Blocking web-request caused by referrer spam bots

Lately this site (and gentoo.linuxhowtos.org) have been hit by massive referer spam requests. Referer spam requests are requests for some pages with a faked referer string (where the user came from). Normally this is just annoying as it apears in your webstats. But these last times, the sites were hit by over 500 requests within ~30 secs. The requests came from very different ip-adresses, so blocking with iptables is not an option.

So what to do?
Well this is a tricky solution if those request happen again (which they did).
Add the following to your apache configuration in the block:

```
RewriteEngine on
RewriteBase /
RewriteCond %{HTTP_REFERER} nastydomain1
RewriteRule  ^.*           -                                          [F]
RewriteCond %{HTTP_REFERER} nastydomain2
RewriteRule  ^.*           -                                          [F]
RewriteCond %{HTTP_REFERER} nastydomain3
RewriteRule  ^.*           -                                          [F]
```

What does that do?
If there is a referer in the request (line 3/5/7)
and on of the words nastydomain1, nastydomain2 or nastydomain3 are in the referrer, send them a 403 forbidden message (line 4/6/8).

This way the server blocks the request early in the processing before any cgi-script is called. This saves CPU resources and bandwidth. |

| 167 | 211 | 1 | 2006-02-28 10:53:54 | Blocking web-request caused by referer spam bots | Lately this site (and gentoo.linuxhowtos.org) have been hit by massive referer spam requests.

Referer spam requests are requests for some pages with a faked referer string (where the user came from). Normally this is just annoying as it apears in your webstats (their intention).

But these last times, the sites were hit by over 500 requests within ~30 secs.

This triggered an alarm on my monitoring system

The requests came from very different IP-adresses, so blocking with iptables is not an option.

So what to do?

Well this is a tricky solution if those request happen again (which they did).

Add the following to your apache configuration in the >directory>>/directory> block:

```
RewriteEngine on
RewriteBase /
RewriteCond %{HTTP_REFERER} nastydomain1
RewriteRule ^.*        -         [F]
RewriteCond %{HTTP_REFERER} nastydomain2
RewriteRule ^.*        -         [F]
RewriteCond %{HTTP_REFERER} nastydomain3
RewriteRule ^.*        -         [F]
```

What does that do?

If there is a referer in the request (line 3/5/7)

and on of the words nastydomain1, nastydomain2 or nastydomain3 are in the referrer, send them a 403 forbidden message (line 4/6/8).

This way the server blocks the request early in the processing before any (cgi-)script is called. This saves CPU resources and bandwidth.

|
| 168 | 211 | 0 | 2006-02-28 10:54:35 | Blocking web-request caused by referer spam bots |

Blocking web-request caused by referer spam bots

Lately this site (and gentoo.linuxhowtos.org) have been hit by massive referer spam requests.

Referer spam requests are requests for some pages with a faked referrer string (where the user came from). Normally this is just annoying as it appears in your webstats (their intention).

But these last times, the sites were hit by over 500 requests within ~30 secs.

This triggered an alarm on my monitoring system

The requests came from very different IP-adresses, so blocking with iptables is not an option.

So what to do?

Well this is a tricky solution if those request happen again (which they did).

Add the following to your apache configuration in the >directory>>/directory> block:

```
RewriteEngine on
RewriteBase /
RewriteCond %{HTTP_REFERER} nastydomain1
RewriteRule ^.*         -         [F]
RewriteCond %{HTTP_REFERER} nastydomain2
RewriteRule ^.*         -         [F]
RewriteCond %{HTTP_REFERER} nastydomain3
RewriteRule ^.*         -         [F]
```

What does that do?

If there is a referrer in the request (line 3/5/7)

and on of the words nastydomain1, nastydomain2 or nastydomain3 are in the referrer, send them a 403 forbidden message (line 4/6/8).

This way the server blocks the request early in the processing before any (cgi-)script is called. This saves CPU resources and bandwidth.

A different approach

Since writing of this, my list increased to ~100 different domains.
Managing this in all virtual hosts became more and more work. A different solution was needed. This solution will be explained here:

Step 1

First create a textfile in your apache configuration directory called blacklist.txt.
This file will contain all nasty and spamming domains.
Example:

```
nastydomain1 -
nastydomain2 -
nastydomain3 -
....
```

The - at the end of the lines is important, but could be any character.
If you use another character, modify the RewriteCond shown later.

