Knowing Your Enemy: Understanding and Detecting Malicious Web Advertising

Z. Li, K. Zhang, Y. Xie, F. Yu, and X. Wang

In Proc. ACM CCS, 2012

Dr. Kevin W. Hamlen
Language-based Security
Loading a Web Advertisement

Client (web browser) -> URL Request -> Page Publisher
Loading a Web Advertisement

Client (web browser) → Page Publisher

web page
Loading a Web Advertisement

Client (web browser)

Page Publisher

Ad Network

ad request
Loading a Web Advertisement

Client (web browser) 

Page Publisher

Ad Network

ad tag

tag
Loading a Web Advertisement

Client (web browser)

Page Publisher

Ad Network

Ad Server

tag

URL request
Loading a Web Advertisement

Client (web browser) 

Page Publisher

Ad Network

Ad Server
Ad Syndication

Client (web browser) → Page Publisher

new tag

Ad Network → Ad Syndicator
Ad Syndication

Client (web browser)

Page Publisher

Ad Network

Ad Syndicator

Ad Server

URL request

tag
Ad Syndication

Client (web browser) ➔ Ad Network ➔ Page Publisher ➔ Ad Syndicator ➔ Ad Server
Malicious Advertisements

• Various goals
  • Click fraud
    • Accrue unmerited ad revenue
    • *pay-per-impression* – advertisers pay by number of URL requests for their ads
    • *pay-per-click* – advertisers pay by number of people who click on their ads
    • Malvertisements trick browsers into sending URL requests that are never displayed
    • Malvertisements redirect clicks to ads, generating false clicks
  • Scams / Phishing
    • Collect private user information (credit card info, usernames/passwords, etc.)
    • Impersonate legitimate sites (e.g., your bank)
    • Harvested info used in other criminal activities (identity theft, spam, etc.)
  • Drive-by-download
    • Infect client machine with malware
    • Exploit browser vulnerabilities
    • Infections facilitate other attacks (botnet zombies, ransomware, all of the above attacks)
Two Steps of Malvertising

• Enablers
  • ad syndicators
  • malicious ad tags
  • malicious ad networks
  • malicious redirectors
  • malicious ad servers

• Payloads
  • the actual malicious code that gets delivered
  • the actual malicious sites to which the client is ultimately redirected

• This paper: Measure and detect the *enabler* half of this picture.
  • Payload detection based on stock products
  • Google Safe-Browsing and Microsoft Forefront
Example Fake-AV Malvertisement Campaign

• Drive-by-download attack
  • victim browsers redirected to fake-AV site
  • fake-AV malware pretends to detect viruses and sells fake fixes

• Impact
  • infected at least 65 publisher pages in summer 2011
  • infected pages include top Alexa sites (e.g., freeonlinegames.com)

• Delivery included five levels of indirection:

Publisher: freeonlinegames → Syndicator: Google → Syndicator: DoubleClick → Syndicator: adsloader → Redirector: enginedelivery → Attack Site: eafive.com
Attacker Gambits

• Domain name impersonation
  • adsloader.com ≠ adloader.com

• Subversion of legitimate (often trusted) ad networks
  • GoogleServices, DoubleClick
  • over 24 ad networks total (!)

• Conditional redirection (cloaking)
  • adsloader.com redirects visitors at most once (per IP)
  • only IE agents redirected
  • empty referrers not redirected

• Honeynet evasion
  • enginedelivery withholds malicious content from Amazon EC2 IPs

• Conditional payload delivery
  • only IE6 received Fake-AV solicitation from eafive.com

• Domain and payload rotation
  • 16 different redirectors
  • 84 different fake-AV scanners
Measurement Study

• Crawl 90,000 web sites continuously for ~3 months (summer ’11)
• Infer redirection chains
  • HTML code (attributes containing URLs)
  • HTTP redirection (302-responses)
  • JavaScript net accesses (mine script texts for domain names of requests)
  • 24.8M chains and 21.9M URLs collected
• Identify malicious nodes
  • detection based on stock products (Google Safe Browsing, Micosoft Forefront)
  • Paths containing malicious nodes are malicious paths.
  • Descendants of malicious nodes might not be malicious!
Distinguishing Features of Malicious Nodes

