Threats to Online Advertising and Countermeasures: A Technical Survey

MARK YEP-KUI CHUA, Nokia Canada, Canada
GEORGE O. M. YEE, Dept. of Systems and Computer Eng., Carleton University, Canada
YUAN XIANG GU, Irdeto Canada, Canada
CHUNG-HORNG LUNG, Dept. of Systems and Computer Eng., Carleton University, Canada

Online advertising, also known as web advertising or Internet marketing, is the means and process of promoting products and services on the Internet, and it has been one of the important business models for the Internet. Due to its lucrative nature and its large scale of adoption, it has also been a target for malicious parties with various attack aims such as getting a cut of online advertising revenues, obtaining a user's privacy, and spreading malware. Over the years, a great deal of research has been conducted on online advertising. Recently, the health of the online advertising ecosystem has become more of a concern for both advertisers and regular Internet users. Advertising budgets have been abused, and Internet users' privacy and security have been infringed. In this article, we broadly study threats to online advertising and trace the root causes from a systems point of view. Existing threat mitigation strategies are also reviewed and analyzed. To protect online advertising, which has been an essential funding source of many free Internet services, several challenges still need to be addressed, including the need for transparency of the advertising ecosystem and software vulnerabilities on the client-side. To overcome these challenges, we conclude by brainstorming some innovative ideas on some potentially interesting and useful research directions.

CCS Concepts: • **Security and privacy → Web application security**; **Domain-specific security and privacy architectures**;

Additional Key Words and Phrases: Online advertising, security, privacy, IoT, blockchain, software protection

ACM Reference format:
Mark Yep-Kui Chua, George O. M. Yee, Yuan Xiang Gu, and Chung-Horng Lung. 2020. Threats to Online Advertising and Countermea-sures: A Technical Survey. *Digit. Threat.: Res. Pract.* 1, 2, Article 11 (May 2020), 27 pages.
https://doi.org/10.1145/3374136

1 INTRODUCTION

Online advertising brings commercial information to Internet users. Advertisers invest in advertising campaigns to promote their products and services, and Internet users receive the commercial information while using the Internet applications and services sponsored by the advertisements (ads). These applications and services range

Authors' addresses: M. Y.-K. Chua, Nokia Canada, 600 March Rd, Kanata, ON, Canada; email: markchuaca@gmail.com; G. O. M. Yee and C.-H. Lung, Dept. of Systems and Computer Eng., Carleton University, 1125 Colonel By Dr, Ottawa, ON, Canada; emails: {gmyee, chlung}@sce.carleton.ca; Y. X. Gu, Irdeto Canada, 2500 Solandt Rd, Kanata, ON, Canada; email: yuan.gu@irdeto.com.

2576-5337/2020/05-ART11 $15.00
https://doi.org/10.1145/3374136

from search engine, social networks, smartphone applications to blogs, and others. Online advertising can have various types, such as display advertising, sponsored search advertising, and social network advertising, all with similar business models. Whether an advertisement is legitimate or not, once it is presented to potential consumers, money flows into the advertising ecosystem from the advertiser's pocket. And hopefully, a portion of the money will be used to support the Internet applications and services themselves. In this article, we use **publisher**, **advertiser,** and **consumer** to refer to the creator of an Internet application or service, the business owner using online advertising to promote his business, and the Internet user consuming the Internet applications or services, respectively.

Regretfully, as we have learned from the media and from our personal Internet experience, online advertising has many security and privacy challenges. From the advertiser's perspective, fraudulent practitioners can create various forms of fraud and significantly drain the advertiser's budget. From the publisher's perspective, malicious parties can inject advertisements into the publisher's platforms (blogs, applications, etc.). The injected advertisements bring no money to the publisher, and what is worse, if the content of an ad is not appropriate, the publisher's reputation could be ruined. From the consumer's perspective, security and privacy are big concerns. "Malvertising" refers to the use of online advertising to spread malware. The widespread occurence of malvertising is clearly a security concern. Targeted advertising infringes a consumer's personal private data without the consumer's awareness. Are there any solutions to these concerns? Would eliminating advertising on the Internet be the only solution? That is indeed why adblocking has become more popular among consumers. However, if there are no ads, who will finance the "free" applications and services? The authors of this work believe that although online advertising probably cannot be eliminated, it can nevertheless be made better with improved security and privacy.

Structurally, between consumers and advertisers, there are complex networks consisting of multiple players. These players can be well-known web companies such as Google and Facebook, but can also be less well-known entities [51]. Admittedly, without them, we probably would not have the Internet's prosperity today. However, arguably, it is also known that the online advertising industry, consisting of a wide range of participants without a single source of transparency, makes online advertising problematic from the security and privacy perspective. Over the past decades, researchers have been trying to address these issues as well as the lack of security and privacy. In our view, these are very challenging problems because of the following reasons: First, the advertising ecosystem is extremely complex and it is not very well-known. It is so complex and non-transparent that even important players in the industry such as Facebook can make mistakes; for example, purchasing a company that generates fraud [52]. In the academic arena, a thorough overview of the security and privacy issues from the perspectives of both consumers and advertisers is still needed, although there are many good works, including a survey paper [15] on online advertising. Second, as in many technological areas, the advertising platforms and technologies are evolving at a very fast pace. See Reference [15] for the technology evolution over time. Accordingly, new types of attacks emerge as the technologies evolve. Last, but most importantly, most of the existing research (see Sections 3, 4, and 5) target only one type of attack or only one aspect of the problem. In a nutshell, a holistic and in-depth understanding of the problem space is crucial to protect the online advertising ecosystem.

In this article, we intend to take the initiative to capture the state-of-art and pinpoint some possible research problems to address the security and privacy threats to online advertising and to discuss the related fundamentals. The first objective of this article is to provide a landscape of security and privacy threats to online advertising. Regretfully, the online advertising ecosystem is broken from the security and privacy perspectives, resulting in the ecosystem earning a bad reputation. This will eventually corrupt the entire online advertising industry if no effective actions are taken. This corruption has already started; for instance, see Reference [3] for the news that Procter & Gamble cut their online advertising expense because of the vulnerabilities inherent in the system. The second objective of this work is to highlight related research problems in web/mobile security at the fundamental level. As we will see, many attacks are generated by taking advantage of software

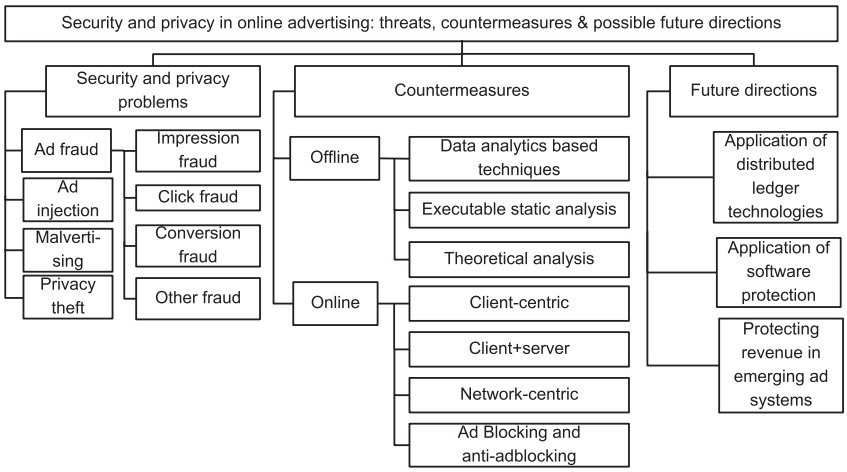

Fig. 1. General threats to online advertising, existing proposals to mitigate threats, and possible future research directions.

vulnerabilities, system vulnerabilities, and even flaws in the implementation of the Internet. Our results will hopefully be applicable to defending against future threats regardless of the type of system that will be vulnerable.

As depicted in Figure 1, this article covers an end-to-end technical overview from multiple dimensions of threats to online advertising and the state-of-the-art countermeasures, as well as provides innovative ideas to address these problems. This survey makes the following contributions:

- This is the first technical survey article covering the threats to online advertising with width and depth. Major threats and their motivations are discussed and the root causes are indicated.
- It overviews and contrasts the state-of-the-art proposals to prior works in addressing ad-related threats.
- It brainstorms the research directions to address the threats given the technology landscape, including disruptive technologies available in the future and software security fundamentals.

The rest of this article is structured as follows: We provide an introduction to the online advertising system in Section 2. Section 3 is dedicated to security and privacy threats and their mechanisms. The knowledge of attacks and potential attack vectors is crucial to understanding the security and privacy problems. Sections 4, 5, and 6 summarize the existing work on addressing these attacks. These proposals range from client-side techniques to data analytics. In Section 7, we outline several research directions taking into account the emerging technical trends and security/privacy requirements for protecting online advertising. Section 8 concludes the article.

2 BACKGROUND OF ONLINE ADVERTISING

Modern advertising systems are called *computational advertising*, in which the core purpose is to match a suitable advertisement with a specific user under a given context [1]. This section provides a brief introduction to online advertising to provide the basic concepts needed for analyzing the security and privacy issues in online advertising. We first provide a high-level overview of the online advertising ecosystem and the accompanying players. After that, we present a simplified ad delivery process. For readers interested in more detailed coverage, an in-depth technical survey of online advertising can be found in Reference [15].

2.1 Brief Overview of the Online Advertising Ecosystem

Figure 2 illustrates a common version of the Internet advertising ecosystem. Here, we intend to cover players and their interactions, even though, in practice, there can be variations from system to system. Essentially, the

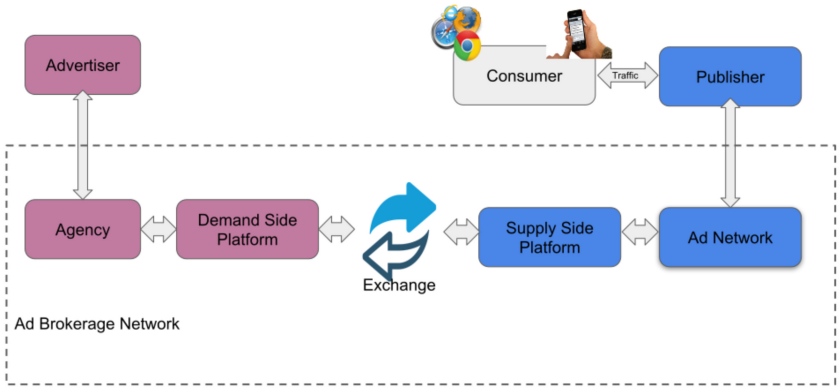

Fig. 2. Conceptual level description of computational advertising ecosystem.

demand side of the ecosystem can have Advertisers, Agencies, and Demand Side Platforms (DSPs). The supply side of the ecosystem consists of Publishers, Ad Networks, and Supply Side Platforms (SSPs). The demand and supply side players are in purple and blue blocks, respectively, as shown in Figure 2. Usually at the center of the ecosystem is an Ad Exchange, which acts like a broker between the supply side and the demand side. To refer to the plethora of intermediate beneficiaries precisely, we call them the **Ad Brokerage Network**. In many cases, there are interactions between multiple players and the interactions between demand side and supply side can occur at multiple levels. For instance, multiple DSPs can contract with multiple Ad Exchanges for traffic trading. In Table 1, we list the definitions for the players in the online advertising ecosystem based on the Interactive Advertising Bureau (IAB)'s documentation at https://wiki.iab.com. IAB, with headquarters in New York City, is an advertising business organization that develops industry standards, conducts research, and provides legal support for the online advertising industry [92]. To some extent, IAB is the de facto online advertising standardization body.

An important aspect of online advertising is the Real-Time Bidding (RTB) process. With RTB, Ad Exchanges aggregate the demand and supply in marketplaces and use an auction to sell an ad impression in real time when it has just been generated from a user visit. When a publisher site with ads is loaded onto a browser or when a mobile application is launched by an end-user, the web page or the app makes a request to the Ad Exchange. At the same time, the Exchange communicates with the buyer side of the system. The DSP has information of the targeted consumer. With the new request bid, the DSP will win out of the bidding. The information will be returned to the supplier side. In the end, the publisher will inject the ad to the opening HTTP(s) connection. This process happens in less than a second and the end-users will not even notice it. For more technical details regarding RTB, Wang et al. detail the technologies and processes in Reference [90].

