Ad Blocking Survival Guide

Tactics and Strategies for Web Publishers

Ad Blocking Survival Guide

Tactics and Strategies for Web Publishers

The Ad Blocking Survival Guide – First Edition

Rev 1.0

Contents

Chapter 1

1. Introduction

A recent report compiled by Adobe and PageFair suggests that almost 200 million people now use an ad blocker when browsing the internet, a 41% increase over the preceding 12 month period. This widespread use of ad blocking technology is expected to result in over $20 billion in lost advertising revenue in 2016 alone.

Despite the magnitude of these numbers, the threat of ad blocking is largely an invisible threat to the average web publisher. Unless steps are taken to assess the impact of ad blocking on an individual website, the only sign that ad blocking is an issue is likely to be a decline in advertising revenue. Given that ad revenue tends to fluctuate for a variety of reasons, even this metric can be misleading.

Ad blocking is certainly far from the only challenge faced by web publishers today. The good news, however, is that ad blocking is one of the few areas where web publishers have some control over how to respond to the issue. We can't cure ad blindness, stop ad budgets migrating to Facebook, and whatever is causing revenues from Google AdSense advertising to decline is far beyond our control. What we do control is our own websites and, by extension, how we react to, communicate with and respond to visitors using ad blocking is entirely up to us.

While there is no "one size fits all" solution, the goal of this book is to outline a range of proven strategies designed not only to detect, quantify and mitigate the threat of ad blocking, but also to move beyond advertising as a sole source of revenue.

The chapters in this book cover a variety of options ranging from passive measuring of ad blocking behavior to the more aggressive step of denying access to those visitors using an ad blocker. While no particular strategy is recommended over another, the inherent risks of a particular option are outlined where necessary so that an informed decision can be made about whether the strategy is right for your situation.

1.1 Taking a Measured Approach

To a large extent, the issue of ad blocking has evolved into a technological arms race that has already spawned counter-measures with names such as “anti-ad blocker”, “anti anti-ad blocker” and “anti-ad blocker killer”.

When considering the strategies in this book, it is important to keep in mind that users have installed ad blockers because they do not want distracting and invasive ads on the sites that they like to visit. As web publishers we have a responsibility to respect this point of view and avoid further fueling the arms race by rushing to adopt the most extreme ad blocking counter-measures. Statistics suggest, for example, that many website visitors will willingly whitelist a website if reasonably requested to do so as long as the content is valuable to them and the ad experience does not impose too great a burden.

While there is no certainty that we will ever reach the point where everyone is blocking ads, we also have an obligation to avoid driving the next 200 million people to install an ad blocker. As part of the process of handling ad blocking, take time to reevaluate the quantity, quality and placement of advertisements on your site with a view to improving the overall visitor experience.

As we hope you will come to appreciate as you read this book, ad blocking survival is a rich and diverse discipline that goes far beyond simply finding ways to make visitors view the same type of ads they have already indicated they do not wish to see.

1.2 Feedback

We want you to be satisfied with your purchase of this book. If you find any errors in the book, or have any comments, questions or concerns please contact us at *feedback@ebookfrenzy.com*.

1.3 Errata

While we make every effort to ensure the accuracy of the content of this book. Any known issues with the book will be outlined, together with solutions, at the following URL:

http://www.ebookfrenzy.com/errata/adblockingsurvival.html

In the event that you find an error not listed in the errata, please let us know by emailing our technical support team at *feedback@ebookfrenzy.com*. They are there to help you and will work to resolve any problems you may encounter.

Chapter 2

2. The State of Online Advertising

As with just about any market driven economy, the online advertising marketplace is based on the concept of supply and demand. As web publishers, however, we sometimes subconsciously invert the supply and demand equation within the context of online advertising. It is all too easy to think of our websites as representing the demand side, requiring ads from our suppliers (in the form of Google AdSense, ad networks and direct advertisers). The reality, of course, is that the advertising space on our websites is actually the supply side (it is, after all, called "inventory" for a reason), with demand taking the form of advertisers looking for places to run their ads.

Many of the challenges facing advertising funded web publishers today are symptoms of a longstanding imbalance in this supply and demand equation. For quite some time now there has been an unhealthy imbalance between supply and demand, the effects of which have contributed to a backlash from internet users in the form of the increased deployment of ad blocking solutions.

As supply outstripped demand, CPM rates dropped. As web publishers made less money per ad impression many websites tried to compensate by displaying more ads per page. Another symptom of this imbalance is the inability of most ad networks to fill all the inventory on a website. This led to the widespread chaining of ad networks where a request to fill an ad space is passed through multiple ad networks until either an ad is available or the end of the chain is reached. Not only did this result in degraded website performance, but the further down the chain the request travelled, the lower the quality of the ad often became.

Further, the trillions of impressions of low CPM ads designed to build brand awareness that were never intended to be clicked have desensitized visitors to the more valuable CPC ads that were, leading to so-called "ad blindness", declining click-through rates and additional pressure to compensate by running more ads per page.

As people began to see re-targeting advertisements for products they had researched on retail websites follow them around the internet, privacy in online advertising has become an additional area of concern (albeit one based on a misunderstanding of how ad retargeting works).

