand, social platforms cannot verify the truthfulness of webpages content. For example, they cannot decide whether an image or a title is appropriate for a given page. Accordingly, social platforms cannot trust webpages. On the other hand, users can leverage on their own experiences and skills to navigate the web and inspect both URLs and the circumstances that led them to see those URLs looking for warning signals, indicating that pages may be dangerous. Experienced users may be trusting webpages they are familiar with, e.g., their web email provider; however, in the general case, they will not trust any page.

In a scenario with these trust relationships, social media platforms act as intermediaries between web pages and users, providing to the latter syntheses of the former. In playing such a role, social platforms should avoid introducing interpretations of the content of the webpages or using processing rules that can hide or distort the preview of the page. Also, social platforms should enforce the presence of security-relevant fields that users can use to decide whether to click, i.e., domain names, and original URLs. While most of the social platforms under test include a domain name or the original URL, four of them, i.e., Facebook, Xing, Plurk, and Slack do not satisfy such a requirement. From the analysis of these four platforms, we derive the following recommendations:

(R2) Show Domain or URL: As reported in Table III and further detailed in Section V-A1a, the link preview created by the social network Plurk does not include any host field, and there is no URL in the post text. As this information is significant in assessing the trustworthiness of the link preview, we include as part of our recommendations that link previews must include either the domain name or the shared URL. Among the platforms under evaluation, only Plurk does not comply with our recommendation.

(R3) Limit Edits of Posts or Refresh Previews: Platforms may want to allow users to edit previous posts. In these cases, they should forbid changing the shared URLs. Alternatively,

```
<head>
 <title>HTML title</title>
 <meta property="og:site_name" content="①">
 <meta property="og:title" content="②" />
 <meta property="og:description" content="③">
 <meta property="og:image" content="④"/>

 <meta name="twitter:title" content="②">
 <meta name="twitter:description" content="③">
 <meta name="twitter:image" content="④">
</head>
<body>
 <!-- Malicious content -->
</body>
```

Listing (2) Example of Malicious Page Shared on Slack

(a) Rendered Preview

Fig. 5: Example of Malicious Link Preview.

when changing the URL is admitted, platforms should re-build the link preview and replace the old preview with the new one. In our experiments, and in particular in Section V-A1c, we observed that Slack allows users to remove URLs from previous messages without updating the link preview. This feature can be misused as shown in Figure 3a, especially if the domain name field contains an arbitrary string rather than the actual domain or URL.

(R4) Create Preview Without Retrieving Referred Pages: Platforms should create link previews using data items contained in the code of the landing page. When the landing page contains external links such as `og:url`, platforms could consider such resources as long as they are in the same domain as the landing page. Furthermore, platforms should not use such URLs to build the entire preview. In Section V-A1b, we observed that Facebook creates the entire preview by using the content of the URL in the `og:url` tag, and an attacker can hide a malicious webpage by creating a link preview with a YouTube video using only the `og:url` meta tag (see, Figure 5a).

(R5) Type Fields: In Section III-A2 we observed that, in a few social platforms, it is possible to override the content of the domain name field by adding the `og:site_name` meta tag. When the platform additionally does not include the shared URL in the text field of the post, as observed in Section V-A1d for the social network Xing, the final link preview contains no trusted information on the URL, as the domain field can contain an arbitrary string. Therefore, we recommend that each field of a link preview should have a well-defined type, e.g., image, description, title, domain, and URL. Then, when creating a preview, platforms should not use the content of a field of type t_1 to fill a field of a different type t_2.

C. Upstream vs Downstream URL Validation

During the lifetime of a link preview, there are different points in time when malicious links can be detected, e.g., when platforms accept the URL and when users click on the preview. In the remaining, we discuss where and how such a check should be enforced.

