The Dark Alleys of Madison Avenue: Understanding Malicious Advertisements

Apostolis Zarras
Ruhr-University Bochum
apostolis.zarras@rub.de

Alexandros Kapravelos
UC Santa Barbara
kapravel@cs.ucsb.edu

Gianluca Stringhini
University College London
g.stringhini@ucl.ac.uk

Thorsten Holz
Ruhr-University Bochum
thorsten.holz@rub.de

Christopher Kruegel
UC Santa Barbara
chris@cs.ucsb.edu

Giovanni Vigna
UC Santa Barbara
vigna@cs.ucsb.edu

ABSTRACT

Online advertising drives the economy of the World Wide Web. Modern websites of any size and popularity include advertisements to monetize visits from their users. To this end, they assign an area of their web page to an advertising company (so called *ad exchange*) that will use it to display promotional content. By doing this, the website owner implicitly trusts that the advertising company will offer legitimate content and it will not put the site's visitors at risk of falling victims of malware campaigns and other scams.

In this paper, we perform the first large-scale study of the safety of the advertisements that are encountered by the users on the Web. In particular, we analyze to what extent users are exposed to malicious content through advertisements, and investigate what are the sources of this malicious content. Additionally, we show that some ad exchanges are more prone to serving malicious advertisements than others, probably due to their deficient filtering mechanisms. The observations that we make in this paper shed light on a little studied, yet important, aspect of advertisement networks, and can help both advertisement networks and website owners in securing their web pages and in keeping their visitors safe.

Categories and Subject Descriptors

K.4.4 [**Computers and Society**]: Electronic Commerce; C.2.0 [**Computer-Communication Networks**]: General; H.3.5 [**Information Storage and Retrieval**]: Online Information Services

General Terms

Measurement; Security

Keywords

Online Advertising; Malware; Malvertising

IMC'14, November 5–7, 2014, Vancouver, BC, Canada.

http://dx.doi.org/10.1145/2663716.2663719.

1. INTRODUCTION

The online advertising industry is constantly growing. A recent report showed that this industry generated a revenue of 42.8 billion dollars in 2013, which is 17% higher than what had been reported in the previous year [14]. In the World Wide Web, where most online services are free of charge, advertisements constitute the main revenue for website administrators, and it is very common to see promotional content alongside the actual information contained in such sites.

Given the profitability of online advertising, many big players have entered the arena. Such players, known as *ad exchanges*, put in contact the *advertisers*, who want their content to be displayed, with the *publishers*, who want to show promotional content on their web pages, and make sure that the most suitable advertisement will be displayed to the visitors of that site at all times. Recent research showed that Google's *Doubleclick* ad exchange service is the largest on the Internet, being present on 80% of the websites that provide advertisements [12]. A publisher looking to generate some revenue can easily subscribe with one of these companies, dedicate a part of her web pages to advertisements, and start serving promotional content to that page's visitors. Publishers are paid either by *impression*, meaning that they get a sum of money every time a visitor watches an advertisement on their site, or by *click*, meaning that they get paid every time a user shows interest in the advertisement and clicks on it, visiting the advertiser's website.

Because of their pervasiveness, online advertisements are not only used by legitimate parties, but also by miscreants. A common scam linked to online advertisements is *click fraud* [29]. In this scheme, cyber-criminals first set up web pages and become publishers. Then, they instruct a *botnet*, which is a network of compromised computers acting under the cyber-criminal's control, to visit the web page and click on the advertisements displayed on it [3]. By doing this, the cyber-criminal will get paid by the ad exchange and make a revenue. Click fraud is a big concern for ad exchanges, and a wealth of research has been conducted to detect and block suspicious clicks on online advertisements [6, 7, 25].

Besides click frauds, online advertisements provide a convenient platform for infecting web users with malware. Attackers can set up malicious advertisements that attempt to automatically exploit the user's browser and install malware with a *drive-by download* attack [23], or they can display an advertisement that lures the victim into installing malware

through social engineering, making the advertisement look appealing to the user [24].

Leveraging advertisements to spread malware has many advantages for attackers. Since advertisements are displayed on very popular websites, miscreants have a chance of infecting a larger number of victims in a short amount of time. Without the use of advertisements, the only way that an attacker would reach a similar goal is by compromising the home page of a popular site, which is a challenging task due to its security mechanisms. In addition, publishers usually trust the advertisement network (*ad network*) that they entertain business with, and are unaware that such networks could actually end up serving malicious content.