The reasons for the popularity of ad blocking aside, the facts are not in dispute. Websites that track usage generally report that between 20% – 30% of visitors are using an ad blocker. For websites that rely solely on advertising, the percentage of lost revenue typically falls into a similar percentage range.

The web publishing industry is far from doomed, however. As outlined in the remainder of this book, there are many strategies and technologies available for addressing the ad blocking issue. While some web publishers may fall by the wayside, the future will reward those publishers that continue to produce unique and quality content, take informed, intelligent and measured steps to address ad blocking and think beyond advertising as the only way to turn website traffic into revenue.

2.1 What's the DEAL?

The digital advertising industry is represented by the Interactive Advertising Bureau (IAB). The IAB is comprised of members that include most of the major media, publishing and technology companies. A key responsibility of the IAB is defining the technical standards for online advertising. The various standard ad sizes in use today, for example, were originally defined by the IAB.

Unsurprisingly, the IAB has also taken a keen interest in the issue of ad blocking and has devised a recommended approach that is based on the acronym "DEAL":

- **D**etect ad blocking, in order to initiate a conversation.
- **E**xplain the value exchange that advertising enables.
- **A**sk for changed behavior in order to maintain an equitable exchange.
- **L**ift restrictions or Limit access in response to consumer choice.

This approach advocates a measured approach when addressing the issue of ad blocking. The key goals are focused on educating users about the importance of advertising in supporting web content and seeking a change in behavior by users of ad blocking technology. Only as a last step does the IAB suggest imposing restrictions on access for those users continuing to use ad blocking.

In suggesting this approach, the IAB appears to be attempting to avoid the escalating technology arms race between ad blocker developers and the advertising and publishing industry. It is better, the IAB contends, to give users a reason to willingly turn off ad blocking than to develop ways to defeat ad blocking technology (which will, in turn, result in the development of more advanced ad blocking solutions).

2.2 **Making Advertising LEAN**

With all the discussion surrounding ad blocking it is easy to lose sight of the fact that people use ad blocking because they feel that advertising has degraded the overall user experience of the internet. If the visitors to a website are viewed as customers, then the old adage that "the customer is always right" suggests that something has gone wrong with the way that online advertising has evolved over the years.

Many people believe that online advertising slows down web page loading, drains phone batteries, exhausts mobile data plans, is overly intrusive, delivers malware and invades their privacy. Some of these perceptions are, of course, based on misunderstandings about how the underlying technology works, but most concerns are valid.

In response to these concerns, the IAB has created a new set of advertising standards referenced by the "LEAN" acronym:

- **L**ight. Limited file size with strict data call guidelines.
- **E**ncrypted. Assure user security with https/SSL compliant ads.
- **A**d Choices Support. All ads should support DAA's consumer privacy programs.
- **N**on-invasive/Non-disruptive. Ads that supplement the user experience and don't disrupt it.

The hope is that the standards for LEAN advertising will be both widely adopted and result in ads that are acceptable to internet users. The widespread deployment of LEAN advertising probably won't be enough to make people uninstall their ad blocker, but it may just prevent others from installing one.

2.3 **Summary**

Online advertising, like any other marketplace, is based on supply and demand. A seemingly infinite supply of ad space has resulted in a decline in revenue for websites that rely on advertising income. As websites added more ads to compensate for declining revenue an unforeseen side effect was the rise in popularity of ad blockers. The IAB, an industry group representing advertisers and media companies recommends a specific approach to dealing with ad blocking, encapsulated by the DEAL acronym. The IAB is also creating new standards for online advertising that advocate lighter weight, secure and non-invasive ads.

Chapter 3

3. An Overview of Ad Blocking Technology

Before exploring strategies to address ad blocking, it is first important to gain a basic understanding of the way in which ad blocking works. This chapter will provide an overview of the different forms of ad blocker and outline the key ways in which most ad blocking technology works.

3.1 Different Types of Ad Blocker

The most common and widely used type of ad blocker takes the form of a web browser extension or plugin, invariably available for installation free of charge with just a few clicks. Custom web browsers are also available with built-in ad blocking capabilities.

Ad blocking within enterprise networks is often implemented using software or hardware devices that are installed within or alongside gateway internet routers.

In addition, some mobile network and internet service providers are considering implementing ad blocking at the network level, thereby stripping out ads from content before it is delivered to customer devices.

3.2 How Ad Blocking Works

Ad blocking typically employs two techniques to eliminate advertising from a web page.

3.2.1 Hiding Advertising Elements

A web page is actually a document that describes how the page is to be rendered within the web browser window. It contains the text, formatting directives, scripts and images that make up the content and behavior of the page that is to be presented to the user.

Browser-based ad blocking is centered around the Document Object Model (DOM). The DOM provides a structured representation of the content of the web page together with a programming interface allowing the content to be accessed and changed. It is the DOM, for

example, that enables the interactive nature of web pages to be implemented, such as hiding and showing sections of a web page based on user selections, or dynamically changing the style or color of different page elements.

Either before or after the web page loads (depending on the browser type), ad blockers scan the web page document using the DOM to identify any content elements that contain advertising. When ad related content is identified, the DOM programming interface is used to hide those elements from view.