(R6) Do Upstream URL Validation: When testing social media platforms against phishing URLs, we observed that not all browsers could show Google Safe Browsing warning messages before loading malicious URLs. In particular, we verified that the in-app browsers as used per default configuration by Messenger, Slack, Telegram, Line, Instagram, and WhatsApp (both on Android and iOS) do not show any warning when loading our phishing URLs. Also, we verified that external browser apps might not reliably show Safe Browsing warnings. We reproduced such behavior on Chrome Browser 76.0.3809.123 for iOS 12.4.1, Chrome for Android (Android 9,Pixel Build/PQ3A.190801.002 and Pixel 2 Build/PQ3A.190801.002), Safari (12.1.2 Mobile), Brave Browser for Android (1.3.2 based on Chromium 76.0.3809.132), and Firefox Focus for Android (8.0.16). Only one mobile browser, i.e., Firefox for Android (68.1), showed the warning correctly. We point out that we used the default configuration of both all tested apps and the operating systems. Finally, desktop browsers were more consistent than the mobile ones in showing the warning. Here, we tested Chrome Browser (77.0.3865.75 for Ubuntu 18.04), Brave Software Browser (0.68.132 based on Chromium 76.0.3809.132 for Ubuntu 18.04), and Firefox (69.0 for Ubuntu 18.04). Independent non-academic research confirmed the presence of a discrepancy between Google Safe Browsing mobile and desktop. See, for example, [26], [19].

The reasons for such a discrepancy are not fully understood, and further research is required. Nevertheless, such results indicate that browsers may fail to or will not detect malicious URLs, and, accordingly, browser-side countermeasures should not be considered as a bulletproof last line of defense. Based on that, we recommend developers to implement upstream URL validation during the generation of link previews. Among the 20 platforms we verified, only two implement such a mechanism.

(R7) Do Proper URL Validation: An HTTP agent can reach web resources by following chains of redirections. While in the past redirections were only implemented via HTTP response codes and the refresh HTML meta tag, nowadays redirections are also implemented via JavaScript code. When validating URLs, it is fundamental that all URLs of a redirection chain are validated as well. Unfortunately, the only two platforms implementing a form of URL validation (Twitter and LinkedIn) did not validate URLs during redirections, allowing attackers

to bypass their countermeasures. Table VI sums up the results of our experiments with these two social networks.

D. Ethical Considerations

Our experiments raise the valid ethical concern of sharing malicious content on social media platforms. For example, users not aware of our experiments may click on our previews and become victim of an attack. To avoid attacking users, we limited the visibility of the shared malicious links of the platform accounts we control. When the platform did not support limiting the post visibility, i.e. for the social networks Medium and Plurk, we did not share the phishing link and, instead of distributing the Win32.Virut malware, we used the innocuous EICAR test file, used to test antivirus software.

The second concern of our experiments is sharing malware from our servers. The main risk of these experiments is that both the network and the domain name of our institute may be blacklisted, affecting the work of the research and administration staff. To avoid such a risk, we registered a first-level domain name and moved our servers on Amazon Web Service EC2.

VII. Related Work

In this section, we review works related to our study. First, we present relevant works in the area of the analysis of malicious URLs in social networks. Then, we related our work with the research done in the area of phishing.

A. Analysis of Clicks on Social Platforms

When deciding whether to click on link previews, users rely on an ensemble of signals that are displayed by the social platform's web pages. For example, Redmiles et al. [27] show that users take into account who shares the web content and the online community the content originates from. Similarly to Redmiles et al. [27], our work intends to shed some light on the dynamics behind user clicks on social networks. However, as opposed to Redmiles et al. [27], our work does not study social connections between users or user properties such as demographics. Instead, our work focuses on the content of a link preview, the trustworthiness of the link preview creation, and it explores the extent to which an attacker can control the fields displayed to the victims.

Clicking on maliciously-crafted link previews is a security concern that Facebook tackled in 2017 by forbidding users to modify link previews from the web site [28]. Also, an independent work by Barak Tawily [33] showed that Facebook link previews can be modified via metatags. Our study expands the one by Tawily [33] and shows that motivated attackers can still control the content of a preview by crafting ad-hoc HTML tags of the shared pages. Also, our study shows that the problem is not affecting only Facebook, but it is a systematic problem affecting most of the social platforms that we evaluated.

