Anti-Adblock

Public Summary

A large proportion of internet “content” sites (media, search, webmail, etc) rely on advertising revenue for support. Frequently this advertising is performed via click-through images and links. However, client-side filtering of such material is becoming increasingly popular – the fastest growing share of the browser market is now “Firefox” (over 23M downloads), for which the second most popular extension is a client side ad-filter (with over 1M users). Such filtering has been available for several years, but is only now becoming “mass market”, and thus having a direct impact on the global US$10Bn/yr internet advertising revenue that allows many content sites to survive.

Anti-AdBlock provides a robust solution to this problem, involving a relatively straight-forward modification to web-server software. The most common web-server in the market is Apache, which is open source and allows for production of derivative works.

Anti-AdBlock is seeking funds to obtain a US Software patent to protect intellectual property before licensing modified web-server software “know-how” to internet media and advertising companies.

Confidential Material

The basis of “client side filtering” is that most web servers provide advertising material either off-site, or in specific site-wide directories. The raw web page source code (HTML / XML) can be inspected “on the fly” to look for such occurrences – for example, with a small number of filters such as “*/sponsor/*, */adv/*, */banner/*” etc it is possible to block the vast majority of ads on the internet. By including specific well-known domains of advertisers (e.g. *doubleclick*) even greater blocking can be achieved.

Although the common client-side filters do not implement it currently, it is clear that by utilising a periodically updated “filter database” to which the entire community of users contributes, this type of filtering will become very powerful. This would represent a considerable income-risk to the major internet advertisers, as well as smaller operations.

The Anti-AdBlock solution to this problem is obfuscation of directories (in the first instance) and filenames (if it becomes necessary) within automatically generated web page source code. This obfuscation would be performed on-the-fly, and would include a high proportion of random data, ensuring that client side filtering is rendered practically useless. Such obfuscation can be performed in a manner that is transparent to both the web developer and the client browser, and utilises minor computational and storage loads on the web-server.

Technical Aspects

The essence of obfuscation is the replacement of a "plaintext" name with an "obfuscated" name. For example, if the server-side source code contains a directory called "/banner/" this could be replaced by "/dkghasr3/" as the web page is being sent to the client. Furthermore, this substitution could be highly random. When the client browser comes to load the content in "/dkghasr3/", a request is simply passed to the web-server in the normal manner. The web-server then translates this back to the required "true" directory on the server itself, and provides the requested content.

The case of internet addresses is more difficult, as an internet address must be globally addressable. This means that randomisation of the address cannot work beyond certain limits, such as scrambling part of an IP address range or randomising the xyz part of xyz.doubleclick.com, both of which can easily be included in a client-side filter. In these cases it may be more appropriate to utilise the HTTP redirect commands, where the client requests some content inside an obfuscated directory, which then translates to an automatic redirect (HTTP 30x commands) to the required off-site content. These automatic redirects are performed within the lowermost "layer" of a web-browser, well before any client-side filtering of web page source.

The crucial issue here is speed of translation between "plaintext" and "obfuscated" directory names (and/or filenames). It is important that this process does not introduce an undue load on the web server, otherwise there is a significant hidden cost in using these methods.

There are essentially 3 simple schemes that may be used, of variable complexity:

1. A simple substitution of characters could be used – all instances of "b" could be changed to "g" and so on. However for this to be effective, the substitutions would have to change over time, and users with "bookmarks" of specific content would find that they could no longer link.
2. Truly random directory names could be allocated. A large look-up table is then maintained to translate between "plaintext" and "obfuscated" names. The content in this table could be updated on a regular basis, such that "learning" filters would be thwarted. However, this requires a large storage element (ram), and searching based on strings (which is slow). This solution also suffers from a user "bookmark" failing if the entire table is not remembered for all time, which would cause a very large storage issue.
3. The generated name could be pseudo-random, containing a random element and a meaningful element. For example, the letters a-z and digits 0-9 provide for 36 characters that are legal for all file systems. By using 32 of these, a 5-bit number can be represented. If 12 characters are used for every filename or directory name, this is 60 bits of data. If 32 bits are used for a static hash look-up table, there is less than a 1 in 10^9 chance of duplicating a directory name, which is likely to be an acceptable risk. This leaves 28 bits of random data that can be mixed in. The hash look-up table can be constructed such that only the used portions of the table reside in memory, and this can be searched rapidly, using standard hashing and look-up methods. The extraction of informational and random bits can be performed in a number of ways, a very insecure (but, for this application, sufficiently effective) method would be to simply use a random bit for every second bit of the data, except for the last 4.