The diagram shown in Figure 3-1 illustrates ad elements hidden by an ad blocker in the form of a web browser extension:

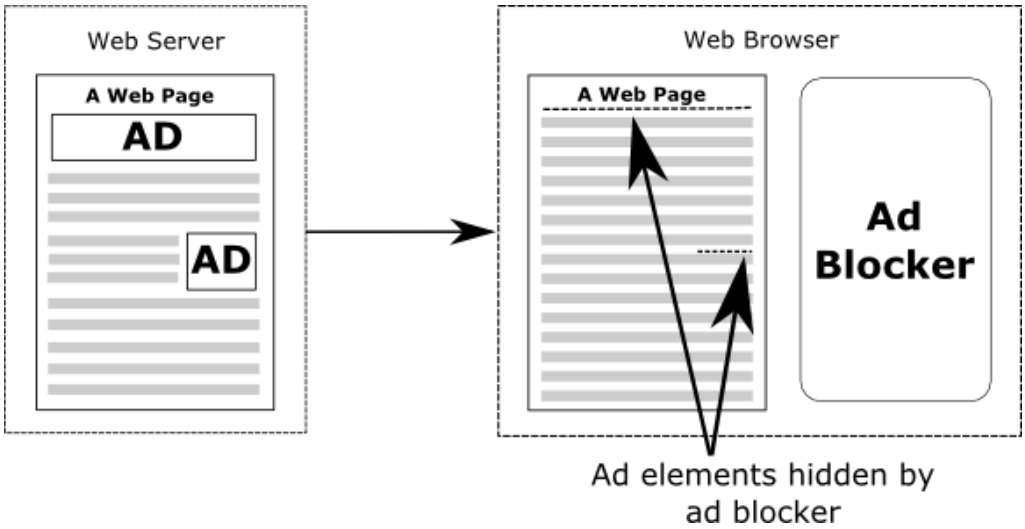

Figure 3-1

3.2.2 Blocking Access to Ad Servers

When a web page containing ads loads into the browser, it contains JavaScript code designed to contact the ad servers that provide the ad content to be displayed. The second technique used by ad blockers involves preventing the browser from contacting those ad servers, thereby preventing the ad from being downloaded and displayed to the user.

Though the exact techniques differ from one web browser type to another, all browsers provide some mechanism for blocking access to external resources such as ad servers. Ad blockers simply use these mechanisms to ensure that any attempt to communicate with a known ad server is blocked (Figure 3-2).

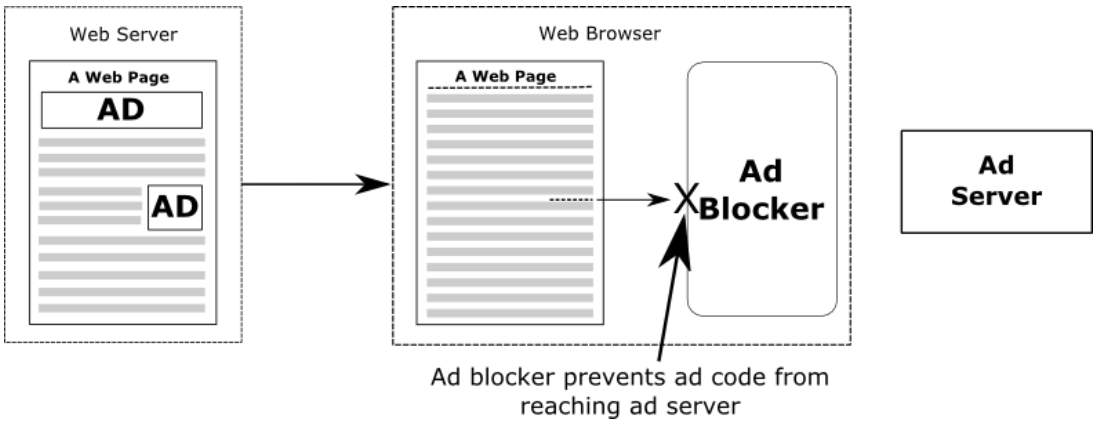

Figure 3-2

The providers of network level ad blocking offer less detail about how their technology works, though it is likely that a combination of hindering ad server communication and on-the-fly modification of web page content prior to delivery of the page to the user's device are employed to provide similar results.

To see this technique in action, install an ad blocker extension into your web browser of choice and enable it. Display the JavaScript console for the browser using the appropriate steps from the following table:

Browser	Steps to Display Console
Google Chrome	Ctrl-Shift-J
Firefox	Ctrl-Shift-J
Internet Explorer	F12
Microsoft Edge	F12
Safari	Select Safari -> Preference menu. Click on the Advanced Tab. Enable "Show Develop menu in menu bar". Press Option-Command-C.