B. Phishing in Social Networks

A typical phishing attack involves an attacker, their victim, and a malicious resource used as a bait, to convince the user to provide sensitive information. To this end, attackers usually impersonate existing institutions or services (e.g. banks) to different degrees of similarity: replicating the impersonated target to a high degree increases their chances of success, e.g. through the choice of a visually-similar domain, or through reusing graphics and logos. With the increase in popularity experienced by social media platforms, attackers found means to either directly reach targeted victims, also having the possibility to collect their data and increase the success likelihood, or to get in touch with large crowds in much broader campaigns. For example, Han et al. [16] mention Facebook among the top-five organizations targeted by phishers, also showing how attackers install off-the-shelf phishing kits in compromised web servers, where the attack is active for a short time before being moved to another location. Phishing attacks usually employ a considerable number of redirections, to avoid detection, evade blacklists and filter traffic. Previous work [29], [31] studied redirection chains for malicious pages detection, also applied in the context of social networks (i.e., Twitter). Detection of phishing pages can also be done by inspecting the content and structure of a webpage (e.g., Pan et al. [24]) or the URL structure (e.g., Chou et al. [7]).

As opposed to this body of works, our study does not present new detection techniques for phishing pages. However, similarly to phishing pages, an attacker can create link previews that are visually similar to benign ones, masking thus the malicious intention of the landing page.

C. Detection of Malicious Content

As social networks gained popularity, attackers started using them as a vector to spread malicious URLs, beyond phishing attacks such as drive-by download. The detection of these URLs has been the focus of several works. For example, Lee et al. [29] proposed a technique to detect malicious URLs based on the chains of redirections. Similarly, Thomas et al. [34] presented a technique to evaluate URLs shared not only on social networks but also on other web services such as blogs and webmails. In another line of work, the detection of malicious pages focused on inspecting their content, for both desktop browsers (e.g., Canali et al. [6]) and mobile browsers (e.g., Amrutkar et al. [1]). As opposed to these works, our paper does not present a detection technique, but it studies how social platforms behave when preparing previews of malicious URLs.

Finally, in a recent work, Bell et al. [4] measured the reactivity of the malicious URL detection system of Twitter, discovering that a significant number of malicious URLs remain undetected for at least 20 days. Such a study is orthogonal to the one present in our work, i.e., our work explores the ways social platforms generate previews in an adversarial setting, whereas Bell et al. [4] perform measurements on the reactivity of countermeasures. Also, to a certain extent, Bell et al. [4] underline the severity of the current state of link previews in social platforms too.

D. Cloaking Attacks

Another area related to our work is the area of cloaking attacks. In a cloaking attack, the attacker significantly alters the web page content when visited by a crawler or bot to conceal the malicious purpose of the page [40]. When compared to our

work, attackers could use cloaking attacks to generate deceptive link previews, where the page content is changed to look benign only when visited by social platforms' bots. However, cloaking attacks can be detected, and over the past years, the research community has proposed several ideas. For example, Wang et al. [39] show four techniques to detect user agent and IP cloaking put in place by web sites to deceive search engine crawlers. Similarly, Invernizzi et al. [18] used ready-to-use cloaking programs retrieved from the underground market to create a classifier for the detection. Social platforms could use these techniques to detect cloaking attacks; however, it is important to point out that it will not be sufficient to prevent the creation of deceptive previews. As we showed in our study, complying to our recommendations is hard in practice, and attackers can exploit a variety of implementation pitfalls (see, Section VI) to craft malicious previews and distribute unwanted content over social platforms.

VIII. Conclusion

In this paper, we presented a comprehensive analysis of link previews on social media platforms. First, we explored different ways in which their content is specified and how most of the platforms studied have a different rendering format for the same meta tags. We highlighted how this variability can cause the user not to understand which preview fields are security critical, leading them to uninformed security decisions. Then, we showed that it is possible to misuse the preview-rendering service, as this relies entirely on the content of the meta tags without inspecting the web page any further: in four social media platforms, we were able to craft benign-looking link previews leading to potentially malicious webpages. Crafting a benign-looking preview for the remaining 16 social media platform requires only the ability to register a new domain.

Next, we observed the presence of any active or passive countermeasures employed by social media platforms against the spread of known malicious URLs and software, and found that only two over 20 platforms perform active checks on the shared URL, and that even in these two cases, cross-checks can be bypassed through client- and server-side redirections. On this matter, we reported possible inconsistencies with the safe browsing services on mobile phones, supporting our recommendation on upstream checks, performed directly by the social media platforms. We concluded our work with a discussion, analyzing the impact of misleading previews on users' behavior, evaluating the resulting security risks, and suggesting seven recommendations for possible improvements.