Previous research showed that malicious advertisements are fairly common in the wild [18, 20, 22, 26]. Similarly, recent news showed the feasibility of having malicious advertisements going undetected by major ad exchanges, and being served to users [11]. However, no comprehensive research has been conducted on understanding the ecosystem surrounding malicious advertisements. The prevalence of malicious advertisements on the Web, the number of ad networks that serve these malvertisements, and the quality of the defense systems deployed by ad exchanges are still open questions.

In this paper, we study the ecosystem of malicious advertisements. We crawled more than 600,000 real-world advertisements, and checked them against multiple detection systems to assess their maliciousness. We show that certain ad exchanges are serving more malicious content than others, probably because they have insufficient detection systems. We also show that, because of the arbitration process, it is common for ad exchanges to serve a malicious advertisement provided by another ad exchange.

In summary, we make the following main contributions:

- We collect a corpus of more than 600,000 real-world advertisements from various web pages and describe the misbehaving advertisements that we discovered.
- We analyze different ad exchanges and show that some of them are more prone to serving malicious advertisements than others.
- We demonstrate that due to the arbitration process, every website that serves advertisements and that does not have an exclusive agreement with the advertiser is a potential publisher of malicious advertisements.
- We show that the vast majority of publishers tend to trust their advertisers not to serve malicious advertisements and thus they do not apply any additional filters to protect their users.

2. MALICIOUS ADVERTISING

Malicious advertising, known as *malvertising*, is the cybercriminals' practice of injecting malicious or malware-laden advertisements into legitimate online advertising networks and syndicated content. It can occur through deceptive advertisers or agencies running advertisements or compromises to the ad-supply chain including ad networks, ad exchanges, and ad servers. Different types of malicious advertisements exhibit different behaviors and in the following sections we briefly describe them.

2.1 Drive-by Downloads

A drive-by download advertisement is an advertisement that hosts one or more exploits that target specific vulnerabilities in web browsers. More precisely, attackers target vulnerabilities in web browsers or in browser plugins, such as Flash or Java, that enable users to experience rich media content within the browser environment. In some cases, the browser vendor pre-installs these plugins. The user may not even use the vulnerable plugin or be aware that it is installed. Users with vulnerable computers can be transparently infected with malware by visiting a website that serves a drive-by download, even without interacting with the malicious part of the page.

2.2 Deceptive Downloads

Deceptive downloads try to lure their victims to download and install a specific software component that is malicious. The main difference from drive-by downloads is that attackers do not try to find a vulnerability in the victim's browser or browser plugins to download and install a piece of malware, but instead they try to trick the users into performing that procedure voluntarily. This happens by having the user believe that there is some desirable content on the visited web page. More specifically, the victims get informed that, in order to gain access to specific content of the page, they need to install a particular software component or to upgrade their supposedly outdated plugins. Of course, the updating/installing procedure installs malware on the user's hosts instead of the advertised software.

2.3 Link Hijacking

Link hijacking allows an advertisement to automatically redirect users to websites that they have not decided to visit. The advertisements are included in iframes, and the advertising scripts cannot access the Document Object Model (DOM) of the publisher's web page due to the Same-Origin Policy (SOP) restrictions [2]. However, a malicious script contained in an advertisement can redirect the entire page to a preselected destination by setting Browser Object Model's (BOM) `top.location` variable [22]. This way, the victim is redirected to an arbitrary location and not to the one she has initially selected.

3. METHODOLOGY

In this section, we present the methodology we used to generate and evaluate a large corpus of advertisements. Our process includes two steps. First, we extract the advertisements from a variety of websites. Second, we use a number of oracles to classify the advertisements as malicious or legitimate. We describe both steps in detail.

3.1 Data Collection

In the first phase, we performed a large web crawl to create a corpus of advertisements. For this purpose, we used two different data feeds. First, we leveraged a data feed obtained from an antivirus company (already used in our previous work [26]). This feed contains web pages that in the past appeared to have a malicious behavior, and was generated by users who installed a browser security product to voluntarily share their data with the company for research purposes. For the second feed, we used Alexa's one million top-ranked websites list. To have a certain degree of diversity in our data, we selected the top and the bottom 10,000 websites,

the top and the bottom 1,000 websites from selected top-level domains, and also 20,000 randomly selected websites from Alexa's ranked websites.

Due to the fact that the content of the advertisements is dynamically generated, we periodically crawled each web page in an attempt to obtain different advertisements. More specifically, our crawler visited each website once per day, and refreshed a web page five times. Our crawler was based on `Selenium`, which is a software-testing framework for web applications that has the ability to programmatically control a real web browser (Mozilla Firefox in our experiments). This approach allowed us to retrieve the whole content of a rendered advertisement, which would not be possible if we used a simple command-line tool like `wget`. Additionally, we captured all the HTTP traffic during crawling for further investigation.