Once the JavaScript console is visible, navigate to a web page containing ads and note the output that appears in the JavaScript console. Among other errors, each blocked attempt to connect with an ad server will be reported within the console. Figure 3-3, for example, shows a list of blocked ad server connections within the JavaScript console of the Chrome web browser:

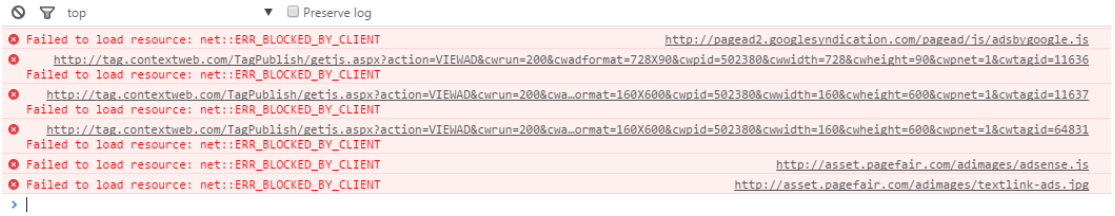

Figure 3-3

3.3 **Ad Blocker Lists**

Having covered how ad blockers hide ads and block ad server communication, the question that remains as to how they know what to block and what to leave alone. This is achieved primarily through the use of filter lists (also referred to as blacklists) which contain sequences of patterns designed to match all of the ad server addresses and web page element names that are known to contain ads. The most widely used filter list is EasyList which is used by the AdBlock Plus ad blocker. The content of this file can be reviewed using the following URL:

https://easylist-downloads.adblockplus.org/easylist.txt

The list contains, for example, the following entry:

```
||contextweb.com^$third-party
```

The web domain contained in the above filter references an ad server used by an online advertising network and inclusion in the list ensures that any attempt by a web page to access this server will be blocked. Similarly, the list contains entries such as the following:

```
##.advert
##.banner_ad
```

These entries dictate that any elements within the web page with class names matching either “advert” or “banner_ad” should be hidden from view. The following element, for example, would be hidden by the ad blocker if encountered in the DOM of a web page:

```
<div class="advert">
    // Ad code here
</div>
```

In addition to the lists containing filters for ad elements and ad servers, privacy lists are also used to block tracking scripts (such as those used by Google Analytics).

In general these lists are user generated in so far as suggestions for additions to the list are submitted by users of the ad blocking tools. If an ad blocker user visits a page where ads are still visible, the user may pass information about the ad element and ad server to the people who maintain the filter list for that particular ad blocker. One of these maintainers will, subject to reviewing the suggestion, create a filter to block the ad and append it to the list.

3.4 **Summary**

Before addressing the issue of ad blocking it is helpful to first gain an understanding of how ad blocking technology usually works. Ad blockers take a variety of forms including web browser extensions, software solutions and hardware devices.

Ad blocking typically uses two techniques to eliminate advertisements. The first involves the identification of ad elements within a web page and, using the Document Object Model (DOM), hiding those elements from view. The second part of the process involves blocking any attempt by the ad code within a web page to connect to the ad server to obtain the ad content.

Many ad blockers (particularly those provided in the form of a web browser extension) access filter lists to identify the content elements to hide and ad server addresses to block.

Chapter 4

4. Basic Ad Blocker Detection

Just about every strategy for dealing with ad blocking begins with identifying when it is happening. This involves adding to the pages of a website some JavaScript code designed to detect when changes have been made to the page content by an ad blocker.

This chapter will outline the ways in which ad blocker detection typically works before introducing two options for implementing this detection within the pages of a website.

4.1 How Ad Blocker Detection Works

As described in detail in the previous chapter, ad blockers work by blocking communication with the servers providing the ads, and by hiding the elements of web pages that contain advertising content. Ad blockers obtain information about what to block by referring to filter lists containing the addresses of all known ad servers and vast sequences of pattern matching rules.

Ad blocking detection works by placing so-called "bait content" within the pages of a website. Although invisible to the user (it might, for example, be a single pixel in size), this bait content is implemented so as to appear to ad blockers as advertising content. After the web page has finished loading, the detection code checks the properties of the bait content to ascertain whether it is still visible. If the bait content has been hidden, the detection code knows that an ad blocker is active.

4.2 An Example of Bait Content

The following web page element will, if encountered by an ad blocker, be treated as advertising content and hidden from view:

```
<div class="banner_ad"> </div>
```

In terms of the content displayed by the above element, all that this consists of is a single space character. What draws the attention of most ad blockers is the class name of "banner_ad" which can be found within most ad blocker blacklists. This can be verified by opening a browser window and loading the EasyList filter list used by a number of ad blocking extensions, the URL for which is as follows:

https://easylist-downloads.adblockplus.org/easylist.txt

Within the list, a search for *banner_ad* should list number of matches, more than one of which will be a perfect match for the bait content outlined above.

4.3 **Detecting the Ad Blocker**

The most basic of ad blocking detector code simply accesses the bait element and performs a range of tests to identify if the content is still visible. The following JavaScript code, for example, obtains a reference to the banner ad element before checking the height and width properties. If these are set to zero the function makes the reasonable assumption that an ad blocker has hidden the content:

```
<script src=
 "https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js">
</script>
<script>
(function() {

        var detector = function() {
            setTimeout(function() {

                if(!document.getElementsByClassName) return;
                var ads =
                      document.getElementsByClassName('banner_ad'),
                    ad  = ads[ads.length - 1];

                if(!ad || ad.innerHTML.length == 0
                              || ad.clientHeight === 0) {
                    console.log('Ad Blocker Detected');
                } else {
                    console.log('No Ad Blocker');
                }
            }, 2000);
        }

        /* Add a page load listener */
        if(window.addEventListener) {
            window.addEventListener('load', detector, false);
        }
})();
</script>
```

Note that the above code listing includes the following lines of JavaScript:

```
if(window.addEventListener) {
     window.addEventListener('load', detector, false);
}
```

These lines of code configure the browser to call the “detector” function when the web page has finished loading. A closer inspection of the lines of JavaScript code within the detector function will reveal that the JavaScript *setTimeout()* function is used to ensure that the test of the banner ad properties takes place after a 2000 millisecond timeout. The purpose of this delay is to allow the ad blocker to complete its work before checking the status of the banner ad.

