Revisiting Deceptive and Malicious Advertisements in the Wild

Clemens Bönnen(✉)[0009−0002−1138−6300] and Ulrike Meyer[0000−0002−2569−1042]

IT Security Research Group, RWTH Aachen University, Aachen, Germany
{boennen,meyer}@itsec.rwth-aachen.de

Abstract. The growth of the online advertisement market and its structure increasingly attract malicious actors that use advertisements to directly or indirectly distribute malicious content to end users. While this problem has been known as malvertising for more than a decade, an up-to-date large-scale analysis of the currently used deception strategies exploiting recent ad delivery mechanisms and the threat associated with them seems missing. We propose an extensible, automated tool to identify and collect advertisements on publicly available websites. Using this ad crawling tool, we collected and analyzed a large data set of online advertisements from frequently visited websites and present our insights.

Keywords: Online ad collection · Malvertising · Deceptive techniques.

1 Introduction

The online advertising market today is an important source for cross-financing operating costs for services offered on the web. The Internet Advertising Revenue Report of 2024 [16] shows a steady increase of the global online advertising revenue that nearly doubled between 2020 and 2024.

Online advertisements (ads) enable an advertiser to reach a large, targeted group of users with low costs and effort. Thus, ads are also attractive for attackers that want to distribute malicious content. The large market volume and the near real-time nature of current market mechanisms make it challenging to filter such content. Real-Time Bidding (RTB) is the most prominent approach to trade ads today [16] making it possible to sell and buy targeted advertising with no time ahead. The ongoing trend of browser vendors to restrict ad blocking [11,13] shifts ads back into the attention and makes abuse even more attractive.

Malvertising is a well-known problem to the ad industry [2, 8, 42] and has consequently been studied for over a decade. While some work focused on collecting data and analyzing the problem in a specific context (e.g., [2] focused on Facebook, [29] on online streaming platforms, and [30] on Android in-app ads), studies concentrating on publicly accessible web sites [43] seem rather outdated and do not cover ad delivery methods beyond the integration as static iframes.

To enable studying recent developments on the web and overcome the limits of prior approaches, we present MalvertCollector, our effort for an extensible, automated approach to collect online ads. MalvertCollector is capable to collect

classic iframe embedded ads and modern native ad elements while rating the ad target and recording a data set containing screenshots, redirections, SSL certificates, and whois records.

We analyzed ads collected on the 80,000 most visited URLs on the Tranco list, as well as search results and sponsored results returned by a Google search using the 1,000 most frequently used search terms. We use the gathered insights to determine today's prevalence of ads during everyday surfing and their occurrence in different ad networks. We also utilize manual inspection to analyze currently used deception mechanisms.

2 Background

The design of MalvertCollector requires a broad understanding of online advertising technology like real-time bidding, user tracking using redirections, ad integration methods, and crawling evasion techniques.

Online Advertising using Real-Time Bidding Online advertising is an increasingly complex process which involves a variety of entities and automated mechanisms [28]. While *advertisers* want to promote products and services using *advertisements* (ads), *publishers* want to generate revenue by presenting ads. Such ad presentations (called *impressions*) are marketed through an *ad network* connecting advertisers with publishers. The most prevalent marketing mechanism is called *real-time bidding* (RTB) [16]. RTB markets impressions using an auction broadcasted in the ad network with the highest bid winning. Thus, the publisher and even the RTB provider do not know the exact online ad in advance, opening up potential for abuse. Thus, providing a trustworthy environment within the ad network is crucial for the reputation of all participating entities, especially for the publishers.

Integration of Online Advertisements Ad impressions include display ads like banners, native ads (e.g., sponsored links), media playback, and push notifications. Native ads are often seamlessly integrated into the look and feel of the publishers website, making legally required indications of ads easy to overlook.

The classic integration of ads using iframes evolved to today's script based direct integrations of ad elements into the publisher's website, allowing a dynamic placement of ad content. Such scrips are typically integrated in the header of a publisher's website and controlled by the ad network partner of the publisher, which allows to transfer the decision of the ad placement to the ad network. Display ads may also allow direct interaction, including scrolling, swiping, and clicking. These interactions can influence the final ad target (e.g., swiping between multiple products of the same web shop).

User Tracking and Redirections Behavioral advertising and retargeting are the most common mechanisms regarding personalized advertising today and crucial for the effectiveness of ad campaigns. Behavioral advertising refers to presenting ads fitting the interests of the users, while retargeting refers to presenting ads for products or services the user already showed interest in. Both

require tracking of the user's online behavior, including visited websites, presented ads, and re-identifying the user during subsequent visits to the same websites. While some operating systems or software provide persistent advertising IDs using an API [1], online ads on websites have to rely on the browser's ability to re-identify a user. The web storage API and cookies require explicit legal consent from the user, often in the form of a popup. The content of the website is often invisible or behind an overlay until the user decided whether to allow such identifiers as the decision influences the presented ads, potentially leading to less personalized ads.

To track users across different websites, cookie synchronization can be used. Instead of directly linking the ad target, the user is subsequently redirected to one or more tracking servers, where the user can be re-identified by previously set cookies and the identifier can be synchronized using user-specific URL parameters. Tracking servers often use HTTP status codes for redirects (i.e., status code 301) but also client side html/script-based redirections can be observed.

Cloaking and Detection Methods Threat actors often use server-side cloaking to prevent their malicious actions being detected by crawlers [46]. Cloaking entails presenting different content to different requests depending, e.g., on the user-agent string or the request's source IP address. Thus, crawlers can be presented benign content, even if users would see malicious content.

3 Requirements

In this section we specify and rationalize general requirements (CR) to enable an effective collection of current online advertising. In addition, we also specify requirements for the measurement study (SR) itself that are in line with our ethical considerations (see Section 6.3).

CR1: The crawler needs to be able the collect ads from a regular users perspective while avoiding any influence of past interactions on the presented ads. Identifiers stored in cookies are used to re-identify users in the context of behavioral advertising and re-targeting. While the crawler thus needs to generally not prohibit to set cookies but **impede the tracking and re-identification** by deleting persistent data set by ad network entities before continuing crawling.

CR2: For ad detection, we require the crawler to automatically **detect current online ad delivery methods**. As websites often present a cookie banner overlaying the rest of the content, we also require **the interaction with cookie banners** to avoid ads being hidden behind an overlay or not being loaded at all.

CR3: To allow for analysis of the ad target, we require to **invoke an ad interaction flow** and to **capture the ad target**. To reach the ad target, usually a click on the ad is required. Due to session specific and time-limited data in URL parameters, payloads, redirection chains, and additional requests, just extracting links from an ad object may hinder to reach the ad target later on.

