Demystifying Ad Fraud

Shilpa Kumari, Xiaohong Yuan, Joshua Patterson, Huiming Yu
Department of Computer Science
North Carolina A&T State University
Greensboro, NC, USA

***Abstract*—The wide spread use of Internet allows advertisers to reach significantly more consumers through online advertisement compared to traditional advertising media. However, currently online advertisement is facing challenges associated with advertisement (ad) frauds such as ad replacement, ad stacking, click fraud, and click hijacking. It is important to teach students this emerging topic, and help students understand the attack flow behind ad frauds. In this work, a course module was developed to teach students about online ad servicing architecture, the associated security vulnerabilities and how the vulnerabilities can be exploited. A scenario of ad replacement was developed which includes an infrastructure to simulate the ad replacement attack flow providing students a real-world context and hands-on experience. This paper describes the course module on ad fraud, and our teaching experience of this course module. Instructors teaching network security, web security, information systems, and business & economics could adopt this course module.**

Keywords—advertisement fraud; ad replacement; click hijacking; DNS cache poisoning

I. INTRODUCTION

Internet-based advertising is transforming the advertisement industry by allowing advertisers to deliver information to consumers who value the information the most and are most likely to act in a cost effective manner compared to traditional advertisement approaches [1]. In a traditional approach, an advertiser will create and distribute printed content to relevant consumers. Finding the relevant consumers is challenging and requires a large amount of resources. In contrast, the online approach can easily find relevant consumers by creating their Internet browsing profile. Such an approach can significantly reduce wasted ad impressions. Furthermore, content of online advertisement can be easily changed and propagated to consumers with much faster speed compared to traditional advertisement media such as newspaper, telephone, television and billboards. Additionally, unlike traditional methods, online advertisement allows advertisers to track their consumers in real time and see what is or is not working for the business. Therefore advertisers can adapt quickly and refine their strategy. Moreover, online advertisement provides far greater exposure compared to traditional media. With one ad on Internet, an advertiser can reach consumers anywhere in the world at a much lower cost compared to traditional advertisement media.

The benefits of online advertisement come with several challenges. There are widespread concerns about user privacy because advertisement networks gather a great deal of user information such as search histories, web browsing behaviors, and online social networking profiles [2, 3]. Moreover, several types of advertisement frauds such as click fraud, ad replacement, click hijacking, and ad stacking pose a significant challenge in front of online advertisement industry. Today these activities are widespread generating fraudulent revenue. For example, 10 to 15% of ads in pay per click online advertisement revenue model are not authentic [4]. These clicks are generated by botnets or individuals trained to click on ads in order to increase the revenue of the websites which display such ads (called click fraud). Click fraud is the most commonly seen advertisement fraud. Federal Bureau of Investigation (FBI) has discovered advertisement replacement fraud and click hijacking attack [5]. In advertisement replacement fraud, malicious publisher used domain name server (DNS) changer malware and rogue DNS to replace legitimate advertisements on websites with substituted advertisements that triggered payments to the malicious publisher. In click hijacking, the attacker infected the computers with malware and when the user of an infected computer clicked on a search result displayed through search engine, the malware rerouted the request to a different website designed by the attacker and the attacker fraudulently generated click triggered revenue from its advertisement network. With these two attacks, malicious publishers/attackers made millions of dollars under their advertisement agreements, not by legitimately displaying advertisements through their publisher networks, but by fraudulently driving internet traffic to websites and ads that earn them money [5]. These examples show that it is of great importance to develop technologies for counteracting advertisement frauds for a fair and transparent online advertisement ecosystem. It is also necessary to introduce this emerging security topic to students of cybersecurity, computer science, information systems, and business and economics.

Based on the security breach/crime of ad fraud [5], and the related research [6], we developed a course module to introduce the topic of ad fraud to university students. This course module is designed to teach students about online ad servicing architecture, major players in ad serving architecture, security vulnerabilities of online advertising and how the vulnerabilities can be exploited. A scenario on ad replacement was developed based on the paper titled “Dissecting Ghost Clicks: Ad Fraud via Misdirected Human Clicks” [6]. The course module has three parts: a) a power point presentation with animation, b) a hands-on lab exercise showing ad replacement attack in action using three virtual machines for mimicking genuine publisher, malicious publisher and DNS server, and c) a set of questions designed for students to

978-1-5090-5920-1/17/$31.00 ©2017 IEEE

discuss. Hands-on-labs are strongly supported by educational theory as a productive and pedagogical practice [7]. Cyber security students are typically familiar with stories of security breaches as reported in the media but do not have technical grasp of necessary steps involved in those breaches [8]. Hands-on lab exercises help students understand the technical mechanism involved in the breaches.

