How to prevent content theft using Apache mod_rewrite or F5 iRules

Lori MacVittie, 2008-21-10

Over the years imaginative developers have come up with a number of ways through which they hope to stop the pilfering of their images. Whether due to copyright issues or the increased bandwidth and associated costs resulting from "hot linking", site owners have tried a variety of solutions from JavaScript that prevents the ability to right-click and "save as" to watermarking high-resolution versions to make their images less appealing to image thieves.

Regardless of the reason you may want to prevent image theft, there's an easier and more effective method than introducing easily countered JavaScript and costly alternative technology solutions.

HTTP REFERER

Every web request made via HTTP comes with a set of standard HTTP headers. One of those headers is the ***referer (***interesting spelled incorrectly), which indicates the domain and URL from which the request was made. If the request was direct (e.g. typed into the address bar by the user or loaded from a bookmark) then the referer header will be empty and usually displays in logs as "-" (at least they do on Apache). Otherwise, the referer header will have the FQDN (Fully Qualified Domain Name) of the referring page.

The ***referer header*** is central to this solution; by checking the ***referer header*** you can determine whether the request for your image came from a page on your site or someone else's or was a direct request. If you're trying to prevent theft obviously you only want to allow access to images if the request came from a page hosted on your site. So the referrer must contain your unique domain name (or any domain name you wish to allow access to) or the request should be denied.

Once you've determined that the referrer is ***not*** allowed access to the image, you'll want to rewrite the URL (there are caveats with this, so be careful) or respond in such a way as to indicate to the client that the image is not available for viewing.

IMPLEMENTING THE SOLUTION

In order to implement the solution you'll need to be able to intercept the request and examine the headers to determine its validity. We'll look at both ***mod_rewrite*** (Apache) and ***F5 iRules (***BIG-IP) as a mechanism to do this.

mod_rewrite

```
(mod_rewrite code courtesy of: Debian Administration)
```

```
Rewriteengine onRewriteCond %{HTTP_REFERER} !^$RewriteCond %{HTTP_REFERER} !^http://example.com/.*$ [NC]RewriteRule .*.(gif|GIF|jpg|JPG)$ - [F]
```

iRules

```
when HTTP_REQUEST {
    set thief 0
    set uri [HTTP::uri]
  if {[matchclass $uri ends_with $::images] > 0 } {
    if {[HTTP::header value Referer] contains "example.com"} {
      set thief 0
    }
    else {
      set thief 1
    }
  }
  if {$thief eq 1} {
    HTTP::respond 200 content ""
  }
  else {
    pool mywebsite_pool
  }
}
```

That's pretty much it. You could get creative and respond with an actual image, or rewrite the URI to be a different image. If you choose the latter, be aware that you'll need to add code to handle that exception case, or you'll put the client into a redirection loop. After all, the referrer is still not your site, so redirecting to another image will fall into the same code unless you specifically catch it. While Firefox will recognize this infinite loop and stop requesting the image, IE 7 just keeps trying, which is somewhat amusing but floods the network with requests that aren't going to get answered and uses up a connection on your web server or BIG-IP.

This solution obviously can be used to stop hotlinking to any type of content: Flash, video, audio, text. You only need change the extensions you are looking for to match those used by the content in question. You could also get more sophisticated and set up a system whereby allowed domains are given a cookie, which you can subsequently check to determine whether access should be allowed. You could also use this logic to stop specific domains from hotlinking to your content by checking the referer header against a list of allowed sites and refusing to serve the content to sites not on the list.

The nature of an intelligent mediator is such that you can pretty much come up with just about any solution involving HTTP headers and implement it fairly easily. There are advantages to using a full-proxy solution over mod_rewrite, but both will definitely provide a platform on which you can deploy a solution that can prevent content theft.

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com