CR4: To allow for a comprehensive inspection of the found ads and their nature, we require the collection mechanism to **capture a set of additional data**

containing a screenshots of the display ads and the ad target page, URLs of redirections, SSL certificates, whois records, and server location data.

SR1: We require the study to provide an **opt-out mechanism** for content providers recognizing the crawler which prohibits the contact to URLs, IPs or subnets for which the owners opted out of the measurement study.

SR2: We require the study to support **source URL generation** that allows to reflect average user behavior, e.g., from a *static list* of frequently visited websites, or dynamically from *search engine results* of frequently used keywords.

SR3: For scalability, we require **the repeated automated rating of ad target URLs** to analyze statistics about maliciously rated ads and pre-filter the data set for potentially malicious ad targets that are to be inspected manually.

4 Related Work

Multiple studies focus on malvertising in the context of specific services or platforms. In 2020, Arrate et al. analyzed the spreading of malicious ads within facebook [2]. As facebook operates its own closed ad marketplace, we reckon that ads displayed on facebook likely differ from the ones displayed on publicly accessible websites with third party ad networks. The authors of [2] compared their results with ads collected during a study [17] investigating targeted ads without a focus on malicious ads. The work by Rafique et al. [29] focuses on content that was displayed alongside free live online streaming services. Due to the often debatable legal background of the service, a larger amount of deceptive and malicious ads can be expected and may not be representative w.r.t. general prevalence and used deception techniques on frequently visited websites. Rastogi et al. [30] investigated in-app ads on Android. While mobile ads are similar to ads displayed on the web, placement of ads in mobile apps is less flexible and the collection process naturally differs.

Subramani et al. investigated in [37] the usage of browser push notifications as online ads and collected push notifications sent to users by ad networks. We also encountered requests to permit push notifications but did not further investigate them. Thus, their work provides complementary context for this type of malicious content (see Section 7.4 for more insights).

In contrast to the prior work discussed so far, our work and the ones discussed next focus on online ads placed on frequently visited websites. In 2014, Zarras et al. in [43] collected online ads on websites based on detecting iframe source URLs matching lists of known ad URLs. While this probably covered the main ad integration method at that time, today ads are often native elements placed on the page using scripts. In 2017, Masri and Aldwairi conducted a measurement of malicious ads in [22] using two initial data sources: the alexa list as benign entries and a collection of block list entries as malicious ones. Like [43], they only did ad detection based on iframes. The assumption that the alexa list generally contains benign domains, seems questionable in the light of the results of our own analysis (see Table 2). In 2019, Vedrevu and Perdisci investigated malicious

Requirements		Approaches [44]	[10]	[2]	[29]	[30]	[37]	[41]	[43]	[22]	This
Impede re-identification	**CR1**	○	◐			○	●	●			●
Interact with cookie banner	**CR2**	○	○			○	○	○			●
Detect current ad delivery methods		●	○	○	○	○	-	○	○	○	●
Ad interaction to invoke ad flow	**CR3**	●	○	●	●	●	●	●		○	●
Capture ad target		●	○	●	●	●	●	●		○	●
Collect additional data	**CR4**	◐	◐				◐	◐			●
Opt-out mechanism	**SR1**	-	-	○			○	○			●
Frequently visited sources	**SR2**	-	-	○	●	◐	●	●	◐	●	●
Repeated automated rating of ad target URLs	**SR3**	-	-	●		●	●	●	●	●	●
Availability		●	●	◐	○	○	●	●	○	○	●

- unapplicable; (empty) unknown; ○not satisfied; ◐partially satisfied; ●fully satisfied

Table 1. Comparison of features supported by ad crawling software and malvertiment studies. Undocumented features were manually checked in the source code, if available.

online ads using social engineering in [41]. They collected ads on websites using a set of known ad-network scripts and a script search engine, resulting in a set of ads from a limited group of ad networks.

How frequently malicious ads are encountered today on frequently visited websites as well as their occurrence through ad networks with differing reputation seems underexplored. In this context, it is also of interest, what types of attacks, deceptive measures, and in which contexts such malicious ads can be found. To this end, we crawl frequently used websites, capture online ads regardless of their embedding type, and analyze the results with the help of rating engines.

Apart from the previously presented approaches focusing on malvertising, ads are also collected in other contexts. OpenWPM [10] is a privacy measurement framework, which was used, among others, to investigate privacy implications of online ads in various studies [5, 31, 32, 38, 39]. However, this framework focuses on privacy measurements and does not directly support ad detection.

In [45], Zeng et al. collected ads to conduct a user study on the perception of online ads. Their AdScraper [44] is able to collect ads displayed on websites but provides limited recording features w.r.t. additional information on the ads collected, and is not able to interact with cookie banners.

We compared our approach with AdScraper in Section 7.1. In Table 1, we compare our ad collection approach to the ones used in previous studies on malvertising as well as approach previously proposed in other contexts.

5 Crawler Design and Implementation

This section provides an overview of MalvertCollector, the crawling tool we developed during this study. First, we provide a general overview of MalvertCollector, its design choices, and discuss some aspects of the implementation.

MalvertCollector Overview A high level overview of MalvertCollector is depicted in Figure 1. The source lists, the rating engines, and the capture services are modular and can be supplemented by new ones if needed. The central component of MalvertCollector is an automated web browser controlled by selenium

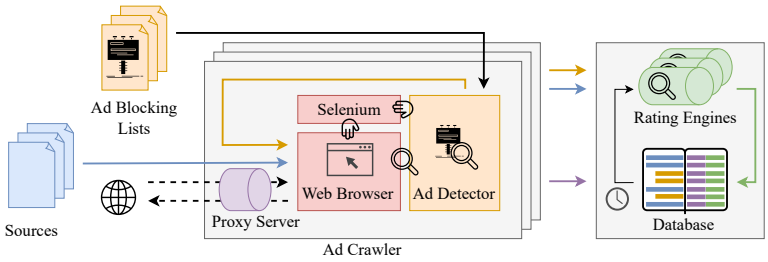


Fig. 1. Overview of MalvertCollector and the crawling process. Blue arrows represent the handling of initial sources, the orange arrow represents the handling of ad targets.

(colored red in Figure 1). The ad detection mechanism is responsible to find and interact with ads on currently visited website using the browser. A proxy server (purple) records network traffic to detect potential redirections and rating engines (green) are used to rate the ad target URLs and pre-filter them. The main aspects of MalvertCollector will be further introduced in the following.