This paper is organized as follows. Section II provides a background in ad serving architecture, ad revenue model, commonly seen advertisement frauds, and DNS cache poisoning. Ad replacement fraud is described in Section III. Section IV describes the teaching method used for this module. Section V describes our teaching experience. Section VI concludes this paper.

II. Background

In online advertisement typically three parties are involved: a) advertiser, b) publisher, and c) ad network. Advertiser is an entity that would like to advertise their product. Publisher is a website which hosts the advertiser's ad. Ad network is a mediator between publishers and advertisers. Ad network creates appropriate ads for its advertisers (stored on ad network's ad server) and embeds those ads in a publisher's online content based on the user's browsing profile. An advertiser pays ad network for managing its ad campaign and ad network splits revenue generated by the ads with the publishers. The following subsections describe ad serving architecture, ad revenue models, commonly seen advertisement frauds, DNS cache poisoning and DNS pharming in detail.

A. Ad Serving Architecture

Fig. 1 shows an ad serving architecture that is commonly used [9]. First, a user visits a website and the browser issues a request for the webpage (Step 1). The publisher sends the webpage content to the user that contains the publisher's own content and a block of HTML code provided by the ad network (Step 2). The HTML code redirects the browser to communicate with an ad server (Step 3). Then the user's browser requests a script from the ad server. Ad server sends the script that collects certain parameter from user's local machine, which influences the ad selection by the ad server (Step 4). This includes HTTP cookies if the ad server deposited them during previous interactions. The user's browser then executes the script and information collected by the script is communicated back to the ad sever along with the request for ads (Step 5). The ad server then chooses the most appropriate ad for the given user and sends it (Step 6).

B. Ad Revenue Models

There are three revenue models commonly used in online advertisement [8]: a) cost per mille (CPM), b) cost per click (CPC), and c) cost per action (CPA). These are used to measure the cost effectiveness and profitability of online advertisement. CPM is the amount advertisers are charged for displaying an ad for 1000 times. CPM is the closest online advertising strategy to those offered in other traditional media such as television, radio, or newspapers that sell

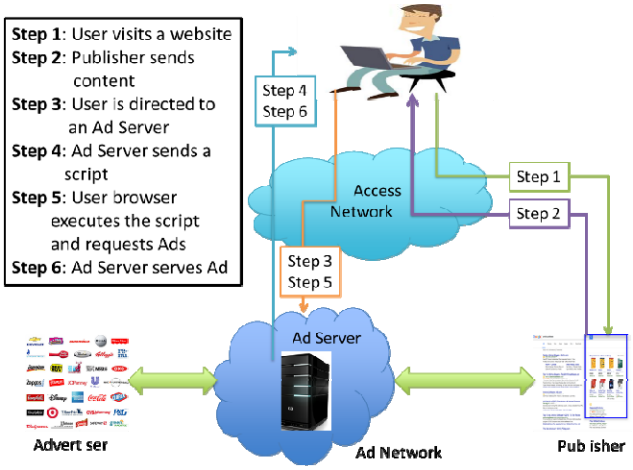

Fig. 1. Online ad serving architecture

advertising based on estimated viewership, listenership or readership [10]. CPC is the amount advertisers are charged per click. CPC is a more effective metric than CPM because it tells how effective the advertisement is. However it is susceptible to click fraud where trained human power or botnets simply click on an ad without any actual interest in that ad to generate CPC revenue. CPA is the amount advertisers are charged per action such as an online sale. CPA is a better metric compared to CPC because it is not susceptible to click fraud.

C. Common Advertisement Frauds

Impression ad fraud is one of the most commonly seen ad fraud. Impression frauds can take several forms. In one form, it involves fabricating HTTP requests to either the publisher's page or the ad server directly to artificially inflate the actual amount of traffic [11]. This type of fraud mainly targets CPM revenue. Another form of impression ad fraud is ad stacking where ads were stacked on top of each other in the same space but only the top ad is visible. However, the lower layer ads are reported as viewed ads to the advertiser. Click fraud is typically more profitable than impression fraud. In click fraud, a person, automated script or computer program imitates a legitimate user of a web browser clicking on an ad for the purposes of generating CPC revenue without having actual interest in the target of the ad's link [12]. Click hijacking is a fraud discovered by FBI in 2011 [5]. In click hijacking, when a user of an infected computer clicks on a search result link displayed through a search engine query, malware causes the user to be brought to a different website (designated by attackers) instead of the website to which the user asks to go by clicking on a search result. Using this attack, attackers can earn CPC revenue from their associated ad network by hijacking the clicks.

