Title:
How To Block Direct Image Linking Using .htaccess

Word Count:
969

Summary:
Most of us have a specified limit to the amount of traffic our web servers will handle for us. That limit seems very generous - until you start looking at image downloads and the bandwidth required. A few dozen users downloading an image - that's one thing. But what if you have an image that dozens of other websites like?

Worst-case scenario: suppose you run a site that gets hold of a picture taken by an Iraqi soldier of an incident that gains a lot of media attention. And...

Keywords:
internet, SEO, Software development, webmaster

Article Body:
Most of us have a specified limit to the amount of traffic our web servers will handle for us. That limit seems very generous - until you start looking at image downloads and the bandwidth required. A few dozen users downloading an image - that's one thing. But what if you have an image that dozens of other websites like?

Worst-case scenario: suppose you run a site that gets hold of a picture taken by an Iraqi soldier of an incident that gains a lot of media attention. And suppose you have the exclusive rights to that image. You want to sell it, of course, not give it away, so you don't post it as a freebie. You may have a downloadable version that others can take away to post on their own website, generally with a link to your site. But your original image is reserved for your use only.

Here's the problem. Webmasters, often amateurs not really understanding why what they are doing is harmful, want to deliver the best possible image to their own readers. So instead of downloading your free file or linking to your site, they embed a link in their own page that downloads your picture, and only your picture, as part of their own website. This is easy to do; all you have to do is use the image link straight out to the other website.

With that excellent and lucrative image referenced above, you may have blogs on

both sides indexing you; you may even have news sites or image sites indexing you. How much bandwidth can your site take before it exceeds your monthly limit? Chances are, even a medium-case scenario is going to turn your site into a DNS instead, and you will lose viewers and site ranking while you are unavailable. Not worth posting that great picture anymore, is it?

But you do have another option: the .htaccess file. This works primarily in Apache servers; if you're not certain what you've got, call and ask. Because not all systems will allow them, and some can even be damaged by improper .htaccessing, you should contact your server administrator before you upload one, anyway.

What Is .htaccess?

.htaccess is a type of file that has, for years, been used to restrict access to protected web pages or areas, such as error pages and password-protected directories. You create it using a text editor such as NotePad or SimpleText, and then save as plain (ASCII) text. Upon saving it, the file extension needs to be changed from .txt to .htaccess – and the rest of the name needs to be left off. Your file should be named nothing except for .htaccess. Not file.htaccess. Just .htaccess.

If your text program insists on appending the .txt, you can right-click the file anywhere you normally open it and select Rename to remove the .txt. If your computer system does not show file extensions, look up how to make it show them! Alternately, telnet and ftp programs will also allow you to rename files and remove extensions.

Creating .htaccess Files

Your first step in creating this file should be to open your text editor and save an empty page as .htaccess. Turn off your word wrap function. .htaccess files are intended to be single-line commands, and a word wrap can throw this completely off and make your file unusable, either by breaking lines in the wrong place or by putting in unwanted characters when it's uploaded.

When you upload an .htaccess file, it should be encoded as ASCII, not binary. CHMOD the file to either 644 or (RW-R—R--) so that the server can use it but a browser cannot; readable .htaccess files can compromise your security by allowing hackers to figure out what you have protected and where the authentication files are. (You can also prevent this problem by putting your authentication files above the root directory so that they cannot be accessed via www.)

An .htaccess file affects the directory it's placed in as well as any subsequent subdirectories; if you have files everywhere you want protected, you can put the .htaccess file in your root directory. If you only want to protect your images, you can put it in your images directory alone. The closest .htaccess file to any given directory, reading up the directory tree, is the one that is applied to that specific directory. Try to use the fewest number of .htaccess files possible; redundancies can cause an infinite loop, which is bad for your site.

Preventing Hot Linking

The most important thing you want to do with an .htaccess file is prevent hot linking of non-HTML objects like images and movies. Hot linking is often referred to as "bandwidth stealing."

Your .htaccess file will disallow hotlinking; instead of the image your thief is trying to use, they'll get something else, like a broken image symbol or content you specify (angry men are popular).

Your .htaccess file content should read as follows:

```
RewriteEngine on
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^http://(www\.)?mydomain.com/.*$ [NC]
RewriteRule \.(gif|jpg)$ - [F]
```

The first two lines you don't have to modify in any way. However, the third line's http://mydomain.com needs to be modified to reflect your URL. The fourth line in the example is set up to deny use of GIFs and JPEGs; you can add, using a pipe (|) separator, any other file type you wish.

If your server is set up to deliver alternate content (call your server administrator and ask), you can add another piece to the fourth line of code in the .htaccess file to do this:

```
RewriteRule \.(gif|jpg)$ http://www.mydomain.com/angryman.gif [R,L]
```

This delivers an angry-man image you have in your directory; just as above, make certain you change that domain name to the proper one.