As currently implemented, the code simply outputs a message to the browser’s JavaScript console to indicate whether an ad blocker has been detected.

4.4 Testing the Detector

If you have access to the server on which your website is running, create a sample html file to test out the above detection code. For example:

```
<html>
<head>
<title>Detector Test Page</title>
</head>
<body>
<h1>Testing the Detector</h1>

<div class="banner_ad"> </div>
<script src=
"https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js">
</script>
<script>
(function() {

        var detector = function() {
            setTimeout(function() {

                if(!document.getElementsByClassName) return;
                var ads =
                        document.getElementsByClassName('banner_ad'),
                    ad  = ads[ads.length - 1];
```

```
                if(!ad || ad.innerHTML.length == 0
                                || ad.clientHeight === 0) {
                    console.log('Ad Blocker Detected');
                } else {
                    console.log('No Ad Blocker');
                }

            }, 2000);
        }

        /* Add a page load listener */
        if(window.addEventListener) {
            window.addEventListener('load', detector, false);
        }
})();
</script>
</body>
</html>
```

Alternatively, if you are using a WordPress installation select the *Appearance* option from the WordPress dashboard followed by the *Editor* entry as illustrated in Figure 4-1 below (note that these options are not available for sites hosted on WordPress.com):

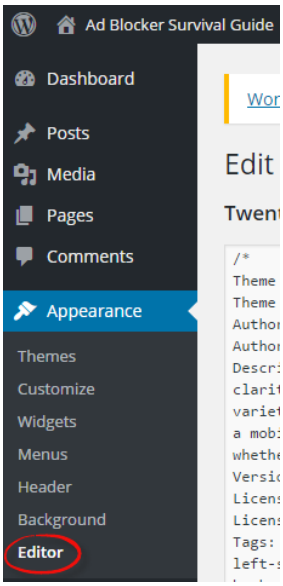

Figure 4-1

From the drop down menu in the upper right hand corner of the main panel, select the theme you are using for your website and click on the Select button:

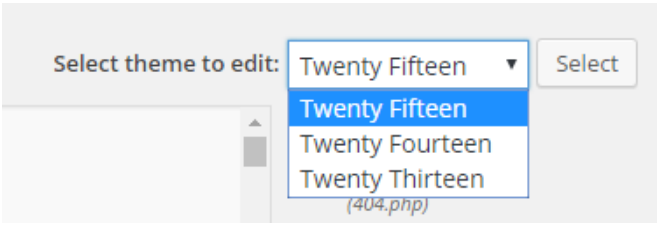

Figure 4-2

Next, select the Footer (footer.php) entry from the *Templates* list located on the right hand side of the page as shown in Figure 4-3:

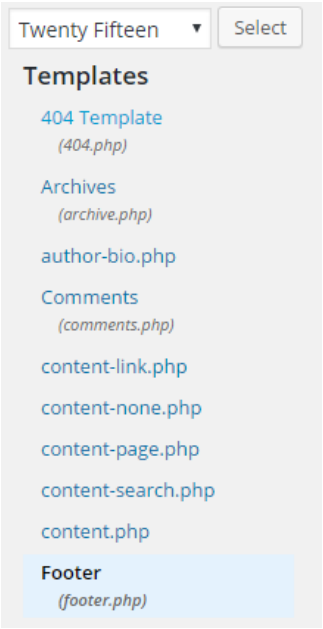

Figure 4-3

Within the code for the template, scroll down until the </body> tag comes into view and place the bait ad and detection JavaScript code immediately above it:

Figure 4-4

Once the code has been added, click on the *Update File* button. If the button is not visible or WordPress indicates the file cannot be saved, follow the steps on the following page to add write permission to the *footer.php* file:

https://codex.wordpress.org/Changing_File_Permissions

The path to the file for which permissions need to be changed will be as follows (where *<theme name>* is replaced by the name of the theme you are using on your site):

```
wp-content/themes/<theme name>/footer.php
```

If you are using another content management system such as Joomla! or Drupal, refer to the documentation for steps on embedding JavaScript code into the web pages of your site.

Once the detector code has been added, open a browser in which an ad blocker extension is installed and display the JavaScript console using the steps outlined in the chapter entitled *An Overview of Ad Blocking Technology*.

With the console displayed, load a web page containing the detector code while the ad blocker extension is disabled and verify that the “No Ad Blocker” message appears (keeping in mind that there will be a 2000 millisecond delay before the message appears). Enable the ad blocker, reload the page and check the console for the appearance of the “Ad Blocker Detected” message.