• Node roles: known publisher / known ad-node / unknown
  • non-malicious paths: 93.1% known
  • malicious paths: 8.4% known
• URL patterns (Example: /showthread.php?t=12345678)
• Short domain name life expectancies
• Short, diverse associations w/publishers
Syndication and Redirection Cloaking

• Syndication Rates
  • 64% of all paths involve syndication (multiple ad networks)
  • 86 well-known networks compromised
  • 92% of DoubleClick-facilitated attacks are via syndicated paths

• Redirection cloaking
  • Malvertisement paths tend to be longer due to redirection cloaking
  • Early malicious redirectors tend to be involved in many attacks
From Measurement to Detection

• Goal: Use path statistics to reliably detect malvertisements

• Major finding:
  • Blindly applying heuristics to full redirection paths doesn’t work well.
    • too slow, difficult to implement
    • too many false positives
  • But heuristically identifying short, suspicious path segments works very well.
    • faster, easier to implement
    • malicious nodes tend to be clustered along the path
    • node roles in the segments are key
MadTracer Architecture
# MadTracer Detection Results

<table>
<thead>
<tr>
<th>Category</th>
<th>#MadTracer</th>
<th>#S&amp;F</th>
<th>#FP</th>
<th>#S&amp;F-MadTracer</th>
<th>#MadTracer-S&amp;F</th>
<th>FD(%)</th>
<th>New findings (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>scam pages</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>0.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>drive-by-download pages</td>
<td>216</td>
<td>104</td>
<td>20</td>
<td>8</td>
<td>120</td>
<td>9.26%</td>
<td>51.85%</td>
</tr>
<tr>
<td>click-fraud pages</td>
<td>89</td>
<td>7</td>
<td>13</td>
<td>1</td>
<td>83</td>
<td>14.61%</td>
<td>92.13%</td>
</tr>
<tr>
<td>all pages</td>
<td>291</td>
<td>111</td>
<td>32</td>
<td>9</td>
<td>189</td>
<td>11.09%</td>
<td>61.86%</td>
</tr>
<tr>
<td>scam domain-paths</td>
<td>23</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>23</td>
<td>0.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>drive-by-download domain-paths</td>
<td>627</td>
<td>216</td>
<td>87</td>
<td>20</td>
<td>431</td>
<td>13.88%</td>
<td>65.55%</td>
</tr>
<tr>
<td>click-fraud domain-paths</td>
<td>3422</td>
<td>42</td>
<td>125</td>
<td>26</td>
<td>3406</td>
<td>3.65%</td>
<td>98.77%</td>
</tr>
<tr>
<td>all domain-paths</td>
<td>4072</td>
<td>258</td>
<td>212</td>
<td>46</td>
<td>3860</td>
<td>5.21%</td>
<td>93.66%</td>
</tr>
</tbody>
</table>
Conclusions

• Malvertising is a significant threat to the internet revenue model
  • much of the internet funded by advertising (billion-dollar industry)
  • at least 1% of top sites fell victim to malvertising campaigns in 2011

• Simple detection approaches don’t work
  • IP black-listing fails because malicious campaigns rotate servers too quickly.
  • Honeypotting is frustrated by highly selective attacks.
  • Full referrer paths of many legitimate ads display “suspicious” characteristics (long path lengths, unknown nodes, short domain lifetimes, etc.). This can result in high false positive rates.

• But detecting short, malicious sub-paths works well
  • Malicious nodes operate in close proximity on a malicious path.
  • Possible to identify node roles in these sub-paths.

• Open problem: It’s still an arms race.
  • As these heuristics catch on, malvertisers will adopt new topologies to counter them.
  • The race will continue as defenders compensate with new heuristics.
Discussion Questions

• Is there a principled answer to the malvertising problem?
  • language-based security?
  • formal methods?
  • browser security?
  • script analysis?

• What about economic/financial solutions?
  • better revenue models?
  • incentive schemes?