In most of the cases, the advertisements were included in an *iframe*. An iframe is an HTML document embedded inside another HTML document. This allows the iframe to be rendered in a consistent way even if it is included by different websites. We leveraged this fact and we created HTML documents based on the contents of the iframes. These iframes included both statically- and dynamically-generated HTML elements. It is important to note that not all the iframes included in a web page contain advertisements. Thus, to distinguish the advertisement-related iframes, we utilized *EasyList*[1]. EasyList include domains and URL patterns for ad-related hosts, and is used by the browser plugin `Adblock Plus` [1] to block advertisements.

After a period of three months, we have created a corpus of 673,596 unique advertisements. We then analyzed this dataset searching for misbehaving advertisements.

3.2 Oracles

To classify if an advertisement exhibits malicious behavior, we utilized an oracle. The oracle constitutes an essential part of our study. It gets as an input the initial request for advertisements from a publisher's website, and by monitoring several behavioral features, it can effectively determine the maliciousness of the advertisement. The lifeblood of our oracle constitutes by three main components: `Wepawet` [4], malware and phishing blacklists, and `VirusTotal`. We describe the contribution of each component in the classification process in the next paragraphs.

3.2.1 Wepawet

As advertisements are included in pages with dynamic content that often changes over time, they are also dynamically-generated. The dynamic nature of advertisements is achieved with the use of JavaScript or Flash. Miscreants can unleash their nefarious activities (drive-by download attacks, phishing attempts, etc.) to victims through advertisements. Therefore, we need to analyze the advertisements' embedded code, which is often dynamically loaded, to detect if there exist any kind of malicious behavior.

To do so, we utilized `Wepawet` [4], a honeyclient that uses an emulated browser to capture the execution of JavaScript code on web pages. `Wepawet` uses anomaly-based techniques to identify signs of maliciousness that are typical of a drive-by download attack.

We submitted the iframes that contained advertisements to `Wepawet`, which executed all the JavaScript code and captured all the network traffic. Finally, with the use of specific heuristics, such as the download of malicious executables or machine learning models, `Wepawet` classified the advertisements as malicious or not.

3.2.2 Malware and Phishing Blacklists

Blacklists are one of the most widespread techniques to protect users against malicious websites. In a domain-based blacklist, a domain is added to the list as soon as it is discovered to host malicious content. Additionally, the domain is classified based on its behavior, such as malware distribution, phishing attempts, stealing users' credentials, and others. In this study, we used a tracking system that constitutes a collection of 49 antivirus, spam and phishing blacklists [17]. We utilized these blacklists by checking against them all the domains we monitored to serve advertisements. Note that it is fairly common for the blacklists to produce false positives. In our study, we wanted to minimize the risk of false classification of an advertisement. To do so, we use an empirically calculated threshold. More precisely, to increase the accuracy of our results, we considered domains as malicious only if they were contained in more than five different blacklists at the same time.

3.2.3 VirusTotal

Among the malicious advertisements, there exist some that try to lure users to install software in their machines. They disguise the software as a media player or an up-to-date browser plugin required to display specific content. Most of the time, this software contains malware that tries to infect the user. Nevertheless, there will be some cases in which a benign plugin is required by the browser to display the content. For instance, a browser could not display Flash content due to Flash plugin absence. Hence, we need a way to decide whether the downloaded software is benign or malicious.

Antivirus products are the best solution for this classification. However, not all vendors can recognize the same malware. Additionally, having access to multiple antivirus products is a time and resource consuming process. Fortunately, `VirusTotal` can solve this problem. `VirusTotal` is an online service[2] that analyzes files using 51 different antivirus products and scan engines to check for malware. One can submit samples to `VirusTotal` and get a report with the classification of these samples by different antivirus companies. We consider `VirusTotal` as a key component of our oracle. Whenever an advertisement tried to force a user to download software, we forward this software to `VirusTotal` and retrieve its classification. This way, we can accurately decide if the downloaded software is benign of malicious.

4. ANALYZING MALVERTISEMENTS

In this section, we analyze the malicious advertisements that we discovered. In particular, we study various aspects of malvertising and try to understand what types of websites are more prone to malvertisements. Furthermore, we investigate whether a website is more secure by selecting a trusted ad network to serve the advertisements. Finally, we examine if the publishers take the users' security into their consideration and thus, take actions to protect their visitors from being infected.