Cookie Handling The ad collection starts with the web browser visiting the source website. As behavioral advertising and retargeting can influence presented online ads and thus introduce a bias in the collected set of ads, we delete all stored cookies and website data before visiting the source page to collect new ads. While browser fingerprinting is hard to avoid, we limit the identifying information of the browser by not installing additional extensions or fonts and leave the setting at their defaults. Section 6.3 discusses limitations regarding anti-tracking measures.

Due to requirement CR2, MalvertCollector needs to interact with cookie banners. To this end, we rely on the browser extension *Super Agent* [24] to automatically grant permission to use cookies for all websites.

Ad Detector After opening and preparing the source page, the ad collection process can start. Ad blocking extensions directly filter requests for scripts loading dynamic ad content and block them before they are executed. As a consequence ads are not loaded and cannot be collected. Instead, MalvertCollector detects the displayed ad itself and does not suppress loading of external resources, like scripts. To fulfill requirement CR1, MalvertCollector identifies ads using two different approaches, both of which are based on regularly updated filter lists of well-known ad-blocking extensions.

The first approach utilizes domains from these filter lists to detect externally fetched resources in iframes. MalvertCollector iterates over all (potentially nested) iframes and checks the URL of the iframe source containing a domain from the filter list. The second approach covers native ads directly inserted into the website, including iframes without a source attribute set. For this, the filter lists contain two-part entries consisting of a Cascading Style Sheet (CSS) selector to identify the native ad elements on the page and an optional regular expression for URLs to restrict this selector to be used only on pages with a matching

URL. MalvertCollector uses all CSS selectors without a filter and with a filter matching the currently visited URL to search the page for matching elements.

To interact with the same ad just once, collected ads are sanitized and only the outermost element of nested ad elements is considered. Also, to filter for tracking elements, we consider only elements that are larger than 5×5 pixels. As required by CR3, we interact with all found ad elements and use a best-effort click-strategy for this: For each ad, a list of potentially clickable elements (links, buttons, images, and the center of the ad; in this order) is clicked on one after another until a new page opens or the ad will be marked as unsuccessful.

Proxy Server During interaction with the ad, MalvertCollector uses a proxy server to capture any occurring redirections. These visited URLs along a chain of redirections can contain valuable data regarding tracking and ad networks (cf. Section 2). We use selenium-wire [18] for this, an extension of selenium that automatically runs a TLS intercepting proxy server alongside selenium and properly configures the browser to use the proxy. After 3 seconds without any new requests, MalvertCollector detect HTTP-based (HTTP status code), HTML-based, and script based redirection chains ending in the currently visited URL.

Rating Engines and Data Storage We implemented an automatic rating mechanism based on external services to pre-filter the set of collected ads for the analysis and manual inspection. MalvertCollector currently includes: the VirusTotal API, the Google Safe Browsing (GSB) API, and a block list lookup. These can be extended using a common interface.

The VirusTotal API is used to retrieve the rating results of the vendors included in VirusTotal. Some of these vendors not only rate URLs as benign and malicious but also report URLs as suspicious. The GSB API is used to maintain a list of hash prefixes. For each URL, as instructed by Google, six hashes for different parts of the URL are checked if their prefix occurs in the prefix list. For matching hash prefixes, the API is used to confirm the rating and get more information. The third engine regularly fetches publicly available block lists and checks if the URL of an ad target contains a domain from these lists.

For each visited URL, MalvertCollector stores: the rating, the reason why it was visited (i.e., on Tranco list, Google search result, or an ad) with parameters (e.g., search keyword or where the ad was found), any redirections, the IP-based geo location of the server[1], screenshots of the whole page and each identified ad, SSL certificates, and whois records.

6 Measurement Study and Analysis

We used MalvertCollector to create a dataset of ads displayed on frequently visited websites using the setup described in Section 6.1 and analyzed the gathered data as detailed in Section 6.2. We discuss technical limitations of our study and their relation to ethical considerations in Section 6.3.

[1] Determined using the GeoLite2 City database from MaxMind [23].

6.1 Setup for the Measurement Study

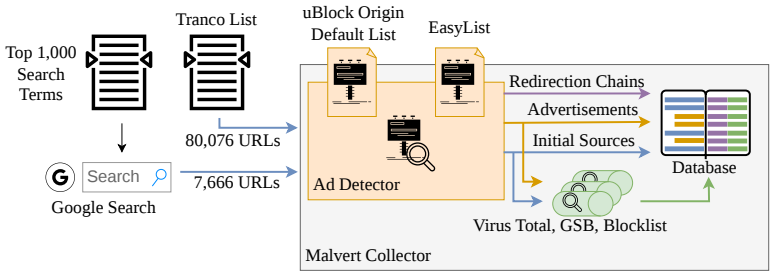


Fig. 2. Setup of the measurement. All used external sources and services are depicted.

Figure 2 provides an overview of the setup of the measurement study. Regarding initial sources, MalvertCollector supports both, a static URL list handed to it and dynamically creating a list of URLs based on online searches. For the static URL list, we used the Tranco list [20] (May 2024, list ID: `J99GY`) and visited the landing pages of the first about 80.000 domains. To dynamically create a list, we used Google search[2] to search the thousand most frequently used search terms worldwide from Aug. 2024 [6]. For each search term we visited the first 5 search results together with all sponsored results.

We used Firefox [25] 130.0, the GeckoDriver [26] 0.35.0 with the Selenium python binding [33,34] (ver. 4.23.1). The proxy server was provided by seleniumwire [18] 5.1.0. For handling cookie banners, we used Super Agent [24] 3.0.3. As rating engines, we used VirusTotal with all available vendors and an academic access, Google Safe Browsing with the list for "social engineering" and "malware", and the block list of Phishing Army [7] which merges the reports of PhishTank, OpenPhish, Cert.pl, PhishFindR, Urlscan.io and Phishunt.io. Each visited website was rated twice with at least a week in between the ratings. Our ad detection used the well-known filter lists EasyList [9] and the uBlock origin lists [15] as starting point. The crawling was conducted between Sep. 2024 and Dec. 2024 from a german IP address associated with our University and overall includes the results for 114,594 unique URLs visited as source URLs or due to ad detection.

6.2 Methodology for Data Analysis

In Section 7.1, we first quantitatively analyzed the results obtained based on the automatically collected data and the rating obtained from the rating engines. All entries were considered malicious if there was at least one positive record by at least one rating engine, even if other engines did not rate the entry as malicious.