Last, for readers interested in knowing the companies on the online advertising ecosystem, lumapartners.com publishes data on companies in each ecosystem subdomain in Reference [51]. Not surprisingly, big names, such as Google and Facebook, are listed together with many other companies that are not very well known to the public.

2.2 Ad Delivery Process: A Case of Display Advertising

Let us take a close look at the ad delivery process. The process begins when a publisher registers with an ad network. If the publisher is using a website for ad delivery, a snippet of JavaScript code will be added to this website. If the publisher is an author of a mobile application (app, for short), a mobile advertising software developer's kit (SDK) will be included into the build of the app. In Figure 3, the ad delivery process in the browser environment during run-time is shown. The difference between browser and mobile application environment is that instead of JavaScript snippets, a mobile ad SDK is the main carrier of the process.

Table 1. IAB's Definition of Players in Online Advertising Based on https://wiki.iab.com

Player	Brief Definition
Advertiser	A person, organization, or company that places promotions of a specific product, service, or event in a public medium to attract potential new or repeat customers.
Agency	An organization that, on behalf of its clients, plans marketing and advertising campaigns, drafts and produces advertisements, places advertisements in the media. Agencies often use third-party technology (ad servers) and may place advertisements with publishers, ad networks, and other industry participants.
Demand Side Platforms (DSPs)	Organizations that provide centralized (aggregated) media buying from multiple sources including Ad Exchanges, Ad networks, and Supply Side Platforms, often leveraging real-time bidding capabilities of said sources.
Ad Exchanges	Organizations that provide a sales channel to Publishers and Ad Networks, as well as aggregated inventory to Advertisers. They bring a technology platform that facilitates automated auction-based pricing and buying in real-time. Ad Exchanges' business models and practices may include features that are similar to those offered by Ad Networks.
Supply Side Platforms (SSPs)	Organizations that provide outsourced media selling and ad network management services for publishers. Also known as sell-side platforms. Their business models and practices are similar to Ad Networks. SSPs are typically differentiated from Ad Networks in not providing services for Advertisers. DSPs and Ad Networks often buy from SSPs.
Ad Networks	Organizations that provide an outsourced sales capability for publishers and a means to aggregate inventory and audiences from numerous sources in a single buying opportunity for media buyers. Ad Networks may provide specific technologies to enhance value to both Publishers and Advertisers, including unique targeting capabilities, creative generation, and optimization. Ad Networks' business models and practices may include features that are similar to those offered by Ad Exchanges. Usually, ad networks rely on Content Delivery Networks to server advertising contents.
Publisher	A person or company that makes content (in any form) available for consumption, for free or for sale.
Consumer	Potential user or buyer of products and services advertised. A limited amount of anonymous data, or non-personally identifiable data, is passed between the browser or the application and the Publisher to help the Publisher identify a machine that has connected to its website via the use of a session cookie.

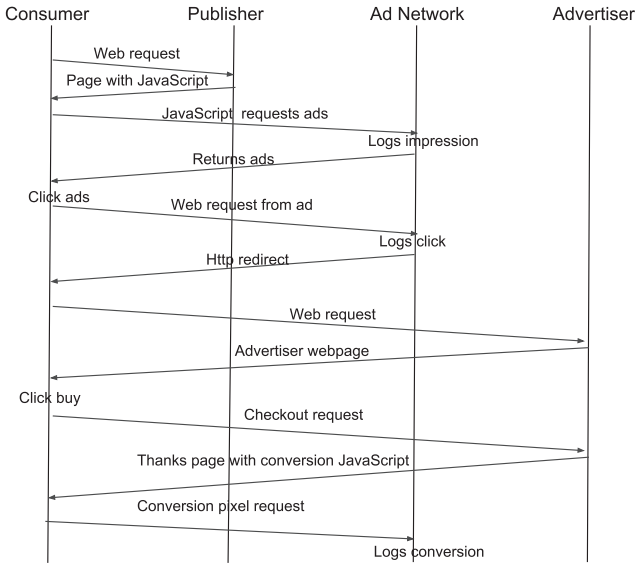

Fig. 3. Ad delivery process: impression, click, and conversion [26].

Table 2. Security and Privacy Requirements of the Major Advertising Ecosystem Stakeholders

Stakeholder	Online advertising security and privacy requirements
Consumer/User	Privacy and being secured against threats such as malvertising.
Advertiser	Effective advertising campaigns—a fraud-free ad ecosystem.
Publisher	Being reputable and increasing ad revenues.
Brokerage network	Sustainable business model and following regulations.

For other forms of online advertising, such as social network advertising and search advertising, the flow can be slightly different. For example, in search advertising, the sponsored ad is usually served by the same domain as the search engine. However, the vulnerable points identified in Section 3 are still valid.

3 SECURITY AND PRIVACY PROBLEMS RELATED TO ONLINE ADVERTISING

In this section, we introduce the security and privacy problems in online advertising. We start with a brief summary of the security and privacy requirements of major stakeholders of the online advertising ecosystem. Next, we list the typical threats and techniques attackers can leverage.

In Table 2, we capture the requirements of major stakeholders in the online advertising ecosystem with a focus on security and privacy in general. In many cases, the requirements can be mutually contradictory. Here are some examples: Between consumers and brokerage networks, consumers' increasing privacy concerns make data collection very difficult for brokerage networks. While for publishers, increasing advertising revenue can motivate them to introduce fraud traffic, which violates the advertisers' desire for a fraud-free ad ecosystem.

Like many cyberattacks, attacks against online advertising can be complex, and as we see in the following, there can be multiple attack vectors and attack surfaces in this systematic problem. We collectively cover ad fraud, ad injection, privacy theft, and malvertising. To our knowledge, these categories cover all the attack scenarios that have been revealed in the public literature to date. We then continue the discussion by investigating two essential dimensions of the ad attacks: where attacks happen and the infrastructure leveraged by attackers.

3.1 Ad Fraud

Advertising fraud, also called ad fraud and ad spam, is the most widely deployed attack on online advertising. Here, fraudulent actors maximize the amount of money from the ad ecosystem, regardless of the presence of a valid audience. Extensive research on this attack (for instance, References [81] and [37]) has been conducted both in academia and in industry. The book by Zhu et al. [100] reviews methods of ad fraud and the techniques to prevent it. Ad fraud is categorized into placement fraud, traffic fraud, and action fraud. Measures and resources to detect each type of fraud are also discussed in the book. In the advertising industry, IAB published a white paper on ad anti-fraud principles and taxonomy in 2014. This is the first time the industry openly requires participants in the space to take ad anti-fraud into account. In the white paper, four principles, including Fraud Detection, Source Identification, Process Transparency, and Building Accountability, are proposed [37]. In the following, we will briefly introduce various types of fraud, including impression fraud, click fraud, conversion fraud, cookie stuffing, and others. These terms are used by most works on ad fraud detection and prevention.

3.1.1 Impression Fraud. This is the simplest form of ad fraud. It involves generating HTTP requests to hit either the publisher's page, or the ad server directly, to artificially increase the actual amount of traffic. Impression fraud targets the Cost Per Impression pricing model, in which, by definition, the advertiser is charged based on how many times its product is viewed through ad traffic.

3.1.2 Click Fraud. Click fraud is probably the most common ad fraud and also the most studied because of the widely adopted Cost Per Click pricing model, in which advertisers are charged based on the number of clicks on their products. As its name indicates, click fraud is done by generating HTTP requests to advertisement click URLs, usually after an ad is shown. There are two kinds of click fraud [81]. *Click inflation* is where publishers make more money than they should by creating traffic using artificial clicking. *Competitor clicking* is a different approach where advertisers make false clicks in a competitor's ads to consume the competitor's advertising budget.

3.1.3 Conversion Fraud. When a visitor follows a click to the advertiser's website, the click is called a conversion if the visitor purchases the product. The conversion is fraudulent if the visitor did not in fact make the purchase but the publisher only made it appear to be a conversion to receive the credit. Conversion fraud only works if the action does not require spending money directly.

3.1.4 Other Frauds. Cookie stuffing or cookie dropping is an activity in which illegitimate actors receive credit for purchases made by web users by defrauding affiliate marketing programs, even if the affiliate marketer did not actively perform any marketing for the affiliate program [76]. The defrauding is carried out by the illegitimate actor depositing a cookie (hence the term "cookie stuffing") on the user's computer for a target website that is unrelated to the one the user is visiting. The illegitimate actor then receives credit when the user later visits the target website. The user is usually unaware of the cookie stuffing. By mining the HTTP logs in a university web server, Snyder et al. [76] found that over one-third of publishers in affiliate marketing programs use cookie-stuffing to claim credit from online retailers for illicit referrals. In addition to cookie stuffing, a new ad fraud mechanism enables publishers to increase their ad revenue by deceiving the ad exchange and advertisers. This new scheme (described in Reference [54]) targets users visiting the publisher's site using higher-paying ads. By issuing requests to content not explicitly requested by the user, the user's online interest profile is corrupted and the ad selection process is influenced.

3.2 Ad Injections

Threats to online advertising involving tampering are usually in the form of ad injection, which has a direct impact on a publisher's revenue. In theory, the Internet follows the end-to-end principle [71], which implies that

intermediary boxes between two communicating processes should not alter the application messages. Tampering of ads, like any tampering of Internet flows, can be considered as a violation of this principle [19].

The ad delivery process is through client-server communications, which is mostly via HTTP. Therefore, tampering in the end-to-end tunnel can be conducted to alter ad contents and their delivery process. Although HTTPS can protect general Man-in-the-Middle (MitM) attacks, proxies at HTTPS endpoints can still modify web pages unknown to the client. What is even worse is that there is still a large number of websites that have not deployed Transport Layer Security (TLS). Ad tampering can happen at the client-side or in the network. For monetization, the injectors syndicate from the large ad ecosystem.

The cases where browser sessions are tampered with, such that tampered ads are imposed on the browsers, are discussed in Reference [85]. To discover ad injection and the value chain behind it, a client-side DOM scanner was deployed in a subset of Google's websites. The scanner is able to detect and report rogue ad elements. With such client-side telemetry technique, the paper reports that more than 5% of unique daily IP addresses accessing Google were impacted by ad injection during the time frame of the study. Furthermore, injected ads were identified to arrive at a client's machine through multiple vectors, and the measurements identified 50,870 Chrome extensions and 34,407 Windows binaries, 38% and 17% of which were explicitly malicious, respectively.

Many legitimate applications use advertisements to earn money while providing the application to users for free. However, malicious applications can take advertising a step further with invasive advertising practices. Rather than placing advertisements alongside legitimate application content, malicious adware will display advertisements when the user is interacting with other applications. This could significantly interfere with a user's experience with other applications (see invasive advertising [30]).

Notably, ad injection can also ruin a publisher's reputation. For example, it would be a disaster for a publisher if politically sensitive advertisements were injected into the publisher's website.

3.3 Privacy Theft

Privacy is secrecy applied to personal data [80]. More precisely, privacy refers to an individual's ability to control the collection, purpose, retention, and distribution of information about himself or herself [95]. Here, collection refers to who can collect the information, purpose is the use that will be made of the collected information, and retention is the amount of time that the collector will retain the information. Distribution refers to the collector sharing the information with other parties. In the online advertising world, a user would most likely want to protect her privacy while accessing publishers. This means that the user would want to retain control or have her preferences respected regarding the collection, purpose, retention, and distribution of her personal information while accessing the publishers.