[1] `https://easylist.adblockplus.org`

[2] `http://virustotal.com`

Type of maliciousness	#Incidents
Blacklists	4,794
Suspicious redirections	1,396
Heuristics	309
Malicious executables	68
Malicious Flash	31
Model detection	3

Table 1: Classification of malvertisements.

4.1 Type of Maliciousness

To investigate to what extent cyber-criminals utilize advertisements to promote their nefarious activities, we analyzed the collected advertisements. For this purpose, we used the following procedure: Initially, we retrieved all the analysis reports from `Wepawet`. Then, we examined the reports looking for the existence of specific heuristics like redirects to NX domains or benign websites like Google and Bing, which suggest the utilization of cloaking techniques. Additionally, we looked for behaviors (models) that are similar to previously-known malicious behaviors. Next, all the executables and Flash files were validated against `VirusTotal`. Finally, we used the previously-mentioned blacklists to monitor if the content of the advertisement was served by a blacklisted domain. Table 1 shows the results of all the misbehaving advertisements that we detected. In general, we identified 6,601 incidents in which the advertisements triggered our detection framework. Surprisingly, we observed that about 1% of all the collected advertisements show a malicious behavior.

4.2 Identifying Risky Advertisers

In the next experiment, we wanted to investigate if there is any preference from the side of the malicious advertisers to specific ad networks. In other words, we wanted to measure if some ad networks are more prone to serving malicious advertisements than others. As we already mentioned, each ad network applies its own policy regarding the acceptance of an advertisement. For instance, some of the biggest ad networks do not allow the promotion of websites infected with malware while others, usually smaller in size, are more tolerant to this. Figure 1 illustrates the proportion of malvertisement in the total advertisements served by an ad network. The ad networks are sorted based on the ratio of malicious ads compared to the legitimate ones served. As we can observe, there are some ad networks that are preferred by cyber-criminals, and therefore show more malicious ads. Interestingly, there are ad networks in which the malvertisements underlie more than a third of their global traffic. Note that in this figure we only display the ad networks that contain at least one malvertisement and omit all these that are able to successfully filter them.

Although the existence of ad networks that serve malvertisements constitute a threat for the users of the Web, the size of this threat can only be quantified if we measure the proportion that these ad networks have in the total served advertisements. Figure 2 shows that most of these ad networks send only a small degree of malicious advertisements. This verifies our initial statement that the bigger ad networks tend to perform a more accurate filtering of the advertisements they serve compared to smaller networks. Nevertheless, we spotted a specific ad network that served almost

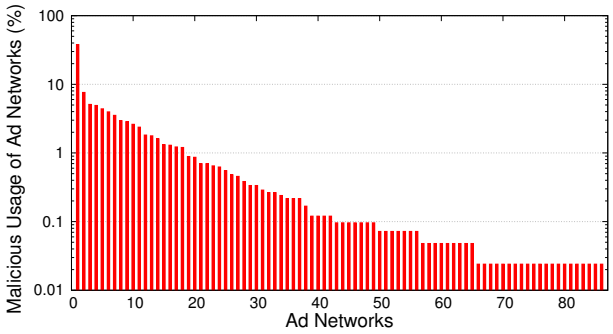

Figure 1: Malvertising distribution from selected ad networks.

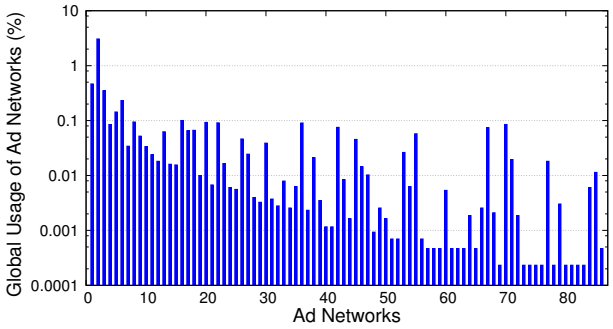

Figure 2: Distribution of advertisements from selected ad networks.

3% of the total advertisements and was responsible for a significant amount of the detected malvertisements. This gave us a significant insight on the ad network filtering mechanisms, which is that no matter how sophisticated the filters used by the ad networks are, there exists a possibility that the cyber-criminals can successfully evade them.