Because the hashing of the directory or filenames can be performed from the known content of the server, and the means of extracting informational and random bits is known (to the server) in advance, the hash table can be re-created as and when required. This is in contrast to the second option, in which the entire table must be maintained forever in order to allow book-marks to function. Additionally, any directory or filenames not found in the hash-table could be passed through to the rest of the web-server unaltered, thus providing seamless backwards compatibility with old (pre-Anti-AdBlock) bookmarks from the site.

It remains up to implementers as to the extent of obfuscation, as some may wish to have “real” content in plaintext directories, and advertising materials simply obfuscated for Anti-AdBlocking purposes. This may lead to a new generation of client side filters that remove pseudo-random content, which would force a move to fully-obfuscated directories and filenames.

The key issue is that obfuscation on-the-fly, including some portion of time-varying randomness, ensures that client side filtering is far less effective. Such obfuscation will yield a relatively low computational and storage load, and will be entirely transparent both to the web developer and the client browser.

Financial Aspects

Anti-AdBlock seeks approximately £10,000 of seed capital in order to file a US Software patent for this process, aiming to license to major internet advertising agencies such as doubleclick.com, google.com. It is anticipated that the protection of revenue offered by this relatively modest software change will be of considerable commercial value to these companies.

The Interactive Advertising Bureau (IAB) and PricewaterhouseCoopers (PwC) announced that Internet advertising totalled nearly US$2.7 billion in the fourth quarter of 2004, with full-year estimates at US$9.6 billion, and growth at between 20%-30% per year for the last 2 years. Banner advertisements (the primary target of client-side filtering) represented over 20% of revenue in 2003 (the last year for which full figures are available). This does not include "google ads" and similar methods, which are also blocked by client side filters, and rapidly gaining importance.

Given that FireFox is the most rapidly growing segment of the browser market (now over 25M downloads, and 8.5% of the browser “market”), and the second most popular extension is a client-side filter with over 1M users, this implies that at least 8.5% / 25 = 0.34% of users are currently running client-side filters. This is a highly conservative estimate, as many other client-side filters are available, and this technology is being developed into extremely popular Internet Explorer toolbars such as the Yahoo and Google bars.

Assuming that only 1% of users presently utilise client-side filtering to block banners, this represents a potential global loss of US$20M per year in revenue, which is clearly orders of magnitude greater than the costs of intellectual property protection for Anti-AdBlock. Furthermore, as client-side filtering technology becomes more rapidly integrated, the value of Anti-AdBlock will increase dramatically.

The top 10 internet advertising companies account for approximately 70% of advertising revenue (IAB & PwC Internet Advertising Revenue Report), indicating

that much of the value of Anti-AdBlock can be targeted at a small number of companies. It is therefore anticipated that negotiations with the relevant industry players could commence rapidly after filing for intellectual property protection.

Inventor

Shamus Husheer is a 3rd year PhD Student in chemistry, developing instrumentation for synchrotron radiation research and x-ray crystallography. His aptitudes are developing instrumentation for research, including mechanical and electronic engineering, firmware and software development, and numerical modelling.

Academic History

2002 – Present	PhD Research, University of Cambridge, UK.
	Topic : X-ray crystallography of photo-excited materials
Funding	NZ Government, Cambridge University, and Daresbury Synchrotron
1999 – 2001	MSc Research, University of Otago, New Zealand.
	Topic : pH Measurement in oceanography for global warming research
Funding	Leubecki Postgraduate Scholarship in applied sciences
1998 – 1999	Research project at Australian National University on powder diffraction
1995 – 1998	BSc(Tech.), Waikato University, New Zealand.
	Specialising in industrial chemistry

In his spare time, Shamus shoots for the University of Cambridge small-bore rifle team.