Privacy theft can and does occur by publishers and/or ad networks simply mistreating the user's personal information, such as selling the user's physical address and phone number to other parties without the user's consent. However, a more stealthy form of privacy theft is the following: To increase its effectiveness, online advertisers have been employing a sophisticated mechanism known as Online Behavioural Advertising (OBA) [46]. This mechanism collects data about a user's online activities (e.g., visits to motorcycle websites), builds models inferring the user's interests (e.g., motorcycles), and then displays ads that target the user's interests (e.g., motorcycle ads). Thus, OBA is able to benefit advertisers by increasing the click-through rates. In fact, OBA has been prevalent. For instance, measurement and analysis on OBA on 1M sites is conducted in Reference [29] with the help of an open-sourced automated framework built on top of Selenium Web driver and machine learning algorithms. On the mobile counterpart, sensitive demographic information and personal interest are shown to be correlated to the mobile ads delivered in the users' mobile applications [53].

Even though some users may like receiving more relevant ads, many users are concerned about being tracked. The corresponding loss of privacy is the user's loss of control over the collection, purpose, retention, and distribution of the user's personal interest information. Regulations do exist on the kinds of targeting data and tracking

techniques that the online advertising industry can use [32]. Despite this, the concern for the attendant loss of privacy has been increasing. Take Facebook's OBA platform as an example. A vulnerability on the Facebook OBA platform is reported in Reference [87]. In Facebook's OBA targeting practice, users' personal identifiable information (PII) is collected and advertisers are allowed to select which users see their ads. Traditionally, the targeting is through attributes. More recently, advertisers can upload personal information of the targeted audience to Facebook's OBA platform to have more specific targeting. The authors of Reference [87] found out that adversaries can infer users' full phone numbers with the targeted users' email addresses by abusing Facebook's OBA platform.

3.4 Malvertising

Online advertising has been exploited by cybercriminals to spread malware. This is known as malvertising, a form of spreading malware to vulnerable devices through online advertising [30]. Take browser-based advertising as an example. In this case, a user navigates to a website that contains an external advertising link in the form of an injected iframe that directs the user's browser to an invisible **exploit kit** landing page. Exploit kits are web-based services designed by hackers to exploit vulnerabilities of browsers by downloading malicious executables. At that point, information about the victim's system is passed along to the attacker's server, which is then used to select a malicious exploit file that is automatically downloaded. The downloaded file exploits a vulnerability on the system that allows the attacker to install a malicious binary or otherwise control the victim's machine. This process is the so-called *Drive By Downloads*—one way by which malvertising works.

From a hacker's point of view, what he needs to do is submit malicious ads to the ad ecosystem. In fact, some ad exchanges are more prone to serving malicious advertisements than others [97]. Some recent infamous malvertising operations are AdGholas [21] and Ramnit [72], which can infect general computer browsers and Android smartphones, respectively. Even with ad networks and ad exchanges working to filter malicious ads, attackers can use many techniques to bypass detection, including fingerprinting, redirection, just-in-time assembling and compilation, obfuscation, and timing-based evasion.

From end-users' perspective, even desktop computers with updated anti-virus protection can to some extent prevent damage due to malvertising. For many smart devices such as smart phones, many of them have not been equipped with anti-virus protection. So, we can expect that the impact of malvertising is still considerable.

3.5 Where Are the Battlefields

In this subsection, we discuss where attacks against online advertising can be launched. Knowing where attacks are launched is important to identify possible threats and security weaknesses in the system and propose solutions to harden it. The ad's journey starts with an HTTP request to an ad server and ends when the ad is rendered in either a browser view port or a mobile application. Accordingly, ad attacks can happen either at the viewer's end or on the network.

3.5.1 At the Internet Endpoints. Advertisements are rendered in either a web browser or a mobile App. That means Man-in-The-Browser (MiTB) attacks and its mobile counterpart Man-in-The-Mobile (MiTMo) attacks are applicable to online advertising [13, 62]. In fact, malware authors are pretty much incentivized to create malware that will infect web browsers and smart phones for various ad-related attacks such as fraud [30], ad injection, and malvertising.

With MiTB attacks, the web browser is infected by malware either in the form of a browser plugin [65] or modifying the browser binary via another piece of malware. In this case, the browser can behave maliciously, including launching many attacks in the background for creating ad fraud, modifying the presented ads (ad injection), downloading malicious files from the Internet (malvertising), and collecting personal private data (privacy theft). In the mobile environment, the same cases apply for both native applications and mobile web browsers. Moreover, when the mobile operating system is infected by malware, more invasive behaviors are observed. For

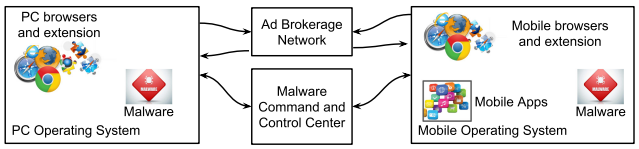

Fig. 4. Man-in-The-Browser (MiTB) attacks and Man-in-The-Mobile (MiTMo) attacks.

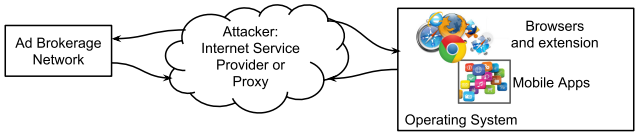

Fig. 5. Man-in-The-Middle (MiTM) attacks.

instance, an empirical study of click fraud is conducted in [18] on the Android platform, for which an automated click generation attack was developed, launched, and evaluated on eight popular advertising networks. It was shown that six advertising networks (75%) out of eight (Millennial Media, AppLovin, AdFit, MdotM, RevMob and Cauly Ads) were vulnerable to the attacks during the time frame of the research. Two fraudulent ad behaviors in applications are revealed in Reference [23]: (1) requesting ads while the application is in the background, and (2) clicking on ads without user interaction.

In Figure 4, we illustrate MiTB and MiTMo attacks. In most cases, there is a malware command and control center that instruments the infected operating system, browser, or application to launch various illegitimate actions against online advertising. In the end, part of the online advertising revenue will be taken by the parties running these attacks.

3.5.2 In the Network. As shown in Figure 3, the data flowing between the consumer, the publisher, and the brokerage network can be targeted by malicious parties. This is a typical Man-in-The-Middle (MiTM) [22] attack scenario. In the MiTM attack, the malicious Internet Service Providers (ISPs) or HTTP proxies can alter or inject ads into the HTTP flows. In Figure 5, we show the MiTM attack setup in online advertising. It is known that an ISP can inject contents into the network traffic and this can happen at both the edge-ISPs (direct Internet access providers to users) and non-edge ISPs [57]. In research conducted by Reis et al. [68], widespread and diverse changes made to web pages between the server and client are discovered. Over 1% of web clients in their study received altered pages. The changes included pop-up blocking scripts inserted by client software, advertisements injected by ISPs, and even malicious code likely inserted by malware using ARP poisoning. Note that the measurement was based on a website over HTTP.

With Transport Layer Security (TLS), MiTM is not as easy as before. However, it is still doable. In fact, the usage of TLS proxies is not negligible. In Reference [60], 1 out 250 TLS connections are TLS-proxied during a study of 2.9M tests. A TLS proxy acts as an endpoint of TLS connections. In particular, A TLS proxy can issue a substitute certificate for any web server the user visits; as a result, the user establishes an encrypted connection to the proxy rather than the desired web server. This means the proxy can decrypt and modify the HTTP payload including ads [60]. Besides HTTPS proxies, other attacks are possible. For instance, a Man-In-The-Middle-Script-In-The-Browser (MITM-SITB) attack is presented in Reference [43]. In the attack, malicious JavaScript is sent to the user's browser within a TLS connection with the attacker, and this JavaScript is used in direct connections. In such cases, the boundary between MiTM and MiTB/MiTMo is blurred.

3.6 What Malicious Parties Can Use

Another important aspect of understanding attacks concerns the tools and infrastructure that attackers have to achieve their goals. For advertising attacks, hackers have utilized malware, botnets, and click farms.

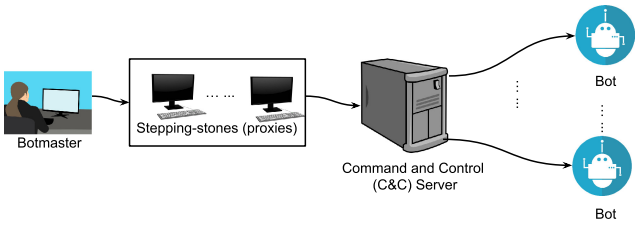

Fig. 6. Typical botnet architecture: botmaster, stepping-stones, C&C server and bots [44].

3.6.1 Malware and Botnets. Malware gains access to a computer system (personal computer, smartphones, IoT devices, and so on) for purposes such as stealing data, damaging the device, or simply annoying the user [30]. Hence, malware is a popular means to create attacks against online advertising. These threats include ad fraud, ad injection, and stealing web cookies. A botnet refers to a network of computers that have been infected by malware and turned into bots. Figure 6 illustrates a typical botnet architecture. A botmaster can remotely control a large network of bots (i.e., infected computers) to execute attacks. Bots receive commands from the attacker's Command and Control (C&C) server [78] and can be turned into machines creating click fraud (which are then known as clickbots [24]). Through the C&C, bots can also be turned into machines that fetch ads from the Internet unknown to the user. Botnets take advantage of vulnerabilities in web browsers and spread malware in the form of browser extensions [65].

Botnets and their operations are complex and are research topics in cybersecurity. Surveys on botnets can be found in the literature, such as Reference [74]. By far, public literature on botnet-related ad security threats focuses only on ad impression fraud and ad click fraud, which we discussed in Sections 3.1.1 and 3.1.2, respectively. The characteristics of the ZeroAccess botnet is studied in Reference [63] based on real data. Methods, such as Reference [42], have been proposed from industry to detect bots. Click fraud generated via misdirecting real human clicks is studied in Reference [5]. The key element behind the scheme is DNS changer malware that changes the DNS resolver setting on victim machines to attacker-controlled revolvers in Eastern Europe. Reference [12] provides a 2012 survey on click-fraud malware used for monetization. The details of the Hapili malware mechanism can also be found there.

Several companies also have released white papers regarding their research on how botnets operate to get a cut of ad revenue. For instance, Spider.io [79], which has been acquired by Google, demonstrates how to use the Zeus malware to impersonate real website visitors and visit ad-supported websites. Fraudulent publishers can buy the traffic generated by the malware and resell the traffic to players in the ad ecosystem such as SSPs or Ad Networks (discussed in Section 2). In Reference [91], WhiteOps, a major player in the ad anti-fraud space, exposes the operations of a Meth botnet. Similar to Zeus and revealed in Reference [79], infected machines would launch browsers to create ad impressions and clicks. The report also lists other similar botnets that monetize ad fraud.

3.6.2 Click Farms. Despite the absence of a thorough study on click farms, i.e., organized activities to generate illegitimate ad clicks and conversions, ad fraud research such as Reference [26] does assume click farms to be one of the sources of fraud.

3.7 Putting It All Together

In this section, we have introduced the threats and threat mechanisms that target online advertising. A brief analysis of the technical causes of these issues is shown in Table 3. In a nutshell, the vulnerabilities of online advertising are mainly due to the lack of ad ecosystem transparency and the weak client-side security of the web.

Indeed, philosophically, the technological causes of threat are mainly due to human or business aspects of weakness or negligence. Generally speaking, most security issues can be at least partially addressed by enforcing

Table 3. Technical Cause of Online Advertising Threats

Threat	Technical cause
Fraud	Non-transparent ecosystem particularly in traffic demand-supply trading; Weak client-side security
Ad injection	Weak client-side security and violation to the End-to-end principle of the Internet
Privacy theft	Weak client-side and server-side security and weak regulation in the industry
Malvertising	Weak client-side security and weak regulation on brokerage networks

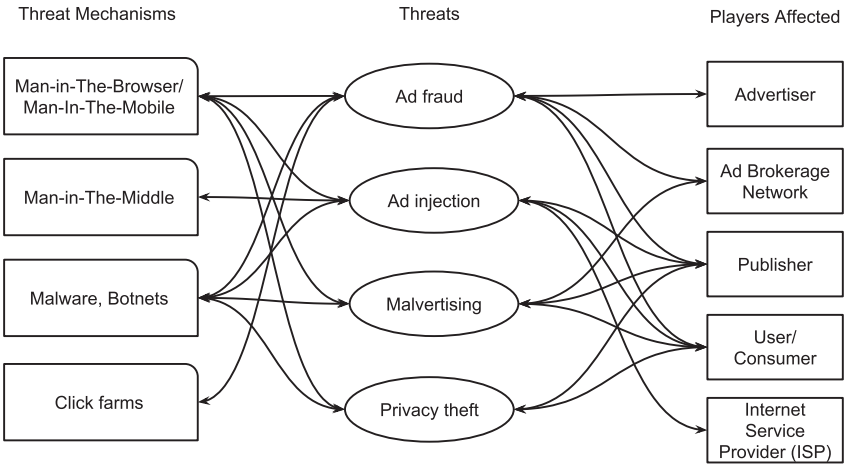

Fig. 7. Linkage between attacker methodologies, threats, and participants.

regulatory frameworks in the industry. However, because of the cost and the lack of awareness, the current online advertising security framework does not seem to be sufficient to fight the threats.