[2] because of it's large market share of about 90% [36]

To gain further insights into the deceptive techniques and context used, we also manually inspected all ads that were rated as malicious and related to online ads or search results, including sponsored search results as native ads. We assigned labels for two aspects for these entries if possible. One label considers the presumed threat type of the malicious ad target, while the other one reflects the context of the ad target and categorizes what the ad target appears as. The set of labels was iteratively created by one researcher during the manual inspection and new labels were added whenever the existing ones do not fit. The labels were assigned as follows: As some rating engines work solely on the domain and not the complete URL, we also considered all information collected from all entries sharing the same domain in the URL for the manual inspection and assigned labels in most cases based on this domains. After reviewing the visual perception of the page using the screenshots, we considered the results from the rating engines and assigned a potential threat category if the information are in line. If there were ambiguities, we created new screenshots using urlscan.io [40] and interacted with the website from our crawling host to further narrow down the threat type. If there were still ambiguities, a quick search of the domain using the search engine Google took place to find potential user reports in forums or other related information before the entry was labeled as unclear. Labels for the context in which the ad target appeared to the user (e.g., a website showing a news article promoting a fraudulent investment) were solely assigned based on the visual perception. We detail selected examples of deceptive measures found during manual inspection in Appendix A and use the labels to provide further statistical insights into the crawled data in Section 7.1.

6.3 Ethical Considerations and Limitations

Our study was guided by best practices and prior work [4,14,19,27]. As required by [19], we considered the impact of our measurement study on all involved entities. As our study did not involve humans directly, we do not process personally identifiable information. However, ad network participants and content providers are affected by our study by the computational load we create by each website visit and the cost of ads shown to our crawler. We limit these affects of our crawling as far as possible: we avoid concurrent or repeated visits to the same site and interact only once with each ad. As prior work, we argue that the financial impact of our ad interactions for a single entity is negligible [45].

MalvertCollector is identifiable as a research project by its IP associated with our university in Germany and by a website hosted on the crawler with contact details and opt-out instructions. All collection attempts and manual inspections are made from the same IP address. The static IP address and browser fingerprinting may affect ads presented to MalvertCollector in general (e.g., by IP-based cloacking or IP location[3]) or due to retargeting and behavioral advertising . We, however, delete all information (e.g., cookies) set by the visited websites during the interaction with it before requesting the next URL. Thus,

[3] The IP address is typically used for the user's geographical location in bid requests.

the ads presented to the crawler on subsequently visited websites will not depend on each other. Using a dynamic IP, a VPN or an anonymization network like TOR may address some of the limitations but would have interfered with a clear opt-out mechanism. Also, IP addresses of public VPN services and TOR exit nodes are publicly known and could be subject to cloaking. MalvertCollector may also be hindered to crawl content by anti-bot measures, like CAPTCHAs, and does not try to solve or circumvent these. Due to the complexity of redirections methods, our detection of redirection chains may miss entries of a chain.

The ratings of ad targets is based on the presented rating engines from which we can not assess the accuracy. Thus, we expect the dataset to contain false positives and false negatives. Especially, the block list collection and GSB showing significant lower number of detections than VirusTotal. The manual inspection is based on perceived visible deception techniques and reports. As only one researcher conducted the analysis, it may contain misclassifications. Only ad targets already rated as malicious were inspected, potentially missing any false negative examples. Due to the amount of collected ads, a manual inspection of all ads was infeasible. Some ad targets were not reachable at the time of manual inspection any more. Their analysis was restricted to the crawled information.

7 Results and Discussion

This section presents the results from our crawling efforts using MalvertCollector. Section 7.1 presents statistics based on the collected data and ratings from the rating engines only, Section 7.2 adds the results of the manual inspection, providing an overview of the different attack types found.

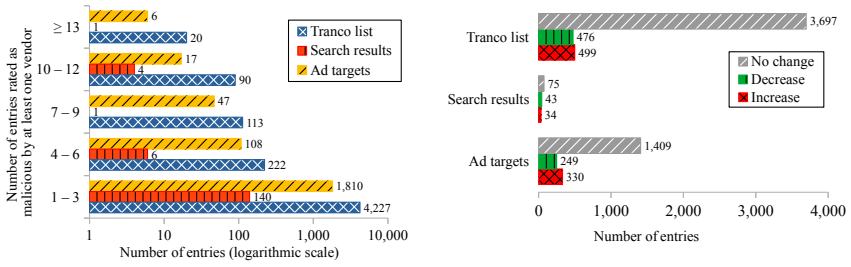
7.1 Dataset Overview and Statistical Features

Due to the variety of ad integration methods, we do not expect to capture all ads and work on a best effort basis. However, for a preliminary evaluation of the effectiveness of MalvertCollector compared to the previously proposed AdScraper, we used 500 websites from the Tranco list. AdScraper detected 152 ads on these, but only 35 of them showed ad content on screenshots. Using the same set of websites, MalvertCollector detected 231 ads with 187 screenshots showing ads. Unsuccessfully captured ads do not only represent false positives and often comprise ads (partially) covered by other content (e.g., parts of cookie banners) or ad content not yet loaded (blank) that would hinder an interaction with them. In our main study we therefore concentrated on MalvertCollector.

Table 2 provides an overview of the crawled sites. The data set contains 114,594 entries in total with 80,076 (69.9%) source URLs from the Tranco list, 7,666 (6.69%) source URLs found using the search engine, and 26,852 (23.4%) ad target URLs. One ad target is associated with an ad found on a source website and both are considered malicious if the ad target is rated malicious. 22,060 ads were directly found on websites from the Tranco list. We indirectly found another 480 ads on websites redirected to from a URL found on the Tranco list. Also,

Found on due to ads on			Total	Rated as malicious			
				GSB	Blocklist	VirusTotal	Total
Total			114, 594	40	19	6, 812	6, 812
Tranco list			80, 076	35	14	4, 672	4, 672
Search results			7, 666	1	2	152	152
Ad targets	Tranco list	22, 060 (480)	26, 852	4	3	1, 988	1, 988
	Search result	4, 306 (6)					

Table 2. Number of database entries and malicious reports.



(a) # of malicious reporting vendors (b) Changes of # of malicous ratings

Fig. 3. Malicious rating of VirusTotal vendors.

4, 306 ads were found on websites of google search results with additional 6 ads also found due to redirections from search results.

Table 2 also contains the number of entries rated as malicious at least once. While Google Safe Browsing (GSB) and the block list collection each rated only a small set of entries as malicious, VirusTotal had the highest number of malicious ratings, including the ones from GSB and block list. 6, 812 entries (5.94%) were rated as malicious at least once by at least one vendor in VirusTotal. While 7.40% (1, 988) of all ad targets were reported as malicious, only 1.98% (152) of search results and 5.83% (4, 672) of Tranco list entries were reported. The corresponding ads of the 1, 988 ad targets rated as malicious were presented on 855 distinct websites. 653 of the source websites were never reported by any of the rating engines and showed 1, 552 of the ads leading to malicious ad targets.