Next, we created three major clusters of websites. The first cluster contained the top 10,000 websites from Alexa's one million top-ranked websites, the second cluster the bottom 10,000, and the third more than 23,000 websites that existed in our advertisement dataset and did not belong to the previous clusters. We wanted to measure from which websites we observe the majority of the malvertisements. We discovered that the first cluster served 82.3% of the whole malvertisements, while the second 6.2%, and the third 11.5%. One can consider that the more famous a website is, the better techniques are applied to protect its visitors. However, the recent event occurred in Yahoo! confirm our hypothesis [11]. In detail, when users visited Yahoo!'s website between 31 December 2013 and 4 January 2014, they were served with malvertisements. Given a typical infection rate of 9%, this incident likely resulted in around 27,000 infections every hour.

In order to discover if the top websites receive more malvertisements because they display more advertisements on their web pages compared to the bottom websites, or whether they are simply preferred by cyber-criminals, we measured the number of the total advertisements (both benign and malicious) the previous clusters displayed. The results revealed that the first cluster served 76.6% of the total advertisements, the second 11.6%, and the third 11.8%. These

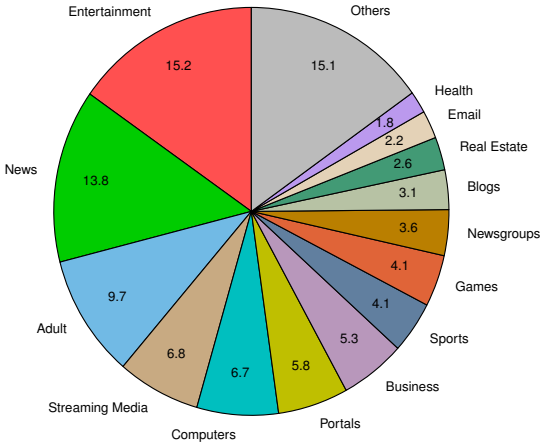

Figure 3: Websites categorization that served malvertisements.

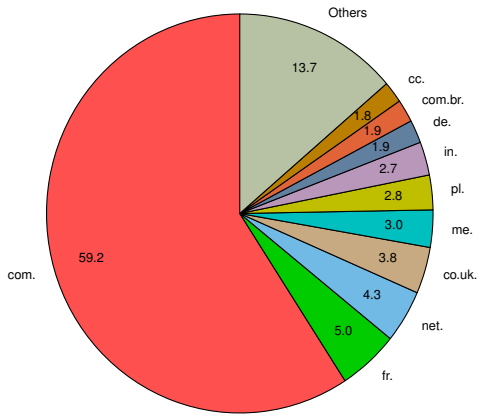

Figure 4: Malvertisement distribution based on top level domains.

results are close to the previously-mentioned malvertising results. Consequently, we believe that miscreants are not interested from which website their malicious code will be delivered, but they are actually concerned about the total amount of infections they will earn from malvertising.

To understand the type of websites that malvertisements are usually targeting, we clustered all the websites we spotted with malvertisement into major categories. Figure 3 shows this categorization. Websites that contain entertainment and news content constitute almost one third of the total websites targeted by malvertisement. Interestingly, the websites that contain adult material are ranked third in the preference of miscreants. This fact conflicts with previous studies, which showed that adult content is tied to increased maliciousness [30].

Finally, we wanted to see the quota of top-level domains that serve malvertisements. Figure 4 shows that the `.com` domains constitute the majority of the websites serving malicious advertisements. Additionally, we noticed that the generic top-level domains (mainly `.com` and `.net`) compose more than 66% of the malvertising traffic. Given the fact that most of the `.com` domains have an American-driven orientation, we believe that malvertising are primarily designed to target United States citizens.

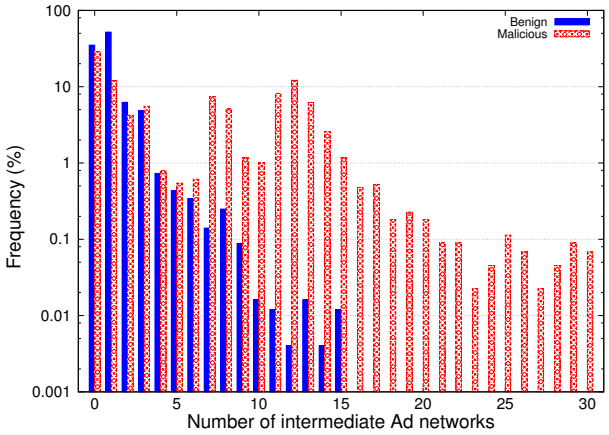

Figure 5: Ad networks involved in ad arbitration for malicious and benign advertisements.