See it in Action

http://www.techotopia.com/survival/detect.html

4.5 Using the BlockAdBlock Script

The above example provides a simple approach to detecting ad blockers and offers an insight into the way in which such detection typically works. A number of different detection scripts are also freely available for download from a variety of websites. Many of these are similar to the example outlined previously in this chapter. For a more robust and extensive detection option, the BlockAdBlock script is a well-regarded alternative that is widely used within the web publisher community. Details of this script and download instructions are available at the following URL:

https://github.com/sitexw/BlockAdBlock

To implement BlockAdBlock, download the *blockadblock.js* file from the above web page and install it onto your web server. Next, add the following to each web page for which detection is required:

```
<script src="/blockadblock.js"></script>

<script>
// Function called if AdBlock is not detected
function adBlockNotDetected() {
    console.log('Ad Blocker Detected');

}
// Function called if AdBlock is detected
function adBlockDetected() {
    console.log('No Ad Blocker');
}
```

```
// Recommended audit because AdBlock locks the file 'blockadblock.js'
// If the file is not called, the variable does not exist
'blockAdBlock'
// This means that AdBlock is present
if(typeof blockAdBlock === 'undefined') {
    adBlockDetected();
} else {
    blockAdBlock.onDetected(adBlockDetected);
    blockAdBlock.onNotDetected(adBlockNotDetected);
}
</script>
```

In addition to containing countermeasures to avoid itself being blocked by ad blockers, the BlockAdBlock script also removes the bait after detection and includes a range of configuration options not available with simpler scripts. Configuration options are available to output debug information to the JavaScript console, control whether the ad blocker detection check is performed automatically when the web page loads and configuration of the number of times the detection is performed and the duration between each attempt. The following code, for example, enables debugging output and disables detection at web page load time:

```
blockAdBlock.setOption({
    debug: true,
    checkOnLoad: false
});
```

If the checkOnLoad option is set to false, the detection can be triggered independently from anywhere within the web page via a call to the blockAdBlock.check() function as follows:

```
<script>
.
.

     blockAdBlock.check()
.
.
</script>
```

4.6 **Summary**

The first step in dealing with ad blocking involves identifying when it is taking place. This involves placing bait content (essentially a page element known to be identified as an advertisement by an ad blocker) and some JavaScript detection code. The web page is then configured to call the

detection code a short time after the web page has loaded. The detection code inspects properties of the bait content to verify if it is still visible. If the bait is hidden then the presence of an ad blocker is assumed.

Chapter 5

5. Assessing the Damage

The extent to which ad blocking impacts a website will vary from one site to another and is often dictated primarily by audience demographics. Current trends suggest that websites appealing to a younger or more technically oriented audience are likely to encounter a greater percentage of ad blocking activity than other types of website.

The first step in considering how to mitigate the effects of ad blocking, therefore, is to identify not only whether the problem exists, but that it does so to the extent that further action is justified. If only 10% of visitors to a website are using an ad blocker, for example, the website owner may reasonably make the judgement that the effort involved in adapting to the threat outweighs the benefits. To make this an informed decision, however, it is first necessary to gather and analyze website visitor behavior data.

The focus of this chapter, therefore, is to outline some steps that can be taken to obtain accurate data on how many visitors to a website have an ad blocker enabled. Two levels of ad blocker tracking will be introduced in this chapter. The first provides a simpler approach using a free service provided by PageFair (another free service, AdBlock X, will also be covered in the chapter entitled *An Overview of BlockAdBlock, AdSorcery and AdBlock X*). The second option makes use of Google Analytics and provides a greater level of detail and data collection options.

5.1 Tracking Ad Blocking with PageFair

For those in a hurry, the path of least resistance to tracking the percentage of ad blocker usage on a website is to sign up for a PageFair account. In addition to providing ad blocker tracking, PageFair also provides an option to generate revenue from some blocked ads, a topic that will be covered later in this book. For the purposes of this chapter, however, only the ad blocker measurement service will be used.

Visit *https://pagefair.com* and create a new account by entering your email address, password and website information. Once signed into PageFair, select the *Setup* tab to locate the ad block detection and measurement JavaScript code as illustrated in Figure 5-1:

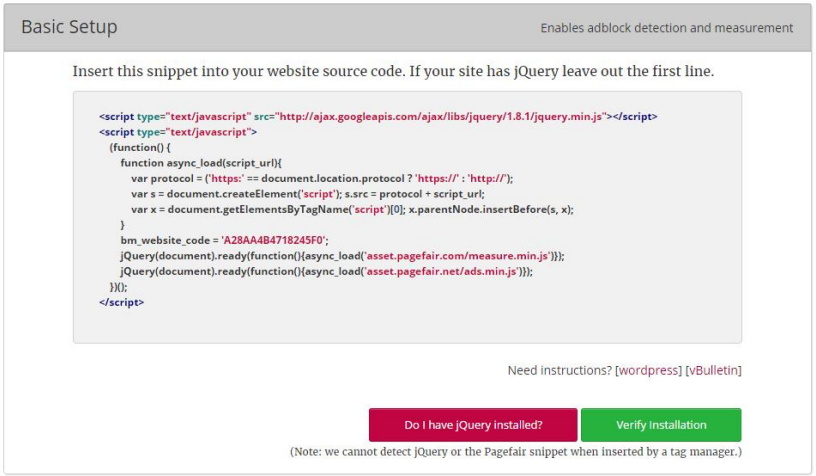

Figure 5-1

The PageFair JavaScript code makes use of the jQuery JavaScript library. If your website already imports this library it will not need to do so again. Before adding the measurement code to your website, click on the *Do I have jQuery installed?* button. If PageFair detects that jQuery is already installed, it is not necessary to include the first line of the code in your web pages.