We also evaluated the number of vendors included in VirusTotal that flagged an entry as malicious. Figure 3a shows the number of maliciously rated entries (logarithmic scale) against the maximal number of vendors rated this entry as malicious in any of the two ratings for each entry type. A large portion of malicious entries has only one or a few reporting vendors. Figure 3b shows for how many entries the number of vendors rating this URL as malicious decreased (green), increased (red), or did not change (grey) between the two ratings. For advertisements the increase is larger than the decrease, but the number of unchanged reports is still largest. While Tranco list and search results had nearly equally many increased and decreased entries each, the portion of unchanged reports was significantly larger for the Tranco list.

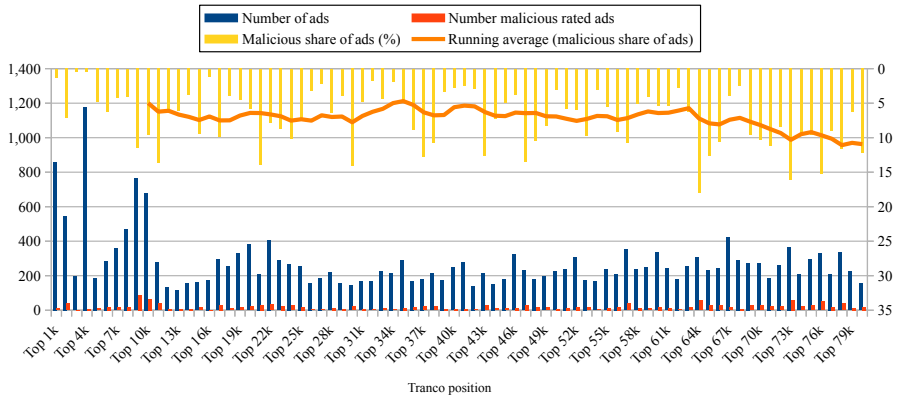


Fig. 4. Number of ads and malicious ads found per position on the Tranco list.

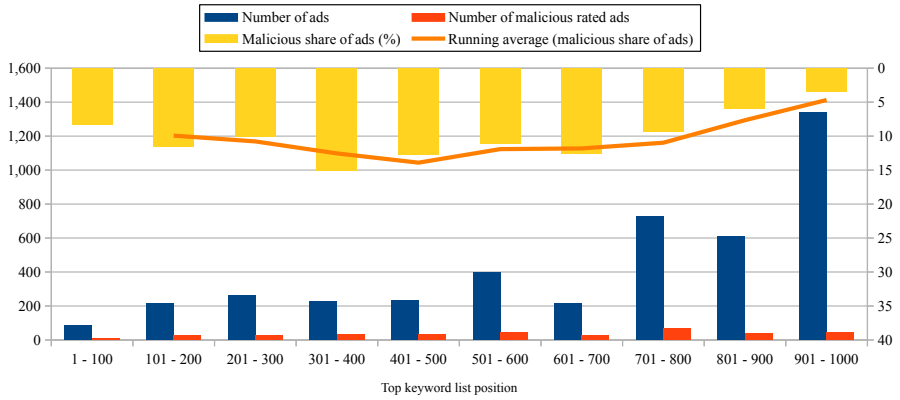


Fig. 5. Number of ads and malicious ads on websites per positions on the keyword list.

Figure 4 shows the total number of ads (blue) we found on source websites sorted by the position of the source on the Tranco list. We use this position as a measure of popularity. We grouped the position into bins representing 1,000 websites each (e.g., "Top 1k" include the positions 1-1,000, and "Top 2k" the positions 1,001-2,000). A higher number of ads per bin is found on the first 10,000 entries of the Tranco list, spiking at 1,200 ads. The rest of the ads found is evenly distributed across the remaining bins, ranging from 200 to 400 ads per bin. Figure 4 also shows the number of ads leading to malicious ad targets (red), the share of these among all ads in the bin (yellow), and the running average of the share over the last 10 bins (orange). For the first 4,000 websites on the Tranco list, Figure 4 shows a significantly smaller share of malicious ads, than for the other websites and a slight increase for entries below the "Top 63k".

All ads	Malicious rated ads
pagead2[.]googlesyndication[.]com	pagead2[.]googlesyndication[.]com
adclick[.]g[.]doubleclick[.]net	trc[.]taboola[.]com
trc[.]taboola[.]com	s[.]magsrv[.]com
googleads[.]g[.]doubleclick[.]net	endowmentoverhangutmost[.]com
ad[.]doubleclick[.]net	googleads[.]g[.]doubleclick[.]net
kdcl[.]pchome[.]com[.]tw	www[.]googleadservices[.]com
www[.]bet365[.]com	ads[.]traffichunt[.]com
beacon[.]taboola[.]com	beacon[.]taboola[.]com
track[.]adform[.]net	novtracker[.]store
s[.]magsrv[.]com	maximparerurehab[.]com

Table 3. The 10 most often used domains in redirection chains

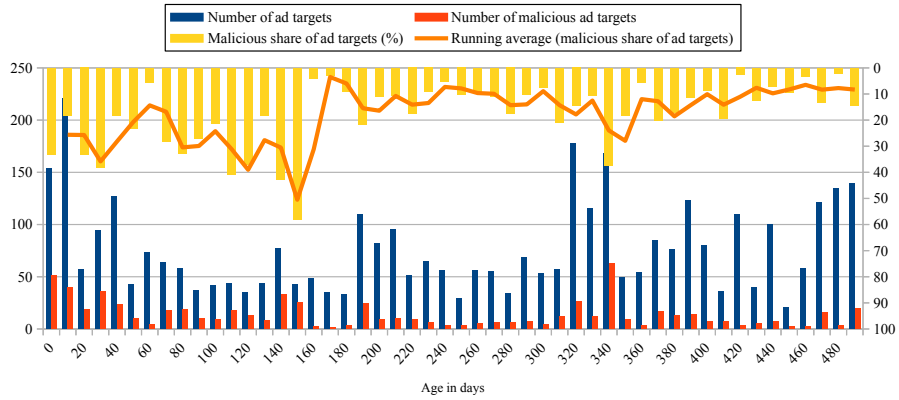


Fig. 6. Number of ad targets and malicious ad targets in age (whois) groups in days.

Figure 5 shows a similar metric for the Google search, grouping the 1000 keywords used into bins of 100 keywords according to their position on the list. Again, the share of ads corresponding to malicious rated ad targets per bin (yellow) is shown together with a running average of 2 bins (orange). It stands out that more ads were be found on websites from Google searches the further down the related keyword was. The number of malicious ads stays relatively constant and thus the share of malicious ads decreases further down the list.