D. DNS Cache Poisoning and DNS Pharming

Ad replacement attack exploits DNS cache poisoning in performing DNS pharming. This section briefly describes DNS cache poisoning and pharming. DNS translates domain names to IP addresses and vice versa. A DNS server typically stores mapping of domain names and IP addresses of frequently received queries in its cache called DNS cache. When a DNS server or resolver receives a query, it searches its cache for a matching mapping. If it finds the matching then response is

sent back to the requestor. However if there is no matching then the DNS resolver can either return a referral response closer to the domain name which is the subject of the query or the resolver can itself initiate the same query to an authoritative DNS sever responsible for the domain name [13]. Each query is identified by a random 16-bit transaction ID. The authoritative server can respond with an answer, a referral, or a failed response. In DNS cache poisoning, an attacker flood DNS resolver with forged responses pretending to be authoritative server's response [14]. In this way an attacker can update the resolver's DNS cache with a malicious IP address corresponding to the domain name requested in the query. DNS pharming is an attack where a user is taken to a different, usually malicious website instead of the one he wanted to visit. An attacker performs DNS pharming attack by poisoning the DNS cache.

III. Ad Replacement Attack Flow Simulation

In ad replacement fraud, attackers are malicious publishers who infected victim machine with DNS changer malware. Using DNS changer malware and rogue DNS servers, the malicious publishers replaced legitimate advertisements on websites with substituted advertisements, which triggered payments to them [5]. For example, when the user of an infected computer visited the ESPN website, an advertisement for "Dr. Pepper Ten" had been fraudulently replaced with an ad for a hotel business [5]. Fig. 2 shows an ad replacement attack scenario and the detailed attack flow. It involves four websites namely a) www.ncatgadget.com, b) www.ncatcamera.com, c) www.ncatmalicious.com and d) www.ncatmobile.com. Here ncatgadget.com, ncatcamera.com, ncatmalicious.com and ncatmobile.com are playing roles of genunine publisher, legitimate advertiser, malicious publisher and substituted advertisement respectively. Ideally ad of ncatcamera should be displayed on ncatgadget, however, with ad replacement attack, ad of ncatcamera is replaced with ncatmobile, which is linked with malicious publisher ncatmalicious.

To illustrate the scenario in a hands-on exercise environment, three virtual machines are used: VM1, VM2 and VM3. They represent victim machine, host of malicious publisher and malicious DNS resolver respectively. VM1 also hosts ncatgadget, ncatcamera, and ncatmobile websites. The attack steps are described below:

Step 1: First the victim machine tries to access website ncatgadget.com. Then the browser asks to resolve the IP address of this website from the DNS resolver.
Step 2: The DNS resolver resolves this correctly and provides 10.0.2.4
Step 3a: This page contains an ad from ncatcamera.com inside an iframe. The browser then asks the DNS Resolver to resolve IP address of ncatcamera.com

Step 3b: The DNS Resolver gives an incorrect resolution 10.0.2.5 of this ad website in order to take the victim to a malicious webpage www.ncatmalicious.com and replace the Ad
Step 4a: The victim machine then visits this malicious machine
Step 4b: This malicious machine then provides an iframe ncatmalicious.com of exactly the same size as the iframe on the webpage ncatgadget.com in order to look legitimate.

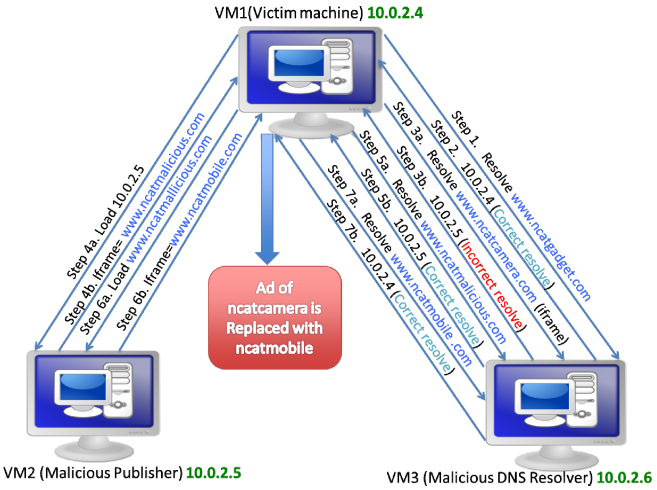

Fig. 2. Ad replacement attack flow

Step 5a: The victim machine requests the DNS Resolver to provide IP address of ncatmalicious.com
Step 5b: The malicious resolver resolves this correctly 10.0.2.5, in order to take user to malicious page and serve an ad with which the malicious publisher is linked to, and make money.
Step 6a: The victim machine then loads ncatmalicious.com
Step 6b: This webpage contains an iframe of ncatmobile.com. This is the ad website with which www.ncatmalicious.com is linked to. Hence, ncatmobile.com iframe is served by ncatmalicious.com
Step 7a: The victim machine asks the DNS Resolver for the IP Address of ncatmobile.com.