In Figure 7, we identify the relationship between attack methodologies, threats, and related players. Consider ad fraud as an example. Ad fraud is related to advertiser, ad brokerage network, publisher, and consumer. To conduct ad fraud, the attacks can be MiTB/MiTMo attacks. Malware, botnets, and click farms can be used to conduct ad fraud.

Among the attack methodologies, MiTB/MiTMo and malware/botnets can be applied to all advertising-related threats. The MiTM attack and click farm are only applicable to ad injection and ad fraud, respectively. Among the stakeholders shown on the right side of Figure 7, publishers are related to all kinds of threats. Specifically, publishers can have the motivation to create ad fraud and privacy theft. At the same time, ad injection would endanger the publishers' interests, since the injected ads will not bring the publishers any revenue; while malvertising can harm the infected publisher's reputation. Consumers, in particular, are impacted by all threats, which also implies that client-side software is the most vulnerable part of ad delivery infrastructure.

4 OFFLINE MITIGATION STRATEGIES AND COUNTERMEASURES

To address threats discussed in Section 3, many mitigation proposals have been raised in the literature. Based on how and when the countermeasures kick in, we can classify them into two categories—offline and online countermeasures. Offline countermeasures intend to prevent and detect the security issues prior or after the advertising delivery process. In contrast, online countermeasures aim at eliminating security issues during the

ad delivery process. The two types complement each other. In this section, we focus on the first category and will discuss the second category in the next section.

4.1 Data Analytics–based Techniques

Data analytics, especially machine learning, has been widely used in detecting ad fraud and malvertising. Prior to data analytics, ad networks usually used ad hoc or simple heuristics to filter out ad fraud. For example, if an advertiser complains of getting thousands of clicks from a single IP address, none of which convert into paying customers, the ad network may start filtering all clicks from that specific IP address or the whole area where the IP address is located. With more and more sophisticated fraud and tampering, the application of more advanced data analytical technologies and algorithms has been proposed.

Reference [59] reports some findings from a data-mining competition in 2012 that used anonymous production data. The task in the competition was to identify fraudulent publishers who generated illegitimate clicks and to distinguish them from normal publishers. One of the main findings is that features derived from fine-grained time-series analysis are crucial for accurate fraud detection. Further, methods consisting of multiple classical data-mining techniques provide the most promising solutions to highly imbalanced nonlinear classification tasks with mixed variable types and noisy/missing patterns. Note that the competition happened before the recent trend of applying deep learning to similar tasks [45]. The fraud detection results may be potentially improved with deep learning, especially in regard to the feature engineering capability. Recently, the application of neural networks in detecting fraud has been reported by Badhe in References [8] and [9], but some details are unfortunately missing.

Based on the observation that publishers using click fraud have a higher return of investment (ROI) than ethical publishers [26], a framework called Viceroi has been proposed. Viceroi consists of an offline component and an online component. In the offline part, click logs are analyzed over multiple timescales to identify click fraud and regions in the associated revenue per user distribution that are anomalous. In the online part, whether or not a given click falls in the anomalous region is identified. The underlying principle is that publishers using click fraud have higher ROI than ethical publishers. In a sense, this higher ROI justifies the higher risk the click fraud must carry, regardless of the specific mechanism the click fraud uses to commit fraud. In Reference [25], a proactive methodology is discussed for ad networks to detect click fraud. The data are collected through a special publisher website that collects information such as mouse movements. A fake ad is added to the website to trick fraudulent parties into clicking it. To deal with false positive and false negative cases, a Bayesian approach is used in the data analytics phase. Similarly, a Bayesian-based framework is also employed in Reference [63] to filter fraud from click streams. Click fraud prediction without knowledge of the ground truth is discussed in Reference [11]. The work is based on the fact that predictions of these ground models are not the same as the unknown ground truth and that the automatically generated class labels are inherently uncertain. The paper suggests that supervised learning from automatically labelled data should be complemented by an interpretation of conflicting predictions between the new classifier and the ground model. In Reference [66], 15K ads were observed using static analysis and behavior analysis. The data are defined in nine features and are fed into a Support Vector Machine (SVM) to detect malvertisements on the publisher side. A malvertising detection system, called MadTracer, potentially deployable in ad networks, is proposed in Reference [48]. MadTracer actively crawls the web's ad delivery processes and leverages a statistical learning framework based on decision trees to automatically generate a set of detection rules.

4.2 Executable Static Analysis

Static analysis of mobile applications, mobile advertising SDKs, and malicious executables used in malvertising can detect security issues before they happen.

In mobile advertising, to enable ad services in mobile applications, developers need to include third-party libraries in their applications, and these third-party libraries are usually binary-only. This has very important

security and privacy implications, which are that the ad library has the same permission as its hosting applications. This means that when a user grants the permissions required by the application during installation, the ad libraries essentially gain the same permissions in the system. To prevent these issues, mobile application and ad library static analysis is the first line of defense. For malvertising, an exploit kit is a malicious executable uploaded to a victim's computer. Static analysis can be used to detect malicious behaviors.

The privacy and security of ad libraries in Android is studied in Reference [33] by collecting 100K applications. The work consists of a system identifying potential risks ranging from collecting users' sensitive data to fetching code from the Internet. This indicates that application stores need to enforce more restrictive regulations on applications with embedded ad libraries. An analysis tool, MAdFraud, is developed in Reference [23]. It runs many applications simultaneously in emulators to trigger and expose ad fraud. It consists of the following three steps: building HTTP request trees, identifying ad request pages using machine learning, and detecting clicks in HTTP request trees using heuristics. The paper reports a study of over 130,339 applications crawled from 19 Android markets, including Play and many third-party markets, and 35,087 applications that likely contain malware and were provided by a security company. It is reported that about 30% of applications with ads make ad requests while running in the background, while 27 applications generate clicks without user interaction.

To automatically discover various placement frauds in Windows-based mobile platforms, a system called DECAF is revealed in Reference [49]. DECAF explores the UI state transition graph of mobile applications. It uses automated application navigation and optimizations to scan through numerous visual elements within a limited time and can detect whether ads within an application violate an extensible set of rules that govern ad placement and display. To better understand the malware executable used in malvertising, an automated system simulating high-risk browsing activities is proposed in Reference [70]. The proposed system crawls over the Internet for a period of seven days and collects more than 800 malicious executables.

4.3 Theoretical Analysis

The theoretical analysis of online advertising security can reveal some insights and provide some possible directions for practical solutions. A large part of the work involves the use of game theory, applied to analyze the interactions between different players in the advertising ecosystem, and between defenders and attackers.

Many have assumed that if an ad network such as Google reimbursed the advertiser for clicks generated by fraud, the ad network would lose money. This indicates that the ad network has no economical incentive to fight against fraud. The game theory model consisting of publishers, ad networks, and advertisers in Reference [55] reveals an opposite view: Letting fraud go unchecked is suboptimal for ad networks. However, what is not discussed is whether fighting fraud is the first priority for most ad networks. We think the answer has to be considered on a case-by-case basis. Focusing on the botnet driven ad fraud problem, a game theoretic model is proposed in Reference [89]. In the game model, the two players are ISPs and ad networks. The main finding is that under some conditions, ad networks cannot solve the ad fraud problem themselves and need to subsidize ISPs to achieve the goal. However, whether the ad network is incentivized or not, addressing the ad fraud problem is another issue. A similar problem is analyzed in Reference [28] with a more advanced economic model—namely, the Hotelling Competition-based Game-theoretic model, in which more factors are considered.

References [35] and [36] address the issue of malvertising using game theoretic modelling. The game consists of the ad network (defender) and the malvertiser (attacker). The effectiveness of the model needs more verification using real-world experiments. In particular, Reference [36] employs a Bayesian game model, because the ad network has only incomplete information regarding whether it is facing a benign or malicious advertiser.

5 ONLINE/RUN-TIME COUNTERMEASURES

Online or run-time countermeasures kick in during the ad delivery process. There are proposals that aim to protect advertisers, publishers, or consumers for each web access. In the literature, including the literature on

patents, there are client-centric, client-server cooperative, and network-centric proposals. In addition, in this section, we also discuss the third type of online countermeasures—namely, adblocking and anti-adblocking, which are a new controversial sub-field of online advertising. We present a survey and an analysis for the above types of proposals from a technical perspective in the following.

5.1 Client-centric Proposals

Currently, there are two types of clients in ad delivery: web browsers and mobile applications. The execution environments and integration technology for these client types are very different. Accordingly, there are proposals to enhance security for these two types of clients, respectively.

5.1.1 Client-centric Proposals for Web Browser Environments. Tripwire [68] is a client-side solution implemented in JavaScript to detect modification to a web page over HTTP. At the time of the work, HTTPS was considered an expensive security mechanism, and Tripwire was considered as an alternative to HTTPS for ad delivery. Today, this assumption is no longer valid. The HTTPS protocol can be achieved at very low cost, for instance, via letsencrypt.org. Nevertheless, the work is novel among client-side approaches to provide basic integrity checks for web servers. The weakness of Tripwire is that MiTB attacks are not covered. Basically, a browser extension or any malicious code in a browser can still tamper with Tripwire and, in the end, the web page that Tripwire intended to protect. The scheme also suffers from a lack of a trusted communication channel. As a result, adversaries with sufficient privileges can strip out integrity verification code and spoof a valid response.

Similarly, a solution for publishers and ad servers to build a secure ad serving system based on authenticated hash-chains is proposed in Reference [88]. A basic operation of the proposal is to compute hash values of web pages. However, as websites become more dynamic, computation of web pages is not as relevant as when the research was done. In contemporary websites or web applications, the only way to measure integrity is to compute the hash value of the DOM (Document Object Model) tree that is rendered by the browser. A series of tests on web browsers to detect bot use (replacing humans) is developed in Reference [93]. JavaScript snippets are developed for testing the browser with mouse event tests, functionality tests, and behavior tests. The advertiser's server will challenge the client using this series of tests. If the client fails during the testing, the request from the client is considered as invalid.

To protect users from threats such as malvertising, AdSentry is proposed in Reference [27]. It is a browser-based solution targeting JavaScript-based advertisements. With the help of a shadow JavaScript Engine, AdSentry enables flexible regulation on ad script behaviors by completely mediating the ad script's access to the web page (including its DOM) without limiting the JavaScript functionality exposed to the ads. AdSentry includes a policy enforcer, which allows both web publishers and end-users to customize the access policy for ads. For web publishers, if the ads are wrapped with special JavaScript variables, the ads' executions are confined to the shadow JavaScript engine. For end-users, AdSentry can leverage adblockers to automatically identify ads and confine them with a customized JavaScript wrapper.

5.1.2 Client-centric Proposals for Mobile Advertising. With insufficient regulation, it is possible that legitimate advertisement service providers can leverage ad libraries to track users without their knowledge, access sensitive data, and execute malicious code downloaded via ad delivery.

