A New Approach to Blocking Ads and Ad Fraud

Students: Sumayah Alrwais, Christopher Dunn, Christopher Frisz, Caitlin Orr
Faculty: Arun Chauhan, Minaxi Gupta
School of Informatics and Computing
Indiana University at Bloomington
Role of Ads in Internet Economy

- Money we pay our ISPs maintains routing infrastructure necessary to carry users packets
- Content on the Internet is paid for almost exclusively by ads: most news sites/portals, search engines rely on ad revenue
- Facebook made over three billion USD from ads in 2011, and Google is approximately eight times larger
- Focus of our work: privacy concerns around ads and ad fraud
Social Tracking

- Facebook is the second largest tracker according to Ghostery
  - Social platforms are well represented as trackers
    - Not a requirement, a design decision
- Social platform tracking and ads often leak profile ids to third parties
  - This allows other trackers to link their behavior profile with a user’s social profile
Reasons to Block Ads

• Privacy & not seeing the ad
• In a sample of 155 million URLs, end users retrieved to 44,210 URLs that were blacklisted as spam, phishing, or malware
• Of those, 20% were inside of, or the user was sent to, by an ad network
  – Ad networks represented 10% of all URLs
Part 1: A New Approach to Blocking Ads

• Ads consume a substantial fraction of user bandwidth
  – Particularly draining on mobile devices
• Individual networks often track a user seven times per day
• Users aware of these concerns want to be able to block ads, entirely or selectively
Existing Ad Blockers

- **Plug-ins**
  - Ad Block, Ghostery, TACO, TPLs

- **Proxies**
  - WebSense, Fortigaurd, Ad Muncher, etc.

- **DNS / Host file manipulation**
  - Resolve ad hosts to loopback
Existing Ad Blockers

- All techniques are based on lists of regular expressions
- These lists are manually created
- Elements are rarely removed
- Blocks only URLs or hosts
- Vulnerable to simple changes in the names of ad content URLs and hosts
Dual-use Content

• Dual-use URLs and hosts
  – Must be blocked or allowed in their entirety
  – Blocking such URLs may break functionality
    • Like buttons, Google +1
    • Rarely used, 0.1% of such displays were clicked

• Work-around: Element Hiding
  – Retrieve the content, but do not display portions of it
  – Does not prevent privacy or security violations
Our Approach

• Static Javascript analysis
  – Using an SVM classifier
• Almost all ads use scripts
• Resilient to host name and URL structure changes since it relies on ad behavior
Features

• Compute various features for each script
• Normalize based on the length of the script
  – This way it works on very long or very short scripts
• Fundamentally similar to Term Frequency-Inverse Document Frequency (TF-IDF) used in natural language classification
Features: Code Obfuscation

• Readable Code:
  – document.write("Hello World!"))

• Obfuscated Code:
  • eval(function(p,a,c,k,e,d){e=function(c){return c};if(!".".replace(/^/,String)) {while(c--) {d[c]=k[c]|c} k=[function(e){return d[e]}]; e=function() {return '\w+'}; c=1}; while(c--) {if(k[c]) {p=p.replace(new RegExp('\b'+e(c)+'\b','g'),k[c])}} return p}('0.1("2 3!")',4,4,'document|write|Hello|World'.split('|'),0,{}))
Features: Code Obfuscation

• Code obfuscation can be easily done by freely available tools
• There are tools that can help reverse it
  – But not completely
• It is more often used by ads than content
  – To hide exactly what they are doing
Features: Dynamic Code, HTML, and URLs

• `eval()` takes the text form of code and runs it
  • This allows the code to change while it is being run
• Such changes are more likely on ads
  – Probably to increase their flexibility
• Likewise, ads often dynamically produce URLs and HTML
  – To allow them to select an ad based on the user
Features: Code Structures

• Try-catch blocks are more common in ads
• Such code structures allow the ad to recover from errors or unexpected occurrences and still display
• They are less needed in first party content, since the context and the page it is running on is known in advance
Features: Functions

• Ads often contain functions for changing the dom tree or script behavior
  – eval
  – document.write

• Ads often use functions for extracting info
  – document.referrer
Features: Event Handling

• Events are responses to things such as clicking or when content has finished downloading
  – Ads rarely respond to clicks, they instead deal with them via HTTP redirects
  – Conversely, ads often respond to content loading
    • That way they start playing the ad only when the page is finished downloading
Features: Script origin

- Ad scripts are typically retrieved from third-party servers
- This feature is simple a flag as to whether or not the script comes from another server
  - This is also used in existing blockers
  - A regular expression can have a handful options in Ad Block.
Features: Presence of keywords

• Advertisements sometimes include notes and comments to that effect
  – News sites often have a tag next to ad stating “Advertisement” or “Sponsored Content”
  – Likewise with Google “Ads related to” query

• We include these keywords as a feature
  – Similar to the current lists of expressions
Training and Testing a Classifier

• Visited 339 unique websites selected from the top Alexa ranked sites for countries from every continent, and from every category

• Of these, we selected 250 unique ad scripts, and 250 unique non-ad scripts

• Trained an SVM-based classifier with features discussed before
Classifier Evaluation in the Wild

• Selected 25 additional random websites
  – In the same way as the test & training sites, without overlapping those