Step 7b: The malicious resolver resolves this correctly and provides the IP Address 10.0.2.4 to the victim.

Hence, finally the victim machine loads the ad from ncatmobile.com. So the ad from ncatcamera.com is replaced with ad from ncatmobile.com. By replacing the ad of ncatcamera with ncatmobile, ncatmailcious is fraudulently generating revenue from ncatmobile. Moreover, ad network associated with ncatcamera and publisher website (ncatgadget) which victim machine originally visited is losing ad revenue.

IV. Teaching Method

A course module was developed to teach students the topic of ad fraud. The course module has the following learning objectives:

1) Students will be able to explain different components of ad serving architecture
2) Students will be able to explain DNS cache poisoning and pharming
3) Students will be able to explain commonly known ad frauds

4) Students will be able to explain ad replacement attack flow in detail
5) Students will be able to discuss the impact of ad replacement attack on different players
6) Students will be able to configure DNS server and web server.

The course module has three components: a) an animated power point presentation, b) a hands-on lab exercise, and c) a set of discussion questions. As a first step, students were given an animated power presentation which visually describe the fundamentals of ad serving architecture, ad revenue model, DNS cache poisoning and DNS pharming. It also includes detailed attack flow of ad replacement attack, click hijacking attack, and several mitigation techniques for ad fraud. The main purpose of this presentation is to teach the fundamental concepts and prepare students for the hands-on lab exercise.

As a second step, students were asked to simulate the attack flow shown in Fig. 2 in a hands-on lab exercise. In order for students to simulate the attack flow using reasonable computing resources they were provided UBUNTU virtual machine (VM) image from SEED labs [15]. Students were instructed to make three copies of VM image representing VM1, VM2 and VM3 shown in Fig. 2. In this way, students were able to run three VMs on a single host machine and simulate the attack flow. Students were provided instructions to connect these VMs on a local network. BIND 9 [16] was used for DNS sever and Apache was used for web server for running different websites. The UBUNTU VM image came with preinstalled BIND 9 and Apache web server. Students were provided instructions to configure DNS server and web server. Students were asked to simulate the attack flow and submit screenshots from different steps in the attack flow.

As a third step, students were provided a set of questions to answer. The questions were designed to check the fundamental understanding of students on ad fraud.

V. Teaching Experience

This course module was taught in a graduate level course Secure Software Engineering at North Carolina A&T State University in spring 2016. This was an online course. Fifteen (15) students were enrolled in the class. The students were given the animated power point presentation, and a detailed lab menu on ad replacement attack with discussion questions embedded in the document. Students were given 10 days to submit the assignment. After that, an anonymous student survey was conducted on the course module. This section presents the results from student survey.

Thirteen students participated in the survey. Fig. 3 shows students' self-ranking of their knowledge/skill level in different learning objectives. It shows most of the students consider themselves as having excellent and high knowledge in different learning objectives after learning this course module.

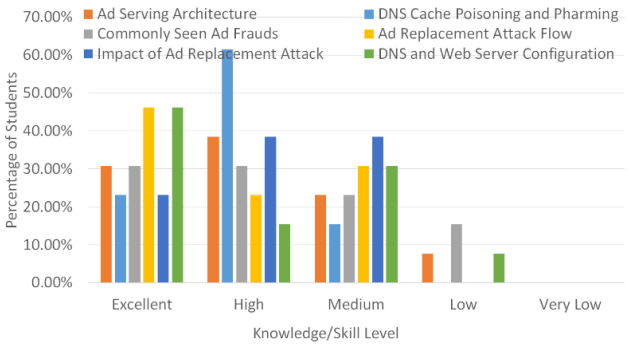

Fig.3. Knowledge/skill-level in learning objectives

Thirty eight percent (38%) of students strongly agreed and 54% of students agreed that learning objectives were met. 46% of students strongly agreed and 54% of students agreed that they enjoyed doing the lab exercise. Furthermore, 58% of students strongly agreed and 42% of students agreed that they were motivated to learn about ad fraud and security vulnerabilities leading to fraud.

