Matthew Lee
5/6/2018
COMP116
Final Project

An Overview of Malicious Advertising

I. Abstract

Advertisements are a 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.[1] 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.[2] The issue really came to the attention of the public in 2009 when from September 11th to 14th, users of The New York Times became infected after falling to a trick that asked users to install disguised malicious software.[3] 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.[4] 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.[5] 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

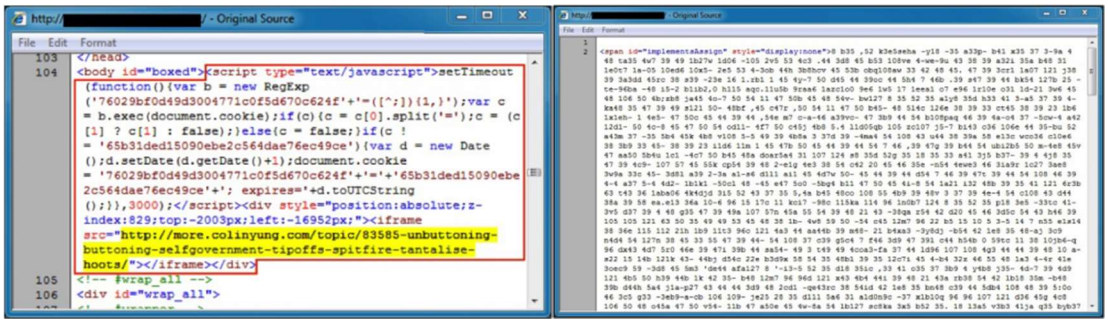
a. Exploit Kits

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

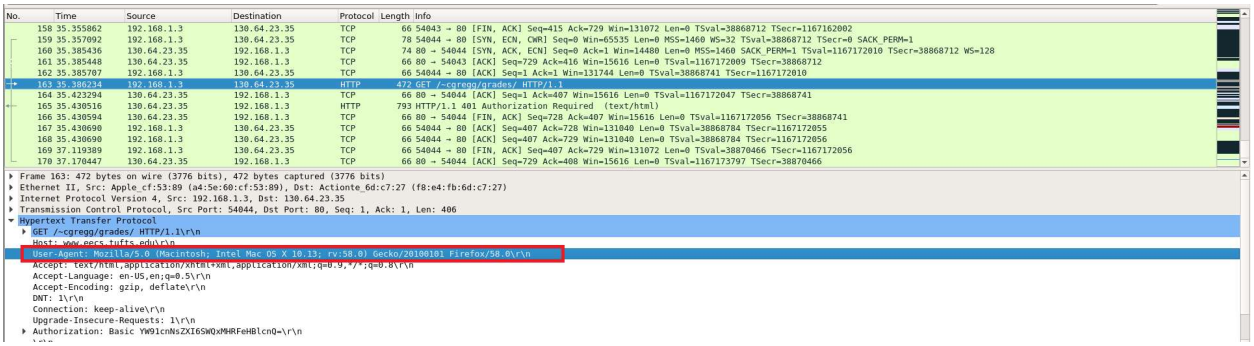
simple to use service to malicious individuals on the black market. Many different exploit kits exist such as the Blackhole and Neptune exploit kits, but in this section, I will focus in specifically on the Angler exploit kit, which was one of the most popular exploit kits associated with malvertising 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.[6]

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.[7]

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.[8] 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.



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:



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.[7] 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.[9] 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 incredibility 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.[10]

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.[11] 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."[9]

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.[12] 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 affects 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/comp116_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/.

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.

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.

[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.

[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.

[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.

[5] "Spotify Ads Hit by Malware Attack." *BBC News*, BBC, 29 Mar. 2011, 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/.

[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/;

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." *Heimdal Security Blog*, 27 Dec. 2016, 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.

[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*.