Step 2

Now we need to tell apache to load this file by adding the following to your apache.conf file:

```
Rewritemap refhashmap txt:/etc/apache2/blacklist.txt
```

Modify the path if your installation differs.

Step 3

Now add the following for every virtual host you want to have the filter running:

```
RewriteEngine on
RewriteBase /
RewriteCond %{HTTP_referer} ^http://([^/]+)
RewriteCond ${refhashmap:%1} ^-$
RewriteRule ^.*        -                                [F]
```

If you already have the RewriteEngine and RewriteBase lines, you don't need to repeat them.
If you changed the char in the blacklist file above, set the char in the RewriteCond ${refhashmap:%1} line into the ^-$ block.
This way you have one sitewide blacklist and cleaner configuration files.
|
| 218 | 211 | 0 | 2008-02-20 17:15:46 | Blocking web-request caused by referer spam bots |

Blocking web-request caused by referrer spam bots

Lately this site (and gentoo.linuxhowtos.org) have been hit by massive referrer spam requests.
Referrer spam requests are requests for some pages with a faked referrer string (where the user came from). Normally this is just annoying as it appears in your webstats (their intention).

But these last times, the sites were hit by over 500 requests within ~30 secs.

This triggered an alarm on my monitoring system

The requests came from very different IP-adresses, so blocking with iptables is not an option.

So what to do?

Well this is a tricky solution if those request happen again (which they did).

Add the following to your apache configuration in the >directory>>/directory> block:

```
RewriteEngine on
RewriteBase /
RewriteCond %{HTTP_REFERER} nastydomain1
RewriteRule ^.*         -         [F]
RewriteCond %{HTTP_REFERER} nastydomain2
RewriteRule ^.*         -         [F]
RewriteCond %{HTTP_REFERER} nastydomain3
RewriteRule ^.*         -         [F]
```

What does that do?

If there is a referrer in the request (line 3/5/7)

and on of the words nastydomain1, nastydomain2 or nastydomain3 are in the referrer, send them a 403 forbidden message (line 4/6/8).

This way the server blocks the request early in the processing before any (cgi-)script is called. This saves CPU resources and bandwidth.

A different approach

Since writing of this, my list increased to ~100 different domains.
Managing this in all virtual hosts became more and more work. A different solution was needed. This solution will be explained here:

Step 1

First create a textfile in your apache configuration directory called blacklist.txt.
This file will contain all nasty and spamming domains.
Example:

```
nastydomain1 -
nastydomain2 -
nastydomain3 -
....
```

The - at the end of the lines is important, but could be any character.
If you use another character, modify the RewriteCond shown later.

Step 2

Now we need to tell apache to load this file by adding the following to your apache.conf file:

```
Rewritemap refhashmap txt:/etc/apache2/blacklist.txt
```

Modify the path if your installation differs.

Step 3

Now add the following for every virtual host you want to have the filter running:

```
RewriteEngine on
RewriteBase /
RewriteCond %{HTTP_referer} ^http://([^/]+)
RewriteCond ${refhashmap:%1} ^-$
```

```
RewriteRule ^.*        -                                [F]
```

If you already have the RewriteEngine and RewriteBase lines, you don't need to repeat them.

If you changed the char in the blacklist file above, set the char in the RewriteCond ${refhashmap:%1} line into the ^-$ block.

This way you have one sitewide blacklist and cleaner configuration files.

current rating:

image:rdf newsfeed / //static.linuxhowtos.org/data/rdf.png (null)
|
image:rss newsfeed / //static.linuxhowtos.org/data/rss.png (null)
|
image:Atom newsfeed / //static.linuxhowtos.org/data/atom.png (null)
- Powered by
image:LeopardCMS / //static.linuxhowtos.org/data/leopardcms.png (null)
- Running on
image:Gentoo / //static.linuxhowtos.org/data/gentoo.png (null)
-

image:Valid XHTML1.1 / //static.linuxhowtos.org/data/xhtml.png (null)
:
image:Valid CSS / //static.linuxhowtos.org/data/css.png (null)
:
image:buttonmaker / //static.linuxhowtos.org/data/buttonmaker.png (null)
- Level Triple-A Conformance to Web Content Accessibility Guidelines 1.0 -
- Copyright and legal notices -
Time to create this page: ms
<!--
image:system status display / /status/output.jpg (null)
-->