• Evaluated every script manually to classify them as ad related or not to establish ground truth

• These sites had 258 ad related scripts and 726 other scripts
Classifier Evaluation in the Wild

• Our classifier had an accuracy of 98% on this set

• Errors were roughly split between false positives and false negatives
  – 12 false positives
    • 6 unique scripts
  – 10 false negatives
    • 4 unique scripts
Part 2: Ad Fraud

• Internet fraudsters want a piece of the multi-billion dollar ad economy

• Many forms of Ad Fraud
  – Clickbots, Hit Inflation, Ad Replacement, etc

• Our focus is on Ad Replacement
  – This attack was taken down and hailed as the largest takedown ever
Overview of Ad Fraud Scheme

• DNS altering malware
• Redirects users to malicious DNS resolvers
  - DNS resolvers convert host names to one or more IPs
• These servers ultimately steal ad display revenue
  – By convincing the ad networks that the attackers are displaying the ads
Overview of Ad Fraud Scheme

• The attack has four main parts
  – Ad Replacement to steal revenue
  – Click hijacking to make it appear that the user clicked on a sponsored search result
  – Phishing
  – Blacklist alterations to prevent detection
Ad Replacement Example

• Step-1: The malicious resolver mis-resolves the ad host name, ad.doubleclick.net

• Step-2: The malicious IP serves an iframe to the victim.
  – The iframe directs the victim to a front-end website to serve an advertisement of the same size as the original ad
Ad Replacement Example

• Step-3: The front-end website, banners.awefulnews.com, is also mis-resolved to the malicious IP.
  – Ensures that anyone scrutinizing the front-end website will be shown legitimate-looking content through a correct resolution

• Step-4: The mis-resolved front-end website serves an iframe. This loads an ad URL which belongs to a different ad network, “Xtend Media”
  – The attacker has an account with this ad network to display its ads on the front-end website.
Ad Replacement Example

• Step-5: The new ad network’s host name, ad.xtendmedia.com, is resolved correctly
  – This is the attacker’s ad, which he wants served

• Step-6: The new ad network serves an ad to the victim. The publisher ID in that ad is set to the attacker’s publisher ID.
  – Thus, the new ad network pays the attacker
  – This also typically increases the amount of tracking as analytics scripts are often not manipulated
Click-hijacking

• Only occurs intermittently
  – Presumably to keep the user from removing the malware or changing their search habits
• Search results would appear correct in a browser
• Clicking on a search link would instead direct the user to one of the attacker’s front sites
Click-hijacking

• It would then either display sponsored search results
  – This can get the attacker revenue for display the results, for networks that pay per impression

• Or it would, in effect, also click on one of those results
  – This gets the attacker revenue, for networks that pay per click
Phishing

- Layout & design follows stand phishing practice
- The main advantage the attacker has is that these phishing site’s URLs appear legitimate due to DNS manipulation
  - As hosts of legitimate domains
Blacklist Fabrication

- This server responds to requests to update blacklists
  - Such as the safe browsing API built-in to Firefox
- It ensures that the attacker’s sites and IPs are not listed
Finding Suspect Resolvers

• We downloaded the malware and extracted suspect resolvers
  – Scans of 6 prefixes, suggested by Trend Micro
    • Located primarily in Eastern Europe
• Malicious resolvers give correct answers for most hosts
• We produced a test set of 1,019 hosts by crawling the top 3,000 Alexa websites for analytics and ad related hosts
Heuristics to Identify Malicious Resolvers

- We compared the suspect DNS resolvers to a broad sample of 4,490 DNS resolvers around the world, covering 74 countries
  - If any IP returned by the suspect resolver was also returned by a valid resolver, we consider the entire resolution to be valid
  - Ad domains typically support SSL
    - Challenge the suspect resolution IP via SSL
    - Compare the certificate against known one
Malicious Resolvers Discovered

• 1,039 malicious DNS resolvers we detected in Eastern Europe
  – Across 4 IP prefixes
  – We had reports of 5
  – Confirmed later to be run out of there

• 1 detected in the 2 North American prefixes
  – We had reports of 99
Modes of Execution of Malicious Resolvers

• We instrumented clients and begin using the malicious resolvers
  – Allowed for the construction of signatures of the attack
  – Identified the vulnerable points in the ad networks

• This was expanded upon further by examining user traffic
Real World Impact of the Attack

• We gathered approximately 155 million HTTP transactions from 17 thousand DSL subscribers
• Located in the United States
• All transactions for a single a day
  – Also used to find additional forms of the attack and more ad networks being attacked
Real World Impact of the Attack

• 37 user lines were infected with DNS changer malware
• Based on our sample, we estimated 3.53 million affected users worldwide
• When it was taken down by the FBI in cooperation with Estonian authorities, they estimated 4 million users
• The attackers stole about 14 million USD over 4 years
Conclusion

• Our approach to blocking ads is fundamentally different from the current approaches.
• It is more robust to evasion. If deployed, it can help privacy concerns arising from ad retrieval and display.
Conclusion

• Ads play a prominent role in the Internet economy. Newer revenue models and fraud containment are needed.
  – Users need defenses against malicious ad content

• Social network ads and tracking need special care, since they can inadvertently link behavior profiles to social identities
Questions?
Ad Replacement Example