A framework called *AdDroid* is proposed in Reference [64] to separate out the privilege associated with an ad library in Android applications. The proposed framework provides a special API to application developers for advertising so advertising-related API calls will no longer have the same privilege as the application itself. Based on ARM's TrustZone technology, a verifiable mobile ad framework called AdAttester is proposed in Reference [47]. It consists of two security primitives: unforgettable clicks and verifiable display. ARM's TrustZone hardware root of trust is leveraged to implement these two primitives to collect proofs, which are piggybacked on ad requests to ad providers for attestation. Based on the observation that a malicious application could simulate the behavior of advertising libraries, a scheme call AdSplit is proposed in Reference [73]. In the

proposal, the Android operating system is modified to allow an application and its advertising to run as separate processes under separate user identifiers, eliminating the need for applications to request permissions on behalf of their advertising libraries and providing services to validate the legitimacy of clicks, both locally and remotely. However, whether this proposal will be accepted by the Android community and ecosystem is unknown.

Finally, FCFraud [39] is an operating system-level proposal to address click fraud. FCFraud consists of a Linux kernel module that constructs HTTP request trees based on publicly available ad-related domain names. It also monitors the hardware (mouse movement and clicks). Once an inferred click from HTTP request trees is not triggered by a hardware click, the request is logged as click fraud.

5.1.3 Client-centric Proposal to Improve Online Behavioral Advertising. Reference [46] describes the online advertising industry's proposal to mitigate the loss of privacy in OBA. The proposal is for the use of OBA disclosures: icons, accompanying taglines, and landing pages intended to inform users about OBA and provide opt-out options. The authors of Reference [46] conducted a 1,505-participant online study to investigate Internet users' perceptions of OBA disclosures. They found that the disclosures failed to clearly notify participants about OBA and inform them about their choices. The authors discuss the challenges in crafting disclosures and provide suggestions for improvement.

5.2 Client+Server Proposals

A fundamental problem with many client-centric solutions is that clients can be hacked, so the security strategies implemented therein will not be effective at all. To overcome this limitation, there have been proposals with client-server cooperation architecture.

Bluff ads [34] are baiting ads that are designed to be detected and clickable by bots or poorly trained click farm workers. When a bluff ad is clicked, the server side component considers the click as fraudulent. To address the performance issues of HTTPS in use cases such as Content Delivery Network (CDN) caching, HTTPi, a HTTP protocol with integrity, is proposed in Reference [75]. To achieve this, both web browsers and web servers need to be enhanced with new modules. This proposal is potentially useful for addressing hacking-related ad threats, but it requires a significant amount of work in the Internet of today. To address the MiTB attacks, Reference [77] presents a method and system to fingerprint transactions between client and server sides. A scheme to authenticate valid clicks, i.e., admitting only verifiably legitimate ones, is proposed in Reference [41]. The scheme is based on authenticating requests via cryptography attestations on clients. Notably, a web token can be used in the authentication process. The basic idea is to issue a unique identification for each session. A similar idea with a focus on increasing the transparency from the advertisers' perspective is proposed in Reference [14]. The idea is also implemented in JavaScript. This code collects relevant information associated with each impression and sends it to a central server. In particular, the following information will be obtained: the User-Agent receiving the impression, the URL where the impression is shown, and user interactions with the ad impression (mouse movements or clicks on the ad). The connection established with the server is used to obtain the IP address of the device receiving the ad impression as well as the time-stamp assigned to the impression. In the end, the exposure time of the ad impression will be estimated as the duration of the connection.

Research in Reference [16] contains an idea that user mouse movement data can be collected to feed into machine learning engines to help detect fraud. This is an effective method to differentiate bots from human beings. However, the user profile data containing mouse movements is much larger than the one without mouse movement information and the data are non-structural, which require more effort in applying machine learning techniques. Further, bots may still be able to fool the machine learning algorithms by injecting noise into the machine-driven mouse movements. A framework to detect mobile click bots is proposed in Reference [2]. It consists of client-side components responsible for collecting and filtering all the mobile-generated events and server-side components to process incoming data and to detect bots.

In contrast, a possible solution without modifying existing browsers is proposed in References [96] and [10]. Figure 8 illustrates a possible architecture to detect security-related issues in a browser environment with the

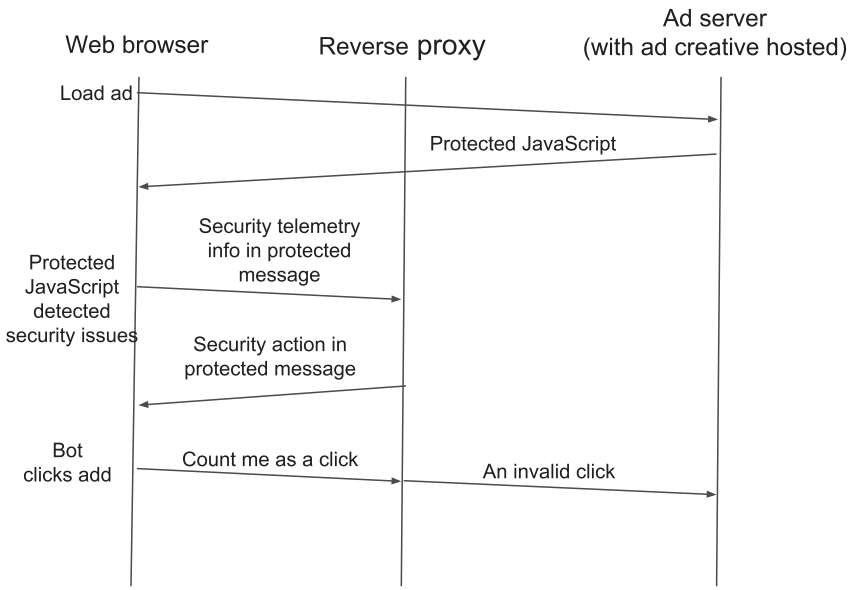

Fig. 8. A possible architecture to improve browser and mobile application security without radical changes to the client's and server's architecture [10].

help of protected JavaScript and a reverse proxy. The protected JavaScript for advertising has functionalities that include self-protection, anti-tampering, bot-detection, and others. With the help of protected JavaScript, the communication between web browsers and the reverse proxy is well protected from MiTB attacks.

5.3 Network-centric Proposals

By analyzing the traffic passing through the network edge, network monitoring devices such as intrusion detection systems (IDSs) can identify malicious traffic such as fraud and malvertising. In Section 4.1, we introduced the off-line approaches utilizing data analytics. There have been proposals to address malvertising using IDSs in a real-time fashion. In malvertising, attackers attempt to hide their web-based exploit kit services from being blacklisted. Hence, more advanced detection of malvertising traffic is needed. To infect a client browser, a web-based exploit kit must lead a web browser to visit its landing page, download an exploit file, and download a malicious payload, necessitating multiple requests to several malicious servers. In Reference [84], an approach is proposed to detect malvertising traffic based on the structure of these web requests, which are in a tree-like form, and uses the structural information for classification. First, an index of malicious tree samples is built using information retrieval techniques. Then, subtree similarity searching is applied to detect HTTP traffic-related to malicious exploit kits on enterprise networks.

Note that these proposals can potentially be scaled up and applied to large operations such as ISPs to detect more advertising-related issues including ad fraud. Technically, the first challenge that needs to be addressed is how to scale up these techniques to accommodate much larger volumes of traffic. One of the economic challenges is how to motivate ISPs to deploy these solutions to address advertising issues, which is briefly surveyed in Section 4.3.

5.4 Ad Blocking and Anti-adblocking

In this subsection, we discuss adblocking-related solutions. Ad blocking itself has the following security and performance benefits: First, users can block most ads listed on the blocklists of adblockers, freeing browsers

from having to run very time-consuming fetching and rendering of ads. Second, some malvertising can also be blocked by adblockers, preventing some browser vulnerabilities from being exploited. Third, to some extent, the users using adblockers can protect their privacy by blocking request to track the user's Internet surfing behavior. Therefore, adblocking is a security-enhancing measure, especially from the user's perspective.

However, adblocking can also be considered as a threat for publishers that rely on advertising to run their services. To maintain their income, publishers can set up a rule so their content or services are only available for those who are willing to see their ads. Such economic phenomenon is studied in Reference [67] using game theoretic modelling, describing how the parties are incentivized. Fundamentally, an adblocker runs as a web browser extension, having higher privilege than an anti-adblocking script, which usually runs in the JavaScript engine sandbox. This means that at the end of the competition between adblocking and anti-adblocking, adblockers have more technical advantages. Admittedly, the increasing use of adblocking is one of the symptoms of a problematic online advertising ecosystem, especially from the user's perspective. However, whether or not the cost incurred by adblocking is justified is unknown. Nevertheless, from a technical perspective, the schemes used in adblocking and anti-adblocking are valuable for increasing the security of online advertising.

A number of publishers deployed mechanisms for detecting and/or counter-blocking adblockers [38, 58]. The basic adblocking strategy is filtering. A list is created based on crowd-sourced feedback, based on which adblocker software will block traffic. To counter anti-adblocker, there are proposals to enhance the current adblocking technologies. In Reference [61], static program analysis is applied to JavaScript source code to identify JavaScript that loads and displays ads. Anti-adblockers scripts are detected on 30.5% of the Alexa top 10K websites [99]. JavaScript rewriting and API hooking is proposed in Reference [99] to counter anti-adblocker scripts. To detect anti-adblockers [40], information from both HTTP, HTML, and JavaScript are fed into a machine learning system to create a filtering list. The authors of Reference [40] propose a scheme to block ads served by applications in mobile devices. The network interface of the device is intercepted and with the information of the packets captured, a machine learning classifier is trained to block ads. The idea of perceptual blocking is proposed in Reference [82].

5.5 Private-by-Design Proposal for Online Behavioral Targeting

Designs do exist for online advertising systems that allow for online behavioral targeting without revealing the user's online behavior, and thus protecting his privacy. For instance, a framework called Adnostic is proposed in Reference [86], which is a browser-assisted approach. The scheme consists of two modules building user profile with privacy and rendering ads into publisher pages when the participated publishers' websites are visited. However, according to Reference [69], none of these designs have been implemented and deployed in the real world. The authors of Reference [69] attempt to remedy this situation by building and evaluating a fully functional prototype of a practical privacy-preserving ad system at a reasonably large scale and deploying it. The authors claim to have more than 13K opted-in users. Their system had been in operation for over two months at the time of writing their paper, serving an average of 4.8K active users daily. These authors found that their system obtained click-through rates that are comparable with Google display ads. Their paper is an account of their experience and lessons learned with this system, the first "privacy-by-design" behavioral advertising and analytics system.

6 SUMMING UP MITIGATION STRATEGIES AND COUNTERMEASURES

In Table 4, we list the countermeasures we have studied in Sections 4 and 5. The table is organized roughly in the order of when the proposals are published. However, there is no single solution that is capable to meet the security and privacy requirements of stakeholders, including advertisers, users, publishers, and brokerage networks. In terms of technologies that can be leveraged, there are the following streams:

- Data analytics: the large amount of data generated from advertising production environments can be useful for detecting fraud, detecting bots, preventing malvertising.