4.3 Ad Arbitration

Website administrators might assume that by using only advertisements from major networks, which are considered trustworthy, they can protect their visitors from potential malvertisements. Unfortunately, this is not the case. There is a practice called *ad arbitration*, which is widely used by ad networks to increase their revenue. During the ad arbitration process, the ad networks buy impressions from publishers as if they were advertisers, and then start a new auction for these ad slots as if they were publishers [25]. Hence, even if an administrator delegates a portion of her website to a specific ad network, she cannot be sure that the advertisements will be only provided by that particular ad exchange.

Although we expected to see a similar behavior in both benign and malicious advertisements, we discovered some cases in which the ad arbitration chain had a much higher length when it came to malvertisements. As we see in Figure 5, in some cases, both benign and malicious advertisements were served directly from the initial ad network. Nevertheless, there were cases in which a specific ad slot participated in up to 15 auctions for benign advertisements and up to 30 auctions for malvertisements. Even though the ad slots that participate in more than 15 auctions constitute only 2% of the malvertisements, we further investigated this phenomenon. Our results revealed that in the initial phases of the auction process, the participants are both popular ad networks and ad networks that we found out being involved in malvertising. However, once the auction process gets longer the last auctions typically happen only among those ad networks that we found to serve malvertisements. An explanation for this could be that smaller and less reputable ad exchanges come into play only when the larger ones failed to obtain an ad slot for a particular arbitration.

Interestingly, we observed ad networks to repeatedly participate in the auctions for the same ad slot. Specifically, we noticed that the same ad networks buy and sell the same slot multiple times. Another interesting fact is the distribution of the ad arbitration chains. Regarding the benign advertisements, the arbitration chain follows a decreasing trend, while, when it comes to malvertisements, it follows a slightly different model. In absolute numbers, the chain follows the same decreasing trend, however, we observe an increase in the frequency of chains in the middle of our graphs.

4.4 Secure Environment

Publishers tend to trust the ad networks that they provide benign advertisements. Hence, they do not secure the environment where the advertisements are displayed. Nikiforakis et al. [22] described the problem of link hijacking, in which advertisements that are contained in iframes redirect the entire web page to an arbitrary destination. This is a serious attack, given the fact that most users open multiple tabs in their browsers for later reading. Hence, the users can be redirected to phishing websites without even noticing that. This problem can be solved in modern browsers with the utilization of the *sandbox* attribute of iframes in HTML5. Unfortunately, none of the websites that we crawled utilized this attribute to protect its users.

5. COUNTERMEASURES

We have shown that malvertising poses a problem to the security of Internet users. In this section, we therefore discuss proactive and reactive countermeasures against malicious advertisements.

5.1 Ad Networks Filtering

Ad networks are the primary mean used by miscreants to deliver their malicious advertisements. Many ad networks have detection mechanisms that successfully filter malvertisements. Yet, there exist others that have poor filtering processes, which are unable to completely eliminate this threat. We believe that collaboration among the ad networks can bring better results in defending against malvertisements compared to individual actions. For instance, the existence of a common blacklist where all the malicious advertisements will be submitted can prevent attackers from submitting their malvertisements to a different network if they get rejected from a former one. Another, more drastic, solution will be penalizing of the ad networks which are inefficient to detect the malicious code embedded in advertisements. For instance, forbidding from participating in ad arbitrations for a certain amount of time, or the application of similar penalties, when an ad network is found delivering malvertisements, can boost the ad networks to invest in better detection algorithms.

5.2 Last Line of Defense

In the case that a malicious advertisement can successfully bypass the filtering mechanisms deployed by ad networks, there should exist reactive countermeasures to protect the users from being infected. Li et al. [18] proposed a browser-based protection mechanism, which can utilize the knowledge of malicious ad paths and their topological features to raise an alarm when a user's browser starts visiting a suspicious ad path, protecting the user from reaching an exploit server. Scarecrow [31], on the other hand, triggers false alarms in a user's browser causing to malicious code, which wants to remain hidden from detection systems, not to be executed. Finally, the safest way for users to protect themselves against malvertisements is to utilize solutions like `Adblock Plus` [1] to prevent advertisements from being delivered to their browsers. Although these solutions appear as the most secure way for the users to protect themselves, and it is already being used by a significant portion of the Web population, a universal adoption of this approach can cause a domino effect in the Internet's economy.

6. RELATED WORK

Detecting malvertisements falls under the category of detecting drive-by downloads. Stringhini et al. [26] and Mekky et al. [20] used the properties of HTTP redirections to identify malicious behavior. Provos et al. [23] introduced *Google Safebrowsing* with the use of high-interaction honeypots. Ford et al. [10] focused on malicious flash-based advertisements by using dynamic and static analysis techniques. A more ad-specific approach was followed with MadTracer, a tool that inspects the advertisement delivery processes and detects malicious activities with machine learning.