Acknowledgments

We would like to thank the anonymous reviewers, Katharina Krombholz, and Sebastian Becking for their valuable feedback. Also we would like to thank Nick Nikiforakis, who shepherded this paper. This work was partially supported by the German Federal Ministry of Education and Research (BMBF) through funding for the CISPA-Stanford Center for Cybersecurity (FKZ: 13N1S0762).

References

[1] C. Amrutkar, Y. S. Kim, and P. Traynor, "Detecting mobile malicious webpages in real time," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2184–2197, 2016.

[2] M. Armstrong, "Referral traffic - google or facebook?" 2017. [Online]. Available: https://www.statista.com/chart/9555/referral-traffic---google-or-facebook/

[3] Ars Technica, "Armed with ios 0days, hackers indiscriminately infected iphones for two years," 2019. [Online]. Available: https://arstechnica.com/information-technology/2019/08/armed-with-ios-0days-hackers-indiscriminately-infected-iphones-for-two-years/

[4] S. Bell, K. Paterson, and L. Cavallaro, "Catch me (on time) if you can: Understanding the effectiveness of twitter url blacklists," *arXiv preprint arXiv:1912.02520*, 2019.

[5] D. Canali and D. Balzarotti, "Behind the scenes of online attacks: an analysis of exploitation behaviors on the web," 2013.

[6] D. Canali, M. Cova, G. Vigna, and C. Kruegel, "Prophiler: a fast filter for the large-scale detection of malicious web pages," in *Proceedings of the 20th international conference on World wide web*. ACM, 2011, pp. 197–206.

[7] N. Chou, R. Ledesma, Y. Teraguchi, and J. C. Mitchell, "Client-side defense against web-based identity theft," in *Proceedings of the Network and Distributed System Security Symposium, NDSS 2004, San Diego, California, USA*, 2004. [Online]. Available: http://www.isoc.org/isoc/conferences/ndss/04/proceedings/Papers/Chou.pdf

[8] J. S. Downs, M. Holbrook, and L. F. Cranor, "Behavioral response to phishing risk," in *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*. ACM, 2007, pp. 37–44.

[9] M. Egele, G. Stringhini, C. Kruegel, and G. Vigna, "Towards detecting compromised accounts on social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 4, pp. 447–460, 2015.

[10] Facebook Inc., "I got a message from facebook saying a file i tried to share has a virus." [Online]. Available: https://www.facebook.com/help/223268604538225

[11] ——, "The open graph protocol." [Online]. Available: https://ogp.me/

[12] ——, "What is facebook doing to protect me from spam?" [Online]. Available: https://www.facebook.com/help/637109102992723

[13] S. Garera, N. Provos, M. Chew, and A. D. Rubin, "A framework for detection and measurement of phishing attacks," in *Proceedings of the 2007 ACM workshop on Recurring malcode*. ACM, 2007, pp. 1–8.

[14] Google Inc., "Google safe browsing." [Online]. Available: https://safebrowsing.google.com/

[15] S. Gupta, A. Khattar, A. Gogia, P. Kumaraguru, and T. Chakraborty, "Collective classification of spam campaigners on twitter: A hierarchical meta-path based approach," in *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee, 2018, pp. 529–538.

[16] X. Han, N. Kheir, and D. Balzarotti, "Phisheye: Live monitoring of sandboxed phishing kits," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 1402–1413.

[17] J. Hong, "The current state of phishing attacks," 2012.

[18] L. Invernizzi, K. Thomas, A. Kapravelos, O. Comanescu, J. Picod, and E. Bursztein, "Cloak of visibility: Detecting when machines browse a different web," in *2016 IEEE Symposium on Security and Privacy (SP)*, 2016.

[19] K. Johnson, "Google safe browsing can differ between desktop and mobile. why?" 2019. [Online]. Available: https://www.wandera.com/mobile-security/google-safe-browsing/

[20] A. Le, A. Markopoulou, and M. Faloutsos, "Phishdef: Url names say it all," in *2011 Proceedings IEEE INFOCOM*. IEEE, 2011, pp. 191–195.

[21] C. Ludl, S. McAllister, E. Kirda, and C. Kruegel, "On the effectiveness of techniques to detect phishing sites," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 2007, pp. 20–39.