For WordPress based websites the code can be added via the Theme Editor within the WordPress dashboard. Instructions to achieve this can be viewed by clicking on the WordPress *Need instructions?* link on the PageFair setup screen. For other sites, cut and paste the code into the appropriate location within your website markup so that it is placed between the header <head> and </head> tags and is included in every page for which tracking is required.

If you are using a WordPress installation, select the *Appearance* option from the WordPress dashboard followed by *Editor* as outlined in the previous chapter. From the drop down menu in the upper right hand corner of the main panel, select the theme you are using for your website and click on the Select button.

Next, select the Header (header.php) entry from the *Templates* list located on the right hand side of the page as shown in Figure 5-2:

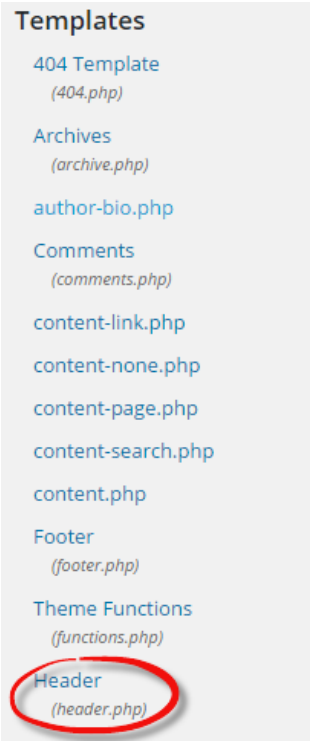

Figure 5-2

Once the code for the header template is displayed, paste the PageFair code into the template so that it is positioned immediately before the </head> tag:

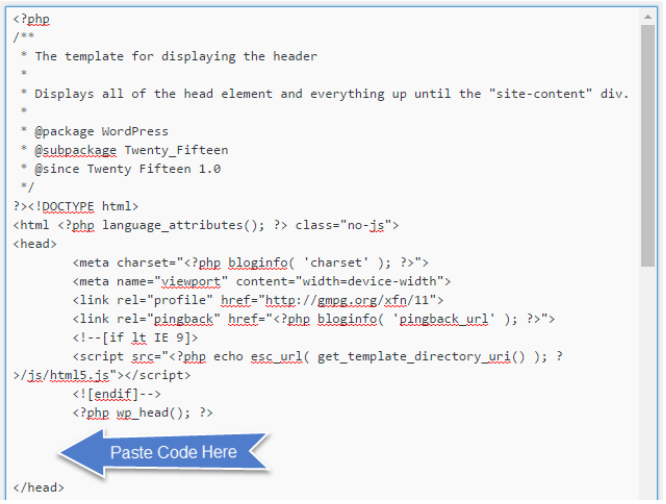

Figure 5-3

Click on the Update File button to save the changes, or follow the steps outlined in the previous chapter to add write permission to the following file (where *<theme name>* is replaced by the name of the WordPress theme you are using):

```
wp-content/themes/<theme name>/header.php
```

Once the code has been added to the website, click on the *Verify Installation* button to check that the code is correctly installed and functioning.

5.2 Reviewing the PageFair Results

It may take up to 24 hours for tracking results to become available within the PageFair dashboard. To view the results, select the *Reports -> Audience* option within the PageFair dashboard to display the data in graph form:

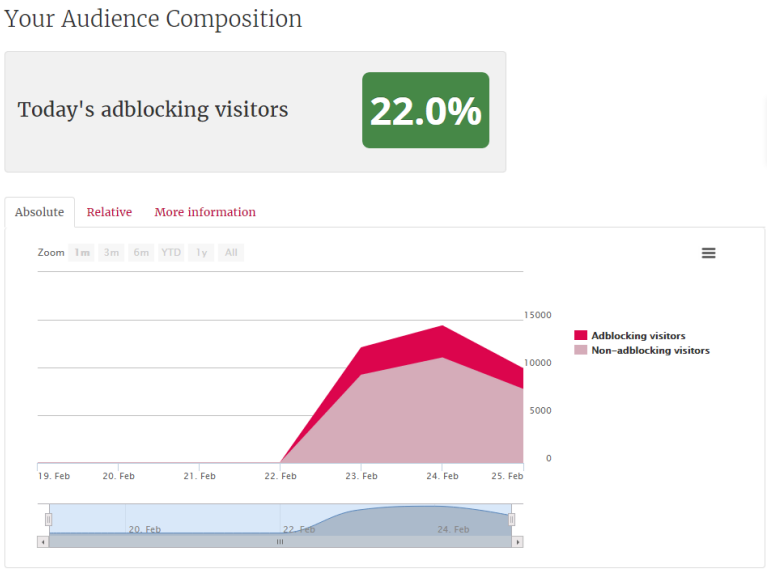

Figure 5-4

The above audience report displays the percentage of ad blocking activity on a per visitor basis. For traffic breakdown based on individual page views, select the *Reports -> Traffic* option.