Instead of detecting malicious advertisements, AdJail [27] focuses on content restriction policies against third-party advertisements. ADSandbox [8] infers maliciousness by executing the suspected JavaScript in an isolated environment and observing the performed actions. AdSentry [9] works in a similar way, by executing advertisements in a sandboxed JavaScript engine with control over the interactions with the user's visited page.

Regarding the malvertising detection techniques, previous works focused on various aspects of detecting click-fraud. Majumdar et al. proposed a content delivery system to verify broker honesty under standard security assumptions [19]. Metwally et al. [21] and Zhang et al. [32] on the other hand proposed algorithms to efficiently detect duplicate clicks. Additionally, Daswani and Stoppelman [5] investigate the ways that malware can exploit ad networks. Immorlica et al. [15] studied fraudulent clicks and presented a click-fraud resistant method for learning the click through rate of advertisements. Finally, Kintana et al. [16] created a system designed to penetrate click-fraud filters to discover detection vulnerabilities.

Studying the operations of ad networks is recent in the literature. Guha et al. [13] explored different classes of advertising, like search, contextual, and social networks. Vallina-Rodriguez et al. [28] studied the mobile advertisement ecosystem and how mobile ads introduce energy and network overhead. A financial aspect of advertising was also studied in works of Gill et al. [12]. We differ from these studies as we focus on malvertisements and how these reach the end users.

7. CONCLUSIONS

In this paper, we performed the first large-scale study of ad networks that serve malicious advertisements. We studied various aspects of the advertising ecosystem and observed how malicious advertisements differ from benign ones. In addition, we found that none of the websites that serve advertisements take advantage of new HTML5 features to protect its visitors. Despite any server-side efforts employed by the ad networks, malicious advertisements still reach the end users.

Acknowledgements

This material is based upon work supported by the German Federal Ministry of Education and Research under grant BMBF-01BY1111 (MoBE), by the ARO under Award No. W911NF-09-1-0553, by the ONR under Award No. N00014-09-1-1042, by the Air Force under Award No. FA8750-12-2-0101, and by the NSF under Award CNS-1408632. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the funding agencies.

8. REFERENCES

[1] Adblock Plus. Surf the web without annoying ads! `https://adblockplus.org`, 2014.

[2] A. Barth. RFC 6454: The Web Origin Concept. `http://tools.ietf.org/html/rfc6454`, 2011.

[3] E. Cooke, F. Jahanian, and D. McPherson. The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets. In *USENIX Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI)*, 2005.

[4] M. Cova, C. Kruegel, and G. Vigna. Detection and Analysis of Drive-by-Download Attacks and Malicious JavaScript Code. In *International Conference on World Wide Web (WWW)*, 2010.

[5] N. Daswani and M. Stoppelman. The Anatomy of Clickbot. A. In *USENIX Workshop on Hot Topics in Understanding Botnets (HotBots)*, 2007.

[6] V. Dave, S. Guha, and Y. Zhang. Measuring and Fingerprinting Click-Spam in Ad Networks. In *ACM SIGCOMM Conference on Data Communication*, 2012.

[7] V. Dave, S. Guha, and Y. Zhang. ViceROI: Catching Click-Spam in Search Ad Networks. In *ACM Conference on Computer and Communications Security (CCS)*, 2013.

[8] A. Dewald, T. Holz, and F. C. Freiling. ADSandbox : Sandboxing JavaScript to fight Malicious Websites. In *ACM Symposium on Applied Computing (SAC)*, 2010.

[9] X. Dong, M. Tran, Z. Liang, and X. Jiang. AdSentry: comprehensive and flexible confinement of JavaScript-based advertisements. In *Annual Computer Security Applications Conference (ACSAC)*, 2011.

[10] S. Ford, M. Cova, C. Kruegel, and G. Vigna. Analyzing and Detecting Malicious Flash Advertisements. In *Annual Computer Security Applications Conference (ACSAC)*, 2009.

[11] Fox-IT. Malicious advertisements served via Yahoo. `http://blog.fox-it.com/2014/01/03/malicious-advertisements-served-via-yahoo/`, Jan 2014.

[12] P. Gill, V. Erramilli, A. Chaintreau, B. Krishnamurthy, K. Papagiannaki, and P. Rodriguez. Follow the Money: Understanding Economics of Online Aggregation and Advertising. In *ACM SIGCOMM Conference on Internet Measurement (IMC)*, 2013.