Table 4. Summary of the Surveyed Mitigation Strategies/Proposals against Threats to Online Advertising

Ref.	Threat	Mitigation strategy	Key points and remarks
[41]	Ad Fraud	Server side	Authenticating clicks. May require radical change to the ecosystem.
[68]	Ad injection	Browser-based	JavaScript to detect tampering in a web app. Hacker can modify the JavaScript to bypass the detection.
[89]	Fraud	Incentive analysis	Game theoretic analysis with focus on ISPs and ad networks.
[34]	Ad Fraud	Browser + Server	Baiting ads to detect click fraud. Hackers can detect the baiting ads based on some rules they learned from the baiting ads.
[88]	Ad injection	Ad-server based	Compute hash of web pages. Not feasible for modern dynamic web applications.
[28, 55]	Ad fraud	Game-theoretic analysis	The model consists of advertisers, ad networks, and publishers. Allowing fraud is suboptimal for ad networks. However, whether fighting fraud is the ad networks' first priority is an open question.
[27]	Malvertising	Browser-based sandboxing	A shadow JavaScript engine is used to sandbox untrusted ads. Modification of browser is required.
[77]	MiTB attacks	Client+Server	Verifying transactions between client and server by fingerprinting the transactions.
[33]	Ad Fraud and privacy	Mobile-ad SDK analysis	Identify SDKs with security concerns by analyzing the applications. It is first line of defense. Malicious SDK can still bypass the tests.
[73]	Ad fraud and privacy	Mobile-ad SDK framework	Ad SDK runs as a separate process. Ad SDK process can still acquire malicious privileges.
[64]	Ad fraud and privacy	Mobile-ad SDK framework	The ad SDK will not have the same privileges as the application at run-time. Radical changes to the smartphone operating system are required.
[48]	Malvertising	Crawler	MadTracer automatically generates detection rules and utilizes them to inspect advertisement delivery processes and detect malvertising activities. Heuristics of the rule generation mechanism needs to be renewable.
[25]	Ad fraud	Baiting ads + Data analytics	Baiting ads to generate some data to which Bayesian analysis is applied. Experiment for advertisers' benefits only.
[75]	Tampering	Client+Server	HTTP protocol with integrity. Modifications of browser and server are required.
[46]	Loss of privacy from OBA	Use of OBA disclosures (icons)	Icons accompany taglines and landing pages to inform the user about OBA and provide opt-out options. Disclosures failed to achieve their purpose—authors provide suggestions for improvement.

(Continued)

Table 4. Continued

Ref.	Threat	Mitigation strategy	Key points and remarks
[26]	Ad Fraud and ad injection	Data analytics	The model is based on the assumption that hackers need a higher return of investment to profit. The ground-truth problem is not discussed.
[59]	Ad fraud	Data analytics	Data-mining competition. Hybrid methods won. Works done before the deep-learning era.
[93]	Ad Fraud	Browser-based	JavaScript to detect bots. Hacker can bypass the tests by modifying the JavaScript code in the browser environment.
[16]	Ad Fraud	Browser + Server	User behaviors are collected to detect fraud using machine learning. No results were disclosed.
[23]	Ad fraud	Detection framework	Run applications in emulator and observe their behaviors. If rules are generated after the analysis, hackers can still bypass some of them.
[49]	Ad fraud	Detection framework for Windows phone	Detected frauds related to visual element placement. Vendor-specific.
[63]	Click fraud	Data analytics	Bayesian approach. Lack of ground truth.
[86]	Loss of privacy from OBA	Browser + Server	The business logic of behavioral profiling and targeting is moved to browsers. Browsers are assumed to be trustworthy, which is usually not the case.
[69]	Loss of privacy from OBA	Ad system built using Design-for-Privacy principles	Builds first deployed online ad system using Design-for-Privacy principles that allow for OBA and protect privacy. System claimed to have click-through rates comparable to Google display ads.
[47]	Ad fraud	Mobile-ad SDK framework	Utilizing ARM's TrustZone hardware. Hardware-specific.
[70]	Malvertising	Crawler	System to mimic Internet users to collect malvertisements. Advanced malvertisements can evade the crawler.
[10]	Tampering	Client+Server	Protect web applications using JavaScript protection. Lightweight, without browser modification and minimal server-side change.
[96]	Ad fraud	Client+Server	Protect ads in the browser environment via client-server cooperation.
[84]	Malvertising	Intrusion Detection System	A network-centric approach based on subtree similarity searching for detecting HTTP traffic related to malicious exploit kits on enterprise networks.
[11]	Ad fraud	Data analytics	Classification with subsequent supervised learning. With better dataset, things can be better.
[17]	Ad fraud	Data analytics	Aftermath estimation of financial loss caused by some malware. Online fraud detection is not discussed.
[14]	Ad Fraud	Browser + Server	Similar to References [68] and [93] with a focus on transparency from advertisers' view. Hacker can modify JavaScript to bypass the tests.

(Continued)

Table 4. Continued

Ref.	Threat	Mitigation strategy	Key points and remarks
[17]	Ad fraud	Loss lower bound estimate	Financial lower bound estimation of advertising revenue loss in a range of four years caused by TDSS/TDL4 Botnet.
[38]	Ad blocking	Publisher	Industry guideline of options available for publishers in face of adblocking.
[8, 9]	Ad fraud	Data analytics	Applying Neural Networks to detect fraud. No insights regarding effectiveness.
[35], [36]	Malvertising	Theoretical analysis	Game theoretic modelling on the battles between the ad network (defender) and the malvertiser (attacker). The effectiveness of the model needs more verification with real-world experiments.
[66]	Malvertising	Machine learning	Application of Support Vector Machine (SVM) to detect malvertisements on the publisher side. How to handle malvertising redirection can be challenging to only focus on the publisher side.
[39]	Ad fraud	Operating system module, Mobile	Sniffing HTTP request and hardware events to detect click fraud. Too much overhead. Far away from being practical.
[2]	Ad fraud	Client + Server	Behavior-related events are collected and processed at the server-side components to detect mobile click bots. Malicious parties can bypass the detection by hacking the client-side components.

- Security enhancements: solutions enhancing the delivery process include adblocking, preventing ads from being tampered with, and limiting privileges of mobile applications.
- Theoretical analysis: game theory is widely used to analyze problems such as motivations between players and defending and attacking in malvertising.

7 CHALLENGES AND RESEARCH DIRECTIONS

Among the various forms of threats towards online advertising discussed in Section 3, the main root challenges in defending are the lack of transparency and the weak security of client-side software. To address these challenges, we propose the following directions for future research and development.

7.1 Application of Distributed Ledger Technologies

One of the well-known pitfalls of online advertising is that it is difficult to measure and quantify from the advertisers' point of view. For instance, advertisers/brands cannot have any access to the data regarding where and how the traffic is sourced. This creates the space for ad fraud and ad injection. In many cases, some participants are reluctant to share the details of their anti-fraud or anti-tampering approaches and other countermeasures or strategies with the public. Moreover, anti-fraud or anti-tampering solution providers keep their services and technologies secret. Actually, advertisers and brands are the source of the value chain and are therefore the most motivated participant to fix the broken ecosystem. This fact can stimulate some new designs on a brand-aware system in which advertisers are capable of measuring their ad campaigns under a more transparent process.

Distributed Ledger Technologies (DLTs) such as blockchain can be a game changer for these situations. DLT is an emerging and disruptive technology beginning with the crytocurrency of bitcoin, followed by smart contracts and even a fully autonomous justice application system [83].[1] Currently, the well-recognized features of DLT are **consensus, provenance, and immutability**. Consensus comes with the fact that all transactions need to be validated by the network consisting of all players. The validation algorithms vary from system to system, but the key is that only transactions satisfying the rules will be considered as valid. Provenance means knowing where the data are from, i.e., their source. The data may be from ad traffic or the targeting information from an ad campaign. Immutability means that the data recorded on the ledger cannot be modified. In principle, with these three features, DLTs solve the problem of managing trust between entities that do not trust each other. And this aligns very well with the lack of trust in the current online advertising ecosystem.

Based on the openness of the blockchain systems, blockchain systems can be classfied as public blockchain, private blockchain, or consortium blockchain. As the name suggests, anyone can access data on a public blockchain. Only allowed/permitted parties can access data on a private blockchain. Access to a consortium blockchain is something in between access to public and private blockchains [98]. Emerging companies that apply blockchains in online advertising are reviewed in Reference [6]. The smart contract–based second-generation blockchain is the main scope of this article. The article lists startup companies including Adchain (https://adchain.com), AdEx (https://www.adex.network/), NYIAX (https://www.nyiax.com/), Madhive (https://madhive.com/), and Papyrus (https://papyrus.global/). Most of these companies' solutions are built on top of the public blockchain and their focuses can be slightly different per use case. However, they face the same issue of technology readiness: limited throughput as the system scales.

Therefore, in our opinion, there are two promising directions to pursue. First, find suitable ad security and privacy use cases with the current blockchain technologies and their limitations. Second, balance the performance of the blockchain and its features to allow the system to scale to address most of the online advertising-related problems. Here is a list of potential use cases of applying DLTs to address advertising-related security problems:

- Brand-aware advertising system: The lack of transparency has caused many advertisers to reduce their investment in online advertising. With DLTs, in particular, private blockchains or consortium blockchains, online advertising participants can store data on blockchains and can control access to their own data. Therefore, it is possible that advertisers can retrieve data on their campaigns without breaking the boundaries between players provided there is permission from the data owners. Without DLTs, it is much more difficult and complex to achieve such level of transparency in such a complex ecosystem. This will make a brand-aware system possible.
- Mechanisms to incentivize consumers: From the very beginning of the industry until today, consumers share their data such as demographic information and consumer behavior "freely" with the ad brokerage network. As consumers become more aware of data privacy and data ownership, it may no longer be possible to use consumer data for free. This is particularly true with more regulations on data privacy. For instance, the European Union's General Data Protection Regulation can be the beginning of user-centric data regulation. However, stricter regulation can restrain progress in the advertising ecosystem. For instance, consumers can only have two choices to allow or not disallow data collection. If there were more possibilities, the situation could bring more healthy dynamics into the advertising ecosystem. DLTs can provide the possibility to motivate consumers to share their data with the brokerage network. With DLTs, first the user can participate in the data collection process and they can decide which party can consume their data. In exchange, the online advertising ecosystem can provide incentive to consumers so they are more willing to share their data. A distributed ledger can essentially make the producer-consumer

[1]Justice application is the term used in Reference [83] for the most advanced blockchain-based application through which justifiability of a dispute can be achieved autonomously.

relationship in the advertising ecosystem possible. With such more interactive approach, hopefully threats such as ad fraud and privacy theft can be at least partially addressed.

7.2 Application of Software Protection on Ad Systems

Software protection [56] provides application programs with the abilities to protect themselves from analysis, reverse-engineering, and tampering. This is powerful technology for protecting applications including the code running at the endpoints of online advertising. For instance, in browser environments, JavaScript is the main programming language to deliver ads. Hackers can analyze the code, since the program itself is just a script. Furthermore, they can tamper or instrument bots to conduct attacks by taking advantage of vulnerabilities in the code. Software protection can at least make it harder for hackers to analyze the ad delivery process.

One of the challenges in applying software protection technologies such as obfuscation [20, 31] is on how to control the overhead introduced by security and protection. For instance, real-time bidding usually takes less than 100 milliseconds. This means software protection technologies should have as little overhead as possible. However, since code obfuscation has been used by almost every malware author, de-obfuscation technology is important for understanding and preventing threats such as malvertising. For instance, a semantics-based approach for automatic de-obfuscation of JavaScript code is proposed in Reference [50]. Reference [94] gives another proposal to detect obfuscated malicious JavaScript code by using static analysis together with runtime inspection.

7.3 Protecting Revenue in Emerging Ad Systems

By far, most of the advertising delivery endpoints consist of browsers running either in mobile phones, desktop computers, or mobile applications. Accordingly, advertisements present themselves as banners or video snippets most of the time. With the introduction of new Internet applications such as the Internet of Things (IoT) and new human computer interaction technologies such as Virtual Reality or Augmented Reality, advertising can find new execution points. In fact, researchers have started investigating possible solutions for advertising in IoT networks [4] and patents such as in Reference [7] have been filed for virtual reality–related advertising.

The emerging new forms of online advertising will create more technical challenges and opportunities in security-and-privacy–related research. The problems discussed in this article will remain relevant to these new systems.

8 CONCLUSIONS

This article is the first one in public literature to provide a technical survey of attacks on online advertising and the corresponding countermeasures. The threats towards online advertising ecosystem are reviewed and examined, the causality behind the threats is treated mainly from a technical perspective. Security and privacy problems related to online advertising are introduced and discussed.

From the investigations we conducted, it is very unlikely that a single solution will fully address the problems. And due to the fact that players in the ecosystem may have conflict of interest in some scenarios, a single solution benefiting a single player will not be viable in real life. Moreover, fighting the threats can be compared to an unending arms race, due to the fact that the current online advertising ecosystem unfortunately relies on security through obscurity to defend against attacks such as the ones discussed in this work.