Based on the ad targets, for which we were able to record redirections, Table 3 shows the 10 most often occurring redirection FQDNs for all ad targets as well as for ad targets that were rated as malicious. As expected, a lot of the domains belong to ad agencies, like Google or Taboola. For the malicious ads, the domain `s[.]magsrv[.]com` is much more frequently used and belongs to ExoClick, an ad agency known to provide ads for high volume websites from the entertainment and adult segment. In addition, the domain `endowmentoverhangutmost[.]com`, which has been associated with malvertising before [21] and is not directly affiliated with an ad agency appears in our measurements. We also analyzed the used eTLD of the ad targets and identified a higher usage of new TLDs and generic TLDs by malicious ad targets.

We estimated the age of the domains of ad targets using the creation date or update date in the whois records where available. Figure 6 shows the distribution of all ad targets (blue) and malicious ad targets (red) into age groups of 10 days. We also depict the share of malicious ad targets for each group (yellow) and the running average over two groups (orange). For all ad targets, multiple spikes are visible: below 40 days, around 200 days, and 330 days. The share of malicious ad targets shows a significant drop at around 150 days and stays at a lower level. As it was not always possible to capture the age of whois entries, malicious ad targets seem to be overrepresented in this statistic with 17.6% compared to their general share of 7.4%. Also, the estimated age of ad targets based on certificates did not reveal any clear pattern.

7.2 Results of the Manual Inspection

We manually inspected 2,140 entries rated as malicious in our database with 1,988 URLs related to ad targets and 152 URLs stemming from search results (cp. Table 2). We omitted any reports for entries from the Tranco List as these are not directly related to ads and are too extensive for a manual inspection.

Section 6.2 describes how context labels were assigned. Table 4 shows the frequency of the 14 most often assigned context labels. The highest number of entries was assigned the label *Browser Extension*, followed by *Adult Content* and *Parked Domains*. Other categories were significant less common.

The ten most common threat types and frequencies are shown in Table 5. The most prevalent types were *Potentially Unwanted Application*, almost exclusively related to browser extensions, *Ad Redirection*, which was assigned to entries with the sole purpose to show other ads or redirect to other (malicious) ad targets, followed by *Credential Phishing*.

Category	#	**Category**	#
Browser Extension	610	Anti-Malware	13
Adult Content	460	Online Casino	13
Parked Domain	302	Online Bets	11
Game	63	Video Streaming	9
Advertorial	29	Checkout	9
CAPTCHA	19	Fake News Site	4
Products	17	URL Shortener	4

Table 4. Most often assigned contexts.

Type of threat	**Number of entries**
PUA	610
Ad Redirection	234
Credential Phishing	190
Malware	64
Browser Notifications	27
Investment Fraud	23

Table 5. Entries by threat types.

It was not possible to always assign exactly one label for threat type and context, as for some entries we could not identify an underlying attack, the context matched multiple labels (e.g., “advertorial” and “fake news site”), or the ad target was not reachable anymore. In total, we identified 891 entries as unlabeled with 753 stemming from ad interactions and 138 from search results. We assigned labels for the context and/or the threat type for the remaining 1,249 entries (1,235 ads, 14 search results). For a significant portion (91%) of

the reported entries from search results we could not assign a label, while this holds only for 38% of the entries in the ads category. The majority (98%) of unlabeled entries were rated as malicious by one or two vendors in VirusTotal.

7.3 Discussion

We observed that the first few thousand sites on the Tranco list provided more ads in comparison to websites ranked lower on it. Even websites with high volume traffic make it attractive to place ads, for search terms, the opposite was the case. If the position on the Tranco list and the position on the list of frequently used search terms are used as a measure of popularity of the corresponding site, this opens up the question of why the distribution of ads differ.

Among the rating engines, VirusTotal produced the most malicious ratings while GSB and the block list collection resulted in few detections, missing most of the later manually reviewed threats. However, this does not allow to draw any conclusion about the accuracy of VirusTotal. The number of detections is not surprising considering that VirusTotal provides the aggregated results of multiple vendors, but is unfortunate as GSB is often implemented in browsers and in good position to protect users against malicious web content.

The fact that we were able to find ads that lead to maliciously rated ad targets across the Tranco list indicates that malvertising is a widespread threat that is not focused on just niche websites, but can be found also on reputable and often visited websites. However, especially the top $4,000$ websites of the Tranco list showed less but not none malicious ads during our measurement and the results indicate a slight increase towards less frequently visited websites. Websites found using the Google search also include malicious ads, but due to an increasing amount of the totally found ads the lower the used keyword was on the list, the share of malicious ads decreases.

While malicious ads seem to be placed throughout frequently visited websites, our analysis of the redirection URLs show that even large ad content provider, like Google, cannot avoid ads that lead to maliciously rated targets. Our data also indicate that some ad networks, like ExoClick, seem more prominently involved in malvertising. We reckon that this may be due to large ad networks, like Google, providing a more reputable ad basis by providing easier options to report ads that violate the terms and conditions of the ad network [12]. In our measurement we also discovered a redirector domain that is known for malvertising, underlining the potential for improvement of protection mechanisms. In addition, it seems that new TLDs and generic TLDs are more prominent among malicious ad targets probably because of cheaper registrations or has less strict requirements.

The observed domain age of the malicious ad targets based on whois records indicate that most of them seem long lasting with longer or potentially multiple campaigns per domain. This suggests that even simpler protection methods like block lists are not implemented. However, the discovered higher share of malicious ad targets from 0 to 160 days could stem from take down efforts that lead to updated whois records and also different policies of the used TLDs as we have already seen that malicious ad targets use new TLDs more often.

7.4 Summary of Observed Deceptive Techniques

During the manual inspection we observed different deceptive measures and examples are described in Appendix A.

The deceptive techniques observed show that malicious advertisers make use of common interactions users also meet on benign websites, like CAPTCHAs to evade content crawlers. This technique seems relatively new and was not mentioned in other malvertising studies yet, but came up in the news recently [35]. Multiple of these malicious ad targets asked for permission to send push notifications. Vadrevu et al. also discovered in [41] that users are lured into allowing push notifications but did not describe how this was accomplished. Push notifications allow for multi-step attacks including a second step with notifications threatening the user of missed updates or found malware. The content of ads in web push notifications was investigated by Subramani et al. in [37].

Scareware involving fake warning of anti-malware was also observed directly on ad targets. Mimicking the design of a locally installed software, shown warnings convey the urgency to act. Scareware is a widely spread threat and was discovered in the context of online ads in several other studies [2, 29, 41]. PUAs were also discovered by Arrate et al. in [2] in the category of slightly risky ads, but without manual inspection or further details.