[13] S. Guha, B. Cheng, and P. Francis. Challenges in Measuring Online Advertising Systems. In *ACM SIGCOMM Conference on Internet Measurement (IMC)*, 2010.

[14] IAB. Internet Advertising Revenue Report. `http://www.iab.net/media/file/IAB_Internet_Advertising_Revenue_Report_FY_2013.pdf`, 2014.

[15] N. Immorlica, K. Jain, M. Mahdian, and K. Talwar. Click Fraud Resistant Methods for Learning Click-Through Rates. *Internet and Network Economics*, pages 34–45, 2005.

[16] C. Kintana, D. Turner, J.-Y. Pan, A. Metwally, N. Daswani, E. Chin, and A. Bortz. The Goals and Challenges of Click Fraud Penetration Testing Systems. In *International Symposium on Software Reliability Engineering*, 2009.

[17] M. Kührer and T. Holz. An empirical analysis of malware blacklists. *Praxis der Informationsverarbeitung und Kommunikation*, 35(1):11–16, 2012.

[18] Z. Li, K. Zhang, Y. Xie, F. Yu, and X. Wang. Knowing Your Enemy: Understanding and Detecting Malicious Web Advertising. In *ACM Conference on Computer and Communications Security (CCS)*, 2012.

[19] S. Majumdar, D. Kulkarni, and C. V. Ravishankar. Addressing Click Fraud in Content Delivery Systems. In *IEEE Conference on Computer Communications (INFOCOM)*, 2007.

[20] H. Mekky, R. Torres, Z.-L. Zhang, S. Saha, and A. Nucci. Detecting Malicious HTTP Redirections Using Trees of User Browsing Activity. In *IEEE Conference on Computer Communications (INFOCOM)*, 2014.

[21] A. Metwally, D. Agrawal, and A. El Abbadi. Duplicate Detection in Click Streams. In *International Conference on World Wide Web (WWW)*, 2005.

[22] N. Nikiforakis, F. Maggi, G. Stringhini, M. Z. Rafique, W. Joosen, C. Kruegel, F. Piessens, G. Vigna, and S. Zanero. Stranger Danger: Exploring the Ecosystem of Ad-based URL Shortening Services. In *International Conference on World Wide Web (WWW)*, 2014.

[23] N. Provos, P. Mavrommatis, M. Rajab, and F. Monrose. All Your Iframes Point to Us. In *USENIX Security Symposium*, 2008.

[24] B. Stone-Gross, R. Abman, R. Kemmerer, C. Kruegel, D. Steigerwald, and G. Vigna. The Underground Economy of Fake Antivirus Software. In *Workshop on Economics of Information Security (WEIS)*, 2011.

[25] B. Stone-Gross, R. Stevens, A. Zarras, R. Kemmerer, C. Kruegel, and G. Vigna. Understanding Fraudulent Activities in Online Ad Exchanges. In *ACM SIGCOMM Conference on Internet Measurement (IMC)*, 2011.

[26] G. Stringhini, C. Kruegel, and G. Vigna. Shady Paths: Leveraging Surfing Crowds to Detect Malicious Web Pages. In *ACM Conference on Computer and Communications Security (CCS)*, 2013.

[27] M. Ter Louw, K. T. Ganesh, and V. Venkatakrishnan. AdJail: Practical Enforcement of Confidentiality and Integrity Policies on Web Advertisements. In *USENIX Security Symposium*, 2010.

[28] N. Vallina-Rodriguez, J. Shah, A. Finamore, Y. Grunenberger, K. Papagiannaki, H. Haddadi, and J. Crowcroft. Breaking for Commercials: Characterizing Mobile Advertising. In *ACM SIGCOMM Conference on Internet Measurement (IMC)*, 2012.

[29] K. C. Wilbur and Y. Zhu. Click Fraud. *Marketing Science*, 28(2):293–308, 2009.

[30] G. Wondracek, T. Holz, C. Platzer, E. Kirda, and C. Kruegel. Is the Internet for Porn? An Insight Into the Online Adult Industry. In *Workshop on Economics of Information Security (WEIS)*, 2010.

[31] A. Zarras. The Art of False Alarms in the Game of Deception: Leveraging Fake Honeypots for Enhanced Security. In *IEEE International Carnahan Conference on Security Technology (ICCST)*, 2014.

[32] L. Zhang and Y. Guan. Detecting Click Fraud in Pay-Per-Click Streams of Online Advertising Networks. In *International Conference on Distributed Computing Systems*, 2008.