At a very high level, we expect viable solutions to address these problems will involve multiple players in the ecosystem. We also proposed several research directions that can serve as starting points for future efforts to combat ad attacks.

REFERENCES

[1] 2011. Stanford University - Introduction to Computational Advertising. Retrieved from http://web.stanford.edu/class/msande239/#lecture-handouts.

[2] Iroshan Aberathne and Chamila Walgampaya. 2018. Smart mobile bot detection through behavioral analysis. In *Advances in Data and Information Sciences*. Springer, 241–252.

[3] Adweek.com. 2017. Procter & Gamble Cut Up to $140 Million in Digital Ad Spending Because of Brand Safety Concerns. Retrieved from http://www.adweek.com/digital/procter-gamble-cut-140-million-in-digital-ad-spending-because-of-brand-safety-concerns/.

[4] Hidayet Aksu, Leonardo Babun, Mauro Conti, Gabriele Tolomei, and A. Selcuk Uluagac. 2018. Advertising in the IoT era: Vision and challenges. *IEEE Commun. Mag.* 56, 11 (2018).

[5] Sumayah A. Alrwais, Alexandre Gerber, Christopher W. Dunn, Oliver Spatscheck, Minaxi Gupta, and Eric Osterweil. 2012. Dissecting ghost clicks: Ad fraud via misdirected human clicks. In *Proceedings of the 28th Computer Security Applications Conference*. ACM, 21–30.

[6] Ashiq Anjum, Manu Sporny, and Alan Sill. 2017. Blockchain standards for compliance and trust. *IEEE Cloud Comput.* 4, 4 (2017), 84–90.

[7] Juha Arrasvuori, Jukka Antero Holm, and Antti Johannes Eronen. 2014. Personal augmented reality advertising. US Patent 8,644,842.

[8] Anup Badhe. 2017. Click fraud detection in mobile ads served in programmatic inventory. *Neural Netw. Mach. Learn.* 1, 1 (2017), 1–1.

[9] Anup Badhe. 2017. Using neural networks to detect supply side fraud in programmatic exchanges. *Neural Netw. Mach. Learn.* 1, 1 (2017), 1–1.

[10] Roozbehani Yaser Eftekhari, Mark Yep-Kui Chua, Benjamin Geoffrey Gidley, Catherine Chambers, and Yuan Xiang Gu. 2018. Securing webpages, webapps, and applications. Patent No. CA3008199A1.

[11] Daniel Berrar. 2016. Learning from automatically labeled data: Case study on click fraud prediction. *Knowl. Inf. Syst.* 46, 2 (2016), 477–490.

[12] Tommy Blizard and Nikola Livic. 2012. Click-fraud monetizing malware: A survey and case study. In *Proceedings of the 7th International Conference on Malicious and Unwanted Software (MALWARE'12)*. IEEE, 67–72.

[13] Chris Cain. 2014. Analyzing Man-in-the-Browser (MITB) Attacks. Retrieved from https://www.sans.org/reading-room/whitepapers/forensics/analyzing-man-in-the-browser-mitb-attacks-35687.

[14] Patricia Callejo, Ruben Cuevas, Angel Cuevas, and Mikko Kotila. 2016. Independent auditing of online display advertising campaigns. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*. ACM, 120–126.

[15] Gong Chen, Jacob H. Cox, A. Selcuk Uluagac, and John A. Copeland. 2016. In-depth survey of digital advertising technologies. *IEEE Commun. Surv. Tutor.* 18, 3 (2016), 2124–2148.

[16] J. Chen, D. Lin, A. Kaufman, and Y. Villa. 2014. Click stream analysis for fraud detection. Retrieved from https://www.google.com/patents/US8880441. US Patent 8,880,441.

[17] Yizheng Chen, Panagiotis Kintis, Manos Antonakakis, Yacin Nadji, David Dagon, Wenke Lee, and Michael Farrell. 2016. Financial lower bounds of online advertising abuse. In *Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 231–254.

[18] G. Cho, J. Cho, Y. Song, and H. Kim. 2015. An empirical study of click fraud in mobile advertising networks. In *Proceedings of the 10th International Conference on Availability, Reliability and Security*. 382–388. DOI: https://doi.org/10.1109/ARES.2015.62

[19] Taejoong Chung, David Choffnes, and Alan Mislove. 2016. Tunneling for transparency: A large-scale analysis of end-to-end violations in the internet. In *Proceedings of the Internet Measurement Conference*. ACM, 199–213.

[20] Christian Collberg. 2018. Code obfuscation: Why is this still a thing? In *Proceedings of the 8th ACM Conference on Data and Application Security and Privacy*. ACM, 173–174.

[21] Lucian Constantin. 2016. The AdGholas malvertising campaign infected thousands of computers per day. Retrieved from http://www.computerworld.com/article/3101823/security/the-adgholas-malvertising-campaign-infected-thousands-of-computers-per-day.html.

[22] Mauro Conti, Nicola Dragoni, and Viktor Lesyk. 2016. A survey of man in the middle attacks. *IEEE Commun. Surv. Tutor.* 18, 3 (2016), 2027–2051.

[23] Jonathan Crussell, Ryan Stevens, and Hao Chen. 2014. MAdFraud: Investigating ad fraud in Android applications. In *Proceedings of the 12th International Conference on Mobile Systems, Applications, and Services (MobiSys'14)*. ACM, 123–134. DOI: https://doi.org/10.1145/2594368.2594391

[24] Neil Daswani and Michael Stoppelman. 2007. The anatomy of Clickbot.A. In *Proceedings of the 1st Conference on First Workshop on Hot Topics in Understanding Botnets*. USENIX Association, 11–11. Retrieved from http://dl.acm.org/citation.cfm?id=1323128.1323139.

[25] Vacha Dave, Saikat Guha, and Yin Zhang. 2012. Measuring and fingerprinting click-spam in ad networks. *ACM SIGCOMM Comput. Commun. Rev.* 42, 4 (2012), 175–186.

[26] Vacha Dave, Saikat Guha, and Yin Zhang. 2013. Viceroi: Catching click-spam in search ad networks. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. ACM, 765–776.

[27] Xinshu Dong, Minh Tran, Zhenkai Liang, and Xuxian Jiang. 2011. AdSentry: Comprehensive and flexible confinement of JavaScript-based advertisements. In *Proceedings of the 27th Computer Security Applications Conference*. ACM, 297–306.

[28] Lemonia Dritsoula and John Musacchio. 2014. A game of clicks: Economic incentives to fight click fraud in ad networks. *ACM SIGMETRICS Perf. Eval. Rev.* 41, 4 (2014), 12–15.

[29] Steven Englehardt and Arvind Narayanan. 2016. Online tracking: A 1-million-site measurement and analysis. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1388–1401.

[30] Adrienne Porter Felt, Matthew Finifter, Erika Chin, Steve Hanna, and David Wagner. 2011. A survey of mobile malware in the wild. In *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*. 3–14.
[31] Pedro Fortuna, Nuno Pereira, and Ismail Butun. 2018. A framework for web application integrity. In *Proceedings of the International Conference on Information Systems Security and Privacy (ICISSP'18)*. 487–493.
[32] Avi Goldfarb and Catherine E. Tucker. 2011. Privacy regulation and online advertising. *Manag. Sci.* 57, 1 (2011), 57–71.
[33] Michael C. Grace, Wu Zhou, Xuxian Jiang, and Ahmad-Reza Sadeghi. 2012. Unsafe exposure analysis of mobile in-app advertisements. In *Proceedings of the 5th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WISEC'12)*. ACM, 101–112. DOI: https://doi.org/10.1145/2185448.2185464
[34] Hamed Haddadi. 2010. Fighting online click-fraud using bluff ads. *ACM SIGCOMM Comput. Commun. Rev.* 40, 2 (2010), 21–25.
[35] Chin-Tser Huang, Muhammad N. Sakib, Charles Kamhoua, Kevin Kwiat, and Laurent Njilla. 2017. A game theoretic approach for inspecting web-based malvertising. In *Proceedings of the IEEE International Conference on Communications (ICC'17)*. IEEE, 1–6.
[36] Chin-Tser Huang, Muhammad Nazmus Sakib, Charles Kamhoua, Kevin A. Kwiat, and Laurent Njilla. 2018. A Bayesian game theoretic approach for inspecting web-based malvertising. *IEEE Trans. Depend. Sec. Comput.* (2018).
[37] IAB. 2014. IAB Anti-Fraud Principles and Proposed Taxonomy. Retrieved from https://www.iab.com/wp-content/uploads/2015/05/IAB_Anti_Fraud_Principles_and_Taxonomy.pdf.
[38] IAB. 2016. Retrieved from https://www.iab.com/wp-content/uploads/2016/03/IABTechLab_Publisher_AdBlocking_Primer.pdf.
[39] Md Shahrear Iqbal, Mohammad Zulkernine, Fehmi Jaafar, and Yuan Gu. 2018. Protecting internet users from becoming victimized attackers of click-fraud. *J. Softw. Evol. Proc.* 30, 3 (2018), e1871.
[40] Umar Iqbal, Zubair Shafiq, Peter Snyder, Shitong Zhu, Zhiyun Qian, and Benjamin Livshits. 2018. AdGraph: A machine learning approach to automatic and effective adblocking. *Arxiv Preprint Arxiv:1805.09155* (2018).
[41] Ari Juels, Sid Stamm, and Markus Jakobsson. 2007. Combating click fraud via premium clicks. In *Proceedings of the USENIX Security Symposium*. 17–26.
[42] Daniel Kaminsky and Michael J. J. Tiffany. 2016. System and method for detecting classes of automated browser agents. Patent No. US9313213 B2 International Classification G06N7/00, H04L29/06; Cooperative Classification H04L63/1416, G06F2221/2133, G06F21/31, H04L63/12, G06N7/00.
[43] Nikolaos Karapanos and Srdjan Capkun. 2014. On the effective prevention of TLS man-in-the-middle attacks in web applications. In *Proceedings of the USENIX Security Symposium*, Vol. 23. 671–686.
[44] Sheharbano Khattak, Naurin Rasheed Ramay, Kamran Riaz Khan, Affan A. Syed, and Syed Ali Khayam. 2014. A taxonomy of botnet behavior, detection, and defense. *IEEE Commun. Surv. Tutor.* 16, 2 (2014), 898–924.
[45] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436.
[46] Pedro Giovanni Leon, Justin Cranshaw, Lorrie Faith Cranor, Jim Graves, Hastak Manoj, Blase Ur, and Guzi Xu. 2012. What do online behavioral advertising privacy disclosures communicate to users? In *Proceedings of the ACM Workshop on Privacy in the Electronic Society*. 19–30.
[47] Wenhao Li, Haibo Li, Haibo Chen, and Yubin Xia. 2015. AdAttester: Secure online mobile advertisement attestation using trustzone. In *Proceedings of the 13th International Conference on Mobile Systems, Applications, and Services*. ACM, 75–88.
[48] Zhou Li, Kehuan Zhang, Yinglian Xie, Fang Yu, and XiaoFeng Wang. 2012. Knowing your enemy: Understanding and detecting malicious web advertising. In *Proceedings of the ACM Conference on Computer and Communications Security*. ACM, 674–686.
[49] Bin Liu, Suman Nath, Ramesh Govindan, and Jie Liu. 2014. DECAF: Detecting and characterizing ad fraud in mobile apps. In *Proceedings of the USENIX Conference on Networked Systems Design and Implementation (NSDI'14)*. USENIX Association, 57–70. Retrieved from http://dl.acm.org/citation.cfm?id=2616448.2616455.
[50] Gen Lu and Saumya Debray. 2012. Automatic simplification of obfuscated JavaScript code: A semantics-based approach. In *Proceedings of the IEEE 6th International Conference on Software Security and Reliability*. IEEE, 31–40.
[51] Lumapartners.com. 2018. Display LUMAscape. Retrieved from https://lumapartners.com/content/lumascapes/display-ad-tech-lumascape/.
[52] Marketingdive.com. 2016. Facebook shuts down ad buying platform after discovering ad fraud. Retrieved from https://www.marketingdive.com/news/facebook-shuts-down-ad-buying-platform-after-discovering-ad-fraud/415236/.
[53] Wei Meng, Ren Ding, Simon P. Chung, Steven Han, and Wenke Lee. 2016. The price of free: Privacy leakage in personalized mobile in-apps ads. In *Proceedings of the Network and Distributed System Security Symposium*.
[54] Wei Meng, Xinyu Xing, Anmol Sheth, Udi Weinsberg, and Wenke Lee. 2014. Your online interests: Pwned! A pollution attack against targeted advertising. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. ACM, 129–140.
[55] Bob Mungamuru, Stephen Weis, and Hector Garcia-Molina. 2008. *Should Ad Networks Bother Fighting Click Fraud? (Yes, they Should.)*. Technical Report. Stanford University.
[56] Jasvir Nagra and Christian Collberg. 2009. *Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection*. Pearson Education.
[57] Gabi Nakibly, Jaime Schcolnik, and Yossi Rubin. 2016. Website-targeted false content injection by network operators.. In *Proceedings of the USENIX Security Symposium*. 227–244.