We also discovered potential credential farming based on some account creation requirement to access content, often involving explicit content. Such deceptive credential farming was discovered previously, e.g., with lottery scams [41]. As opposed to others [30,37,41], we did not observe any technical support scams, fake lotteries, or free giveaways, but our study revealed investment scams related to crypto currencies and AI-based investment using advertorials on imitated news sites. This can indicate a shift towards new topics, but can also be related to a location or time based bias.

8 Conclusion

We presented MalvertCollector that is capable to automatically collect, interact, record, and rate online ads. We used it to collect 26,852 ads from the 80,000 most frequently visited websites gathered from the Tranco list, Google search results, and sponsored links. In total we found 1,988 malicious ads from which we could assign labels to 1,249. 1,552 of all found malicious ads were shown on benign websites. Our analysis shows that malvertising is a prevalent threat to users even on frequently visisted websites and also through often used ad networks. We manually inspected the content of malicious ads and identified common threats and deception techniques.

Some of the found deceptive measures require timely manual inspection or specialized automatic collection approaches (e.g., browser push notifications) to gain more detailed insights, which opens up potential for further enhancements.

Our ad crawler can be used for future automatic detection efforts and insights into currently used deception techniques and threats. It can be used to support user sensitization efforts and future developments of countermeasures.

Acknowledgments This work used academic access of VirusTotal, which was kindly provided for this project. In addition, this project used the freely available Google Safe Browsing API. Information from the GeoLite2 database created by MaxMind were used (available from https://www.maxmind.com).

Open Science All software and tools that we publish together with this work are available under: https://gitlab.com/rwth-itsec/malvertcollector

The repository contains the software used for crawling (MalvertCollector), additional tools used during analysis, and a web-interface used during the manual investigation to view the collected data. More detailed instructions for the usage of the tools is available in the repository. In addition, the dataset created during this measurement study will be made available and the repository contains instructions to access the dataset.

A Exemplary Deceptive Methods

In this section, we highlight selected examples of deception techniques.

Potentially Unwanted Applications (Browser Extension) A Potentially Unwanted Application (PUA) is software that is willingly installed by users but includes a potentially unwanted functionality. During our analysis we found ads for several browser extensions in this category and present one further. The ad promotes a browser-extension based speed test which can be directly installed from the ad target, but the mentioned browser extension was not listed in any of the official extension stores of the browsers. Small text below the install button of the extension informs the user that the installation would change the browser's default search engine and some "additional offerings" are provided, most likely generating some revenue for the extension's publisher. While this may not be illegal, using deceptive techniques like a very small text size, may result in unwanted changes.

Phishing (Adult Content) One of the largest groups of reported samples contains ads leading to explicit adult content. Some of these show identical content on differing URLs. The different vendors in VirusTotal classified most of the ad targets either as *phishing* or just as *malicious*. One example lured the user with explicit content but required a registration to access it. After a short questionnaire, the registration wizard asked for the current location before requesting login credentials. During the registration it is unclear for which service one registers or how to login later on. All tested credentials (even Gmail addresses) immediately led to an error that the email provider was not supported and the hint to use Gmail. As a password was asked in the same step, we suspect just credential farming. On another page, we also discovered a *Login with Google* button that actually invokes an OAuth flow, potentially enabling access to sensitive information.

Malware (Fake Anti-malware Software) We also discovered an ad target mocking the interface of a widely known anti-malware software. The website

depicts a window that appears to be popped over the web browser, but is just content shown on the website. The window shows the progress of an ongoing anti-malware scan with multiple warnings that malware was found. Dynamic content, like a slowly filling progress bar and notifications popping up on the lower end of the website, support the impression of an ongoing scan. We were able to visit the page, shortly before it was shutdown. We observed that after some time the scan was marked as completed with infections found and we were redirected to the landing page of the original manufacturer's website without any further explanation or any hints of an affiliate link to generate revenue. This redirection could be due to a cloaking mechanism and the use of an institutional IP address when visiting the website. We were unable to identify a malicious download, but reports from VirusTotal with the category malware underlines the presumed threat type.

Investment Fraud (Advertorial and Fake News) We identified multiple different advertorial campaigns that seem to fall into the category investment fraud. One campaign promoted an artificial intelligence powered investment service and mimic the look and feel of the well-known Swedish news agency "Dagens Nyheter" to give the impression of an editorially revised article. However, it was hosted under a domain not affiliated with the original agency in any way.

While advertorials may market legitimate products using questionable methods, mimicking a well-known news site constitutes a deceptive measure.

Browser Push Notifications (CAPTCHA and Download) The widespread "Completely Automated Public Turing Test to Tell Computers and Humans Apart" (CAPTCHA) is a common method to secure a service against automated attacks and well-known to users. The design and operating principle changed over time to compete with more modern attacks against these, resulting in greatly varying mechanisms and low user suspicion when confronted with new mechanisms. We encountered several CAPTCHA-like examples in which the user was deceived into permitting web push notifications. One example asks the user to click allow to prove the user is a human. The presented logo closely resembles the original reCAPTCHA logo and a permission popup for web push notifications is displayed. However, the screenshot shows a "soft ask" permission request, which is website content and invokes the browser permission request only after the user accepted [3]. Due to increased abuse of web push notifications, browser vendors started to silently drop the permission request when there was no user interaction first. We found similar attacks in other contexts, like a fake download page asking to click allow to continue the download.

Granted permissions allow to send push notifications with potentially malicious content. Pretending that granting this permission is part of a CAPTCHA, lures users into granting it and constitutes a deceptive measure. As the crawler is not capable to collect web push notifications, no statement on the content of the push notifications can be made. Subramani et al. showed in [37] that 51% of their collected push notifications were malicious, but the focus of their study relied on push notifications used as ads, not them being registered using deceptive measures.