On the effectiveness of teaching methods, 38% of students strongly agreed and 54% of students agreed that the animated power point presentation was helpful in understanding the material. Moreover, almost all students found that hands on lab exercise helped them better understand the material compared to animated power point alone.

VI. Conclusion

This paper describes a course module designed to teach students about online ad serving architecture, the security vulnerabilities associated with online advertising, and how the vulnerabilities can be exploited. The course module consists of an animated power point presentation and a hands-on lab exercise simulating ad replacement attack. The hands-on lab was designed based on the real life security breach/crime and research related to it. In the hands-on lab exercise students were provided UBUNTU virtual machine image with pre-installed necessary software. Students were taught different steps in ad replacement attack in detail and were asked to simulate these steps using the provided virtual machine image.

The course module was taught in spring 2016 in an online class. Our teaching experience showed that this course module was well received by the students. This course module could be adopted by instructors teaching network security, web security, information systems, and business and economics.

In the current lab exercise, DNS cache was statically poisoned by changing the DNS mapping in DNS resolver. As a future work, this step could be replaced by asking students to implement dynamic DNS cache poisoning.

REFERENCES

[1] Evans, David S., "The Online Advertising Industry: Economics, Evolution, and Privacy", Journal of Economic Perspectives, Forthcoming, April 2009

[2] Guha, Saikat, Cheng, Bin, Francis Paul, "Challenges in measuring online advertising systems", Proceedings of the 10th ACM SIGCOMM conference on Internet measurement (IMC '10). New York, NY, USA,

[3] Goldfarb, Avi, Tucker, Catherine, "Privacy Regulation and Online Advertising", Journal of Management Science, 2011

[4] Haddadi, Hamed, "Fighting online click-fraud using bluff ads", ACM SIGCOMM Computer Communication Review, Rev. 40, 2 (April 2010), 21-25

[5] FBI. New York Field Office. Manhattan U.S. Attorney Charges Seven Individuals for Engineering Sophisticated Internet Fraud Scheme That Infected Millions of Computers Worldwide and Manipulated Internet Advertising Business. Retrieved August 19, 2016 from https://www.fbi.gov/newyork/press-releases/2011/manhattan-u.s.-attorney-charges-seven-individuals-for-engineering-sophisticated-internet-fraud-scheme-that-infected-millions-of-computers-worldwide-and-manipulated-internet-advertising-business

[6] Alrwais, Sumayah, Gerber, Alexandre, Dunn, Christopher, Spatscheck, Oliver, Gupta, Minaxi, Osterweil, Eric, 'Dissecting ghost clicks: ad fraud via misdirected human clicks', 28th Annual Computer Security Applications Conference (ACSAC '12).

[7] Son, J., Irrechukwu, C. and Fitzgibbons, P., "Virtual Lab for Online Cyber Security Education", Communications of the IIMA, 2012

[8] Mateti, P., "A laboratory-Based course on Internet security," 34th SIGCSE Tech. Symp. Computer Science Education, Reno, NV, Feb. 2003.

[9] Vratonjic Nevana, Manshaei, Mohammad, Hubaux, Jean, "Online Advertising Fraud". Retrieved August 19, 2016 from http://infoscience.epfl.ch/record/165674/files/OnlineAdFraud.pdf

[10] Fain, D. C., Pederson, J., O., "Sponsored Search : A brief history", Bulletin of the American Society for Information Science and Technology, 2005

[11] Gross, B., Stevens, R., Zarras, A., Kemmerer, R., Kruegel, C., Vigna, G., "Understanding Fradualent Activities in Online Ad Exchanges", Internet Measurement Conference, 2011

[12] Daswani, N., Mysen, C., Rao, V., Weis, S., Gharachorloo, S, "Online Advertisement Fraud", Proceedings of Crimeware, 2008

[13] Sooel, S., Vitaly, S. "The Hitchhiker's Guide to DNS Cache Poisoning", Security and Privacy in Communication Networks, 2010

[14] Atkins, D., Austein, D., "Threat Analysis of the Domain Name System (DNS)", RFC 3833, August 2004

[15] SEEDS Lab. Retrived August 19, 2016 from http://www.cis.syr.edu/~wedu/seed/

[16] Bind9 DNS Server. Retrieved August 19, 2016 from http://www.bind9.net/manual/bind/9.3.2/Bv9ARM.ch01.html