[58] Rishab Nithyanand, Sheharbano Khattak, Mobin Javed, Narseo Vallina-Rodriguez, Marjan Falahrastegar, Julia E. Powles, E. D. Cristofaro, Hamed Haddadi, and Steven J. Murdoch. 2016. Adblocking and counter blocking: A slice of the arms race. In *CoRR*, Vol. 16. USENIX. *arXiv:1605.05077*

[59] Richard Oentaryo, Ee-Peng Lim, Michael Finegold, David Lo, Feida Zhu, Clifton Phua, Eng-Yeow Cheu, Ghim-Eng Yap, Kelvin Sim, Minh Nhut Nguyen, et al. 2014. Detecting click fraud in online advertising: A data mining approach. *J. Mach. Learn. Res.* 15, 1 (2014), 99–140.

[60] Mark O'Neill, Scott Ruoti, Kent Seamons, and Daniel Zappala. 2016. TLS proxies: Friend or foe? In *Proceedings of the Internet Measurement Conference*. ACM, 551–557.

[61] Caitlin R. Orr, Arun Chauhan, Minaxi Gupta, Christopher J. Frisz, and Christopher W. Dunn. 2012. An approach for identifying JavaScript-loaded advertisements through static program analysis. In *Proceedings of the ACM Workshop on Privacy in the Electronic Society*. ACM, 1–12.

[62] OWASP. 2016. Man-in-the-browser attack. Retrieved from https://www.owasp.org/index.php/Man-in-the-browser_attack.

[63] Paul Pearce, Vacha Dave, Chris Grier, Kirill Levchenko, Saikat Guha, Damon McCoy, Vern Paxson, Stefan Savage, and Geoffrey M. Voelker. 2014. Characterizing large-scale click fraud in zeroaccess. In *Proceedings of the ACM Conference on Computer and Communications Security*.

[64] Paul Pearce, Adrienne Porter Felt, Gabriel Nunez, and David Wagner. 2012. AdDroid: Privilege separation for applications and advertisers in Android. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security (ASIACCS'12)*. ACM, 71–72. DOI: https://doi.org/10.1145/2414456.2414498.

[65] Raffaello Perrotta and Feng Hao. 2018. Botnet in the browser: Understanding threats caused by malicious browser extensions. *IEEE Sec. Priv.* 16, 4 (2018), 66–81.

[66] Prabaharan Poornachandran, N. Balagopal, Soumajit Pal, Aravind Ashok, Prem Sankar, and Manu R. Krishnan. 2017. Demalvertising: A kernel approach for detecting malwares in advertising networks. In *Proceedings of the 1st International Conference on Intelligent Computing and Communication*. Springer, 215–224.

[67] Abhishek Ray, Hossein Ghasemkhani, and Karthik N. Kannan. 2017. Ad-blockers, advertisers, and internet: The economic implications of ad-blocker platforms. In *Proceedings of the International Conference on Information Systems*.

[68] Charles Reis, Steven D. Gribble, Tadayoshi Kohno, and Nicholas C. Weaver. 2008. Detecting in-flight page changes with web tripwires. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation*, Vol. 8. 31–44. Retrieved from https://www.usenix.org/legacy/events/nsdi08/tech/full_papers/reis/reis.pdf.

[69] Alexey Reznichenko and Paul Francis. 2014. Private-by-design advertising meets the real world. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. 116–128.

[70] Muhammad N. Sakib and Chin-Tser Huang. 2015. Automated collection and analysis of malware disseminated via online advertising. In *Proceedings of the IEEE Trustcom/BigDataSE/ISPA*, Vol. 1. IEEE, 1411–1416.

[71] Jerome H. Saltzer, David P. Reed, and David D. Clark. 1984. End-to-end arguments in system design. *ACM Trans. Comput. Syst.* 2, 4 (1984), 277–288.

[72] Jerome Segura. 2017. Canada and the U.K. hit by Ramnit Trojan in new malvertising campaign. Retrieved from https://blog.malwarebytes.com/threat-analysis/exploits-threat-analysis/20 17/03/canada-u-k-hit-ramnit-trojan-new-malvertising-campaign/.

[73] Shashi Shekhar, Michael Dietz, and Dan S. Wallach. 2012. AdSplit: Separating smartphone advertising from applications. In *Proceedings of the USENIX Security Symposium*. 553–567. Retrieved from https://www.usenix.org/system/files/conference/usenixsecurity12/sec12-final101.pdf.

[74] Sérgio S. C. Silva, Rodrigo M. P. Silva, Raquel C. G. Pinto, and Ronaldo M. Salles. 2013. Botnets: A survey. *Comput. Netw.* 57, 2 (2013), 378–403. DOI: https://doi.org/10.1016/j.comnet.2012.07.021

[75] Kapil Singh, Helen J. Wang, Alexander Moshchuk, Collin Jackson, and Wenke Lee. 2012. Practical end-to-end web content integrity. In *Proceedings of the 21st International Conference on World Wide Web*. ACM, 659–668.

[76] Peter Snyder and Chris Kanich. 2016. Characterizing fraud and its ramifications in affiliate marketing networks. *J. Cybersec.* 2, 1 (2016), 71–81. DOI: https://doi.org/10.1093/cybsec/tyw006

[77] William E. Sobel and Sourabh Satish. 2012. Methods and systems for detecting man-in-the-browser attacks. US Patent 8,225,401.

[78] Aditya K. Sood, Sherali Zeadally, and Rohit Bansal. 2017. Cybercrime at a scale: A practical study of deployments of HTTP-based botnet command and control panels. *IEEE Commun. Mag.* 55, 7 (2017), 22–28.

[79] Spider.io. 2013. How to Defraud Display Advertisers with Zeus. Retrieved from http://www.spider.io/blog/2013/11/how-to-defraud-display-advertisers-with-zeus/.

[80] Mark Stamp. 2011. *Information Security: Principles and Practice*. John Wiley & Sons.

[81] Brett Stone-Gross, Ryan Stevens, Apostolis Zarras, Richard Kemmerer, Chris Kruegel, and Giovanni Vigna. 2011. Understanding fraudulent activities in online ad exchanges. In *Proceedings of the ACM SIGCOMM Conference on Internet Measurement*. ACM, 279–294.

[82] Grant Storey, Dillon Reisman, Jonathan Mayer, and Arvind Narayanan. 2017. The future of ad blocking: An analytical framework and new techniques. *Arxiv Preprint Arxiv:1705.08568* (2017).

[83] Melanie Swan. 2015. *Blockchain: Blueprint for a New Economy*. O'Reilly Media.
[84] Teryl Taylor, Xin Hu, Ting Wang, Jiyong Jang, Marc Ph Stoecklin, Fabian Monrose, and Reiner Sailer. 2016. Detecting malicious exploit kits using tree-based similarity searches. In *Proceedings of the 6th ACM Conference on Data and Application Security and Privacy*. ACM, 255–266.
[85] K. Thomas, E. Bursztein, C. Grier, G. Ho, N. Jagpal, A. Kapravelos, D. Mccoy, A. Nappa, V. Paxson, P. Pearce, N. Provos, and M. A. Rajab. 2015. Ad injection at scale: Assessing deceptive advertisement modifications. In *Proceedings of the IEEE Symposium on Security and Privacy*. 151–167. DOI: https://doi.org/10.1109/SP.2015.17
[86] Vincent Toubiana, Arvind Narayanan, Dan Boneh, Helen Nissenbaum, and Solon Barocas. 2010. Adnostic: Privacy preserving targeted advertising. In *Proceedings of the Network and Distributed System Symposium*.
[87] Giridhari Venkatadri, Athanasios Andreou, Yabing Liu, Alan Mislove, Krishna P. Gummadi, Patrick Loiseau, and Oana Goga. 2018. Privacy risks with Facebook's PII-based targeting: Auditing a data broker's advertising interface. In *Proceedings of the IEEE Symposium on Security and Privacy (SP'18)*. IEEE, 89–107.
[88] Neenav Vratonjic, Julien Freudiger, and Jean-Pierre Hubaux. 2010. Integrity of the web content: The case of online advertising. In *Proceedings of the International Conference on Collaborative Methods for Security and Privacy*. USENIX Association, 2–2.
[89] Nevena Vratonjic, Mohammad Hossein Manshaei, Maxim Raya, and Jean-Pierre Hubaux. 2010. ISPs and ad networks against botnet ad fraud. In *Proceedings of the International Conference on Decision and Game Theory for Security*. Springer, 149–167.
[90] Jun Wang, Weinan Zhang, and Shuai Yuan. 2016. Display advertising with real-time bidding (RTB) and behavioural targeting. *Arxiv Preprint Arxiv:1610.03013* (2016).
[91] WhiteOps. 2016. The METHBOT operation: WhiteOps has exposed the largest and most profitable ad fraud operation to strike digital advertising to date. Retrieved from https://www.whiteops.com/methbot.
[92] Wikipedia. 2014. Interactive Advertising Bureau. Retrieved from https://en.wikipedia.org/wiki/Interactive_Advertising_Bureau.
[93] Haitao Xu, Daiping Liu, Aaron Koehl, Haining Wang, and Angelos Stavrou. 2014. Click fraud detection on the advertiser side. In *Proceedings of the European Symposium on Research in Computer Security*. Springer, 419–438.
[94] Wei Xu, Fangfang Zhang, and Sencun Zhu. 2013. JStill: Mostly static detection of obfuscated malicious JavaScript code. In *Proceedings of the 3rd ACM Conference on Data and Application Security and Privacy*. ACM, 117–128.
[95] George O. M. Yee. 2017. Visualization and prioritization of privacy risks in software systems. *Int. J. Adv. Secur.* 10, 1&2 (2017), 14–25.
[96] Benjamin Geoffrey, Gidley Yuan, Xiang Gu, Andrew Augustine Wajs, and Wim Mooij. 2015. Online advertisements. International Classification G06Q30/0241 Advertisement, Patent No. PCT/EP2016/057109.
[97] Apostolis Zarras, Alexandros Kapravelos, Gianluca Stringhini, Thorsten Holz, Christopher Kruegel, and Giovanni Vigna. 2014. The dark alleys of Madison Avenue: Understanding malicious advertisements. In *Proceedings of the Conference on Internet Measurement*. 373–380.
[98] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. 2017. An overview of blockchain technology: Architecture, consensus, and future trends. In *Proceedings of the IEEE International Congress on Big Data (BigData Congress'17)*. IEEE, 557–564.
[99] Shitong Zhu, Xunchao Hu, Zhiyun Qian, Zubair Shafiq, and Heng Yin. 2018. Measuring and disrupting anti-adblockers using differential execution analysis. In *Proceedings of the Network and Distributed System Security Symposium (NDSS'18)*. Internet Society.
[100] Xingquan Zhu, Haicheng Tao, Zhiang Wu, Jie Cao, Kristopher Kalish, and Jeremy Kayne. 2017. *Fraud Prevention in Online Digital Advertising*. Springer.

Received April 2019; revised October 2019; accepted November 2019