5.3 Tracking Ad Blocking with Google Analytics

Measuring ad blocking using Google Analytics requires slightly more work than using PageFair but has the advantage of bringing the full power of Google Analytics to bear when measuring ad blocking behavior. Data available within Google Analytics includes the geographical location of

visitors, browser and device type in use and the operating system on which the browser is running. The fact that many web publishers already use Google Analytics also makes this an ideal choice for measuring ad blocking activity on a website.

The process of tracking ad blocking behavior within Google Analytics involves adding ad blocker detection code to each website page. This code is designed to trigger a different Google Analytics event depending on whether or not ad blocking is being used by visitors to the site.

5.4 Creating a Google Analytics Account

If you do not already have a Google Analytics account, create one now for free by going to *https://www.google.com/analytics/*, clicking on the "Sign In" link and selecting Google Analytics from the drop down menu. Enter the email address and password associated with your Google account if prompted to do so (or create a Google account if you do not already have one).

Having reached the Google Analytics sign up screen (Figure 5-5) click on the Sign Up button to begin the account creation process.

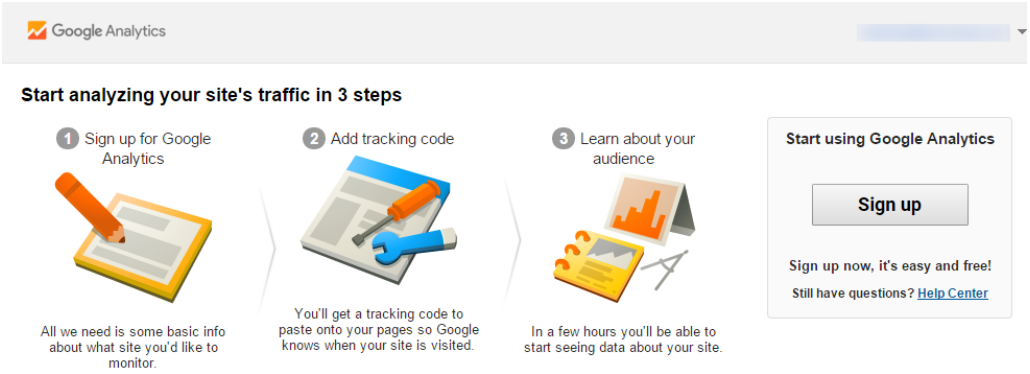

Figure 5-5

On the subsequent New Account form, select the "Website" option and enter the necessary information for your site to configure tracking. Having filled in the appropriate form fields, click on the *Get Tracking ID* button and add the code to your website so that it is included on every page that is to be tracked. If you are using WordPress, follow the steps above to place the code in the <head> section of the theme *header.php* template.

5.5 Detecting Ad Blocking

The next step in implementing ad blocker tracking is to add the detection code to the pages of the website. Once again, this code will need to be present in every page for which tracking is

required. For the purposes of this example, the JavaScript detection code outlined in the chapter entitled *Basic Ad Blocker Detection* will be repurposed to trigger Google Analytics events instead of sending output to the browser's JavaScript console. As outlined in the previous chapter, the sample detection JavaScript code reads as follows:

```
<div class="banner_ad"> </div>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.j
s"></script>
<script>
(function() {

        var detector = function() {
            setTimeout(function() {

                if(!document.getElementsByClassName) return;
                var ads =
                      document.getElementsByClassName('banner_ad'),
                    ad  = ads[ads.length - 1];

                if(!ad || ad.innerHTML.length == 0
                              || ad.clientHeight === 0) {
                    console.log('Ad Blocker Detected');
                } else {
                    console.log('No Ad Blocker');
                }

            }, 2000);
        }

        /* Add a page load listener */
        if(window.addEventListener) {
            window.addEventListener('load', detector, false);
        }
})();
</script>
```

As an initial test, embed the above code into your website so that it appears on every page that is to be tracked. Make sure to place this code so that it is positioned beneath the Google Analytics tracking code added earlier in this chapter.

Display the JavaScript console for your browser (as outlined in the chapter entitled *Basic Ad Blocker Detection*) and load different pages while enabling and disabling the ad blocker. Check the console output to verify that the code correctly detects the presence of an ad blocker by outputting “Ad Blocker Detected” and “No Ad Blocker” messages.

5.6 Triggering Google Analytics Events

Having verified that the code is correctly detecting the presence of an active ad blocker, the code now needs to be enhanced to trigger Google Analytics events. Edit the JavaScript detection code so that it reads as follows:

```
<div class="banner_ad"> </div>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.j
s"></script>
<script>
(function() {

        var detector = function() {
            setTimeout(function() {

                if(!document.getElementsByClassName) return;
                var ads =
                      document.getElementsByClassName('banner_ad'),
                    ad  = ads[ads.length - 1];

                if(!ad || ad.innerHTML.length == 0
                              || ad.clientHeight === 0) {
                    ga('send', {
                        'hitType': 'event',
                        'eventCategory': 'AdBlocker',
                        'eventAction': 'Blocked',
                        'nonInteraction': 1
                });

                    console.log('Ad Blocker Detected');
                } else {
                    ga('send', {
                        'hitType': 'event',
                        'eventCategory': 'AdBlocker',
                        'eventAction': 'Not Blocked',
                        'nonInteraction': 1
```