[22] A. Oest, Y. Safaei, A. Doupé, G.-J. Ahn, B. Wardman, and K. Tyers, "Phishfarm: A scalable framework for measuring the effectiveness of evasion techniques against browser phishing blacklists," in *PhishFarm: A Scalable Framework for Measuring the Effectiveness of Evasion Techniques against Browser Phishing Blacklists*. IEEE, 2019, p. 0.

[23] Open DNS, "PhishTank." [Online]. Available: https://www.phishtank.com/

[24] Y. Pan and X. Ding, "Anomaly based web phishing page detection," in *2006 22nd Annual Computer Security Applications Conference (ACSAC'06)*, 2006.

[25] G. Pellegrino, O. Catakoglu, D. Balzarotti, and C. Rossow, "Uses and Abuses of Server-Side Requests," in *Proceedings of the 19th International Symposium on Research in Attacks, Intrusions and Defenses*, September 2016.

[26] L. L. Porta, "Google's security efforts are falling short on mobile," 2019. [Online]. Available: https://www.brianmadden.com/opinion/Google-Safe-Browsing-differs-between-desktop-and-mobile

[27] E. M. Redmiles, N. Chachra, and B. Waismeyer, "Examining the demand for spam: Who clicks?" in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI '18, 2018. [Online]. Available: http://doi.acm.org/10.1145/3173574.3173786

[28] M. Robertson, "Modifying link previews," 2017. [Online]. Available: https://developers.facebook.com/blog/post/2017/06/27/API-Change-Log-Modifying-Link-Previews

[29] Sangho Lee and Jong Kim, "Warningbird: Detecting suspicious urls in twitter stream," in *NDSS*, 2012.

[30] S. Sheng, B. Wardman, G. Warner, L. F. Cranor, J. Hong, and C. Zhang, "An empirical analysis of phishing blacklists," in *Sixth Conference on Email and Anti-Spam (CEAS)*. California, USA, 2009.

[31] G. Stringhini, C. Kruegel, and G. Vigna, "Shady paths: Leveraging surfing crowds to detect malicious web pages," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, ser. CCS '13, 2013. [Online]. Available: http://doi.acm.org/10.1145/2508859.2516682

[32] Stringhini, Gianluca and Kruegel, Christopher and Vigna, Giovanni, "Detecting spammers on social networks," in *Proceedings of the 26th Annual Computer Security Applications Conference*, ser. ACSAC '10, 2010. [Online]. Available: http://doi.acm.org/10.1145/1920261.1920263

[33] B. Tawily, "Can you trust facebook links?" 2017. [Online]. Available: https://quitten.github.io/Facebook/

[34] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and evaluation of a real-time url spam filtering service," in *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, ser. SP '11, 2011. [Online]. Available: https://doi.org/10.1109/SP.2011.25

[35] K. Thomas, D. McCoy, C. Grier, A. Kolcz, and V. Paxson, "Trafficking fraudulent accounts: The role of the underground market in twitter spam and abuse," in *Presented as part of the 22nd {USENIX} Security Symposium ({USENIX} Security 13)*, 2013, pp. 195–210.

[36] Twitter Inc., "About unsafe links." [Online]. Available: https://help.twitter.com/en/safety-and-security/phishing-spam-and-malware-links

[37] ——, "Optimize with twitter cards." [Online]. Available: https://developer.twitter.com/en/docs/tweets/optimize-with-cards/overview/abouts-cards

[38] A. Vishwanath, T. Herath, R. Chen, J. Wang, and H. R. Rao, "Why do people get phished? testing individual differences in phishing vulnerability within an integrated, information processing model," *Decision Support Systems*, vol. 51, no. 3, pp. 576–586, 2011.

[39] Wang, David Y. and Savage, Stefan and Voelker, Geoffrey M., "Cloak and dagger: Dynamics of web search cloaking," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, ser. CCS '11, 2011. [Online]. Available: http://doi.acm.org/10.1145/2046707.2046763

[40] B. Wu and B. D. Davison, "Detecting semantic cloaking on the web," in *Proceedings of the 15th International Conference on World Wide Web*, ser. WWW '06, 2006. [Online]. Available: https://doi.org/10.1145/1135777.1135901