References

1. Ahmad, Z., Casarin, S., et al.: An Empirical Analysis of Web Storage and Its Applications to Web Tracking. ACM Transactions on the Web **18**(1) (2023)
2. Arrate, A., González-Cabañas, J., et al.: Malvertising in Facebook: Analysis, Quantification and Solution. Electronics **9**(8) (2020)
3. Carboneri, A., Ghasemisharif, M., et al.: When Push Comes to Shove: Empirical Analysis of Web Push Implementations in the Wild. ACSAC '23, ACM (2023)
4. Cerf, V.G.: Guidelines for Internet Measurement Activities (1991), RFC 1262
5. Cook, J., Nithyanand, R., et al.: Inferring Tracker-Advertiser Relationships in the Online Advertising Ecosystem using Header Bidding. Proceedings on Privacy Enhancing Technologies **2020**(1) (2020)
6. DataForSEO: Top 1000 Keywords Searched on Google. https://dataforseo.com/free-seo-stats/top-1000-keywords/ (2024), [Accessed: 2025-08-22]
7. Draghetti, A.: Phishing Army - The Blocklist to filter Phishing! https://www.phishing.army/, [Last accessed: 2026-07-06]
8. Dwyer, C., Kanguri, A.A.: Malvertising - A Rising Threat To The Online Ecosystem. CONISAR '16, ISCAP (2016)
9. EasyList: EasyList. https://easylist.to/, [Last accessed: 2026-07-06]
10. Englehardt, S., Narayanan, A.: Online Tracking: A 1-million-site Measurement and Analysis. CCS '16, ACM (2016)
11. Ghostery: Manifest V3: The Ghostery Perspective. https://www.ghostery.com/blog/manifest-v3-the-ghostery-perspective/ (2021), [Last accessed: 2026-07-06]
12. Google: How to Report an Ad. https://support.google.com/google-ads/answer/7660847/, [Last accessed: 2026-07-06]
13. Google: Replace Blocking Web Request Listeners. https://developer.chrome.com/docs/extensions/develop/migrate/blocking-web-requests/ (2023), [Last accessed: 2026-07-06]
14. Hantke, F., Roth, S., et al.: Where Are the Red Lines? Towards Ethical Server-Side Scans in Security and Privacy Research. SP 2024, IEEE (2024)
15. Hill, R., Rolls, N., et al.: uBlock Filters. https://github.com/uBlockOrigin/uAssets/blob/master/filters/filters.txt (2025), [Last accessed: 2026-07-06]
16. IAB, pwc: Internet Advertising Revenue Report 2024. Tech. rep. (2025)
17. Iordanou, C., Kourtellis, N., et al.: Beyond Content Analysis: Detecting Targeted Ads Via Distributed Counting. CoNEXT '19, ACM (2019)
18. Keeling, W.: Seleium Wire. https://pypi.org/project/selenium-wire/, [Last accessed: 2026-07-06]
19. Kenneally, E., Dittrich, D.: The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research. SSRN Electronic Journal (2012)
20. Le Pochat, V., Van Goethem, T., et al.: Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. NDSS '19, Internet Society (2019)
21. MalwareBytes: MalwareBytes Threat Alert | endowmentoverhangutmost.com. https://www.malwarebytes.com/blog/detections/endowmentoverhangutmost-com, [Last accessed: 2026-07-06]
22. Masri, R., Aldwairi, M.: Automated Malicious Advertisement Detection Using VirusTotal, URLVoid, and TrendMicro. ICICS '17, IEEE (2017)
23. MaxMind: GeoLite2 City Database. https://dev.maxmind.com/geoip/geolite2-free-geolocation-data/ (2024), [Last accessed: 2026-07-06]
24. Minhava, D.: Super Agent - Automatic Cookie Consent. https://addons.mozilla.org/de/firefox/addon/super-agent/, [Last accessed: 2026-07-06]

25. Mozilla: Firefox. https://www.firefox.com/, [Last accessed: 2026-07-06]
26. Mozilla: GeckoDriver. https://github.com/mozilla/geckodriver/, [Last accessed: 2026-07-06]
27. Partridge, C., Allman, M.: Ethical Considerations in Network Measurement Papers. Communications of the ACM **59**(10) (2016)
28. Pooranian, Z., Conti, M., et al.: Online Advertising Security: Issues, Taxonomy, and Future Directions. IEEE Communications Surveys & Tutorials **23**(4) (2021)
29. Rafique, M.Z., Van Goethem, T., et al.: It's Free for a Reason: Exploring the Ecosystem of Free Live Streaming Services. NDSS '16, Internet Society (2016)
30. Rastogi, V., Shao, R., et al.: Are these Ads Safe: Detecting Hidden Attacks through the Mobile App-Web Interfaces. NDSS '16, Internet Society (2016)
31. Sakamoto, T., Matsunaga, M.: After GDPR, Still Tracking or Not? Understanding Opt-Out States for Online Behavioral Advertising. SPW '19, IEEE (2019)
32. Schmeiser, S.: Online advertising networks and consumer perceptions of privacy. Applied Economics Letters **25**(11) (2018)
33. Selenium Project: Selenium Python Binding. https://pypi.org/project/selenium/, [Last accessed: 2026-07-06]
34. Selenium Project: Selenium WebDriver. https://www.selenium.dev/documentation/webdriver/, [Last accessed: 2026-07-06]
35. Son, D.: Lumma Stealer Resurfaces After Takedown: New Stealth Tactics Target Users via Fake Cracks, CAPTCHAs & GitHub. https://securityonline.info/lumma-stealer-resurfaces-after-takedown-new-stealth-tactics-target-users-via-fake-cracks-captchas-github/ (2025), [Last accessed: 2026-07-06]
36. Statcounter: Search Engine Market Share Worldwide. https://gs.statcounter.com/search-engine-market-share/ (2025), [Last accessed: 2026-07-06]
37. Subramani, K., Yuan, X., et al.: When Push Comes to Ads: Measuring the Rise of (Malicious) Push Advertising. IMC '20, ACM (2020)
38. Urban, T., Tatang, D., Degeling, M., et al.: A Study on Subject Data Access in Online Advertising After the GDPR. In: Data Privacy Management, Cryptocurrencies and Blockchain Technology. Springer (2019)
39. Urban, T., Tatang, D., et al.: Measuring the Impact of the GDPR on Data Sharing in Ad Networks. ASIA CCS '20, ACM (2020)
40. urlscan GmbH: URL and Website Scanner. https://urlscan.io/, [Last accessed: 2026-07-06]
41. Vadrevu, P., Perdisci, R.: What You See is NOT What You Get: Discovering and Tracking Social Engineering Attack Campaigns. IMC '19, ACM (2019)
42. Xing, X., Meng, W., et al.: Understanding Malvertising Through Ad-Injecting Browser Extensions. WWW '15, WWW Conferences Steering Committee (2015)
43. Zarras, A., Kapravelos, A., et al.: The Dark Alleys of Madison Avenue: Understanding Malicious Advertisements. IMC '14, ACM (2014)
44. Zeng, E.: Adscraper: A Web Crawler for Measuring Online Ad Content. https://github.com/UWCSESecurityLab/adscraper (2020), [Last accessed: 2025-09-10]
45. Zeng, E., Kohno, T., et al.: What Makes a "Bad" Ad? User Perceptions of Problematic Online Advertising. CHI '21, ACM (2021)
46. Zhang, P., Oest, A., et al.: CrawlPhish: Large-scale Analysis of Client-side Cloaking Techniques in Phishing. SP '21, IEEE (2021)