链滴

nginx限制某个IP同一时间段的访问次数

作者：figo930
原文链接：https://ld246.com/article/1370399114232
来源网站：链滴

如何设置能限制某个IP某一时间段的访问次数是一个让人头疼的问题，特别面对恶意的ddos攻的时候。其中CC攻击（Challenge Collapsar）是DDOS（分布式拒绝服务）的一种，也是一种常见网站攻击方法，攻击者通过代理服务器或者肉鸡向向受害主机不停地发大量数据包，造成对方服务器源耗尽，一直到宕机崩溃。
cc攻击一般就是使用有限的ip数对服务器频繁发送数据来达到攻击的目的，nginx可以通过HttpLmitReqModul和HttpLimitZoneModule配置来限制ip在同一时间段的访问次数来防cc攻击。
HttpLimitReqModul用来限制连单位时间内连接数的模块，使用limit_req_zone和limit_req指配合使用来达到限制。一旦并发连接超过指定数量，就会返回503错误。
HttpLimitConnModul用来限制单个ip的并发连接数，使用limit_zone和limit_conn指令
这两个模块的区别前一个是对一段时间内的连接数限制，后者是对同一时刻的连接数限制

<h3>HttpLimitReqModul 限制某一段时间内同一ip访问数实例</h3>
<pre>http{
    ...

#定义一个名为allips的limit_req_zone用来存储session，大小是10M内存，
#以$binary_remote_addr 为key,限制平均每秒的请求为20个，
#1M能存储16000个状态，rete的值必须为整数，
#如果限制两秒钟一个请求，可以设置成30r/m

limit_req_zone $binary_remote_addr zone=allips:10m rate=20r/s;
...
server{
    ...
    location {
        ...

        #限制每ip每秒不超过20个请求，漏桶数burst为5
        #brust的意思就是，如果第1秒、2,3,4秒请求为19个，
        #第5秒的请求为25个是被允许的。
        #但是如果你第1秒就25个请求，第2秒超过20的请求返回503错误。
        #nodelay，如果不设置该选项，严格使用平均速率限制请求数，
        #第1秒25个请求时，5个请求放到第2秒执行，
        #设置nodelay，25个请求将在第1秒执行。

        limit_req zone=allips burst=5 nodelay;
        ...
    }
    ...
}
...

}</pre>

<h3>HttpLimitZoneModule 限制并发连接数实例</h3>
limit_zone只能定义在http作用域，limit_conn可以定义在http server location作用域
<pre>http{
    ...

#定义一个名为one的limit_zone,大小10M内存来存储session，

```
#以$binary_remote_addr 为key
#nginx 1.18以后用limit_conn_zone替换了limit_conn
#且只能放在http作用域
limit_conn_zone   one  $binary_remote_addr  10m;  
...
server{
    ...
    location {
        ...
        limit_conn one 20;          #连
数限制

        #带宽限制,对单个连接限数，如果一个ip两个连接，就是500x2k
        limit_rate 500k;          
nbsp;


        ...
    }
    ...
}
...

}</pre>

<p> </p>
<h3>nginx白名单设置</h3>
<p>以上配置会对所有的ip都进行限制，有些时候我们不希望对搜索引擎的蜘蛛或者自己测试ip进行
制，<br />对于特定的白名单ip我们可以借助geo指令实现。<br />1.</p>
<pre>http{
    geo $limited{
        default 1;
        #google
        64.233.160.0/19 0;
        65.52.0.0/14 0;
        66.102.0.0/20 0;
        66.249.64.0/19 0;
        72.14.192.0/18 0;
        74.125.0.0/16 0;
        209.85.128.0/17 0;
        216.239.32.0/19 0;
        #M$
        64.4.0.0/18 0;
        157.60.0.0/16 0;
        157.54.0.0/15 0;
        157.56.0.0/14 0;
        207.46.0.0/16 0;
        207.68.192.0/20 0;
        207.68.128.0/18 0;
        #yahoo
        8.12.144.0/24 0;
        66.196.64.0/18 0;
        66.228.160.0/19 0;
        67.195.0.0/16 0;
        74.6.0.0/16 0;
```

```
        68.142.192.0/18 0;
        72.30.0.0/16 0;
        209.191.64.0/18 0;
        #My IPs
        127.0.0.1/32 0;
        123.456.0.0/28 0; #example for your server CIDR
    }</pre>
<p>geo指令定义了一个白名单$limited变量，默认值为1，如果客户端ip在上面的范围内，$limited值为0</p>
<p>2.使用map指令映射搜索引擎客户端的ip为空串，如果不是搜索引擎就显示本身真是的ip，这样索引擎ip就不能存到limit_req_zone内存session中，所以不会限制搜索引擎的ip访问</p>
<p>map $limited $limit {<br />1 $binary_remote_addr;<br />0 &quot;&quot;;<br />}</p>
<p>3.设置limit_req_zone和limit_req<br />limit_req_zone $limit zone=foo:1m rate=10r/m;</
>
<p>limit_req zone=foo burst=5;</p>
<p><strong>最后我们使用ab压php-fpm的方式，对上面的方法效果实际测试下</strong></p>
<p>例1：限制只允许一分钟内只允许一个ip访问60次配置，也就是平均每秒1次<br />首先我们准一个php脚本放在根目录下$document_root<br />test.php</p>
<blockquote>
 <p>for( $i=0; $i &lt; 1000; $i++)<br />echo 'Hello World';<br />?&gt;</p>
</blockquote>
<p>nginx配置增加limit_req_zone 和 limit_req</p>
<pre>http{
    ...
    limit_req_zone $binary_remote_addr zone=allips:10m rate=60r/m;
    ...
    server{
        ...
        location {
            ...
            limit_req zone=allips;
            ...
        }
        ...
    }
    ...
}</pre>
<p>ab -n 5 -c 1 http://www.weizhang.org/test.php</p>
<p>118.144.94.193 - - [22/Dec/2012:06:27:06 +0000] &quot;GET /test.php HTTP/1.0&quot; 2
0 11000 &quot;-&quot; &quot;ApacheBench/2.3&quot;<br />118.144.94.193 - - [22/Dec/201
:06:27:06 +0000] &quot;GET /test.php HTTP/1.0&quot; 503 537 &quot;-&quot; &quot;Apach
Bench/2.3&quot;<br />118.144.94.193 - - [22/Dec/2012:06:27:07 +0000] &quot;GET /test.php
HTTP/1.0&quot; 503 537 &quot;-&quot; &quot;ApacheBench/2.3&quot;<br />118.144.94.19
 - - [22/Dec/2012:06:27:07 +0000] &quot;GET /test.php HTTP/1.0&quot; 503 537 &quot;-&q
ot; &quot;ApacheBench/2.3&quot;<br />118.144.94.193 - - [22/Dec/2012:06:27:07 +0000] &
uot;GET /test.php HTTP/1.0&quot; 503 537 &quot;-&quot; &quot;ApacheBench/2.3&quot;</
>
<p>未设置brust和nodelay可以看到该配置只允许每秒访问1次，超出的请求返回503错误</p>
<pre>http{
    ...
    limit_req_zone $binary_remote_addr zone=allips:10m rate=60r/m;
    ...
    server{
        ...
```

```
        location {
            ...
            limit_req zone=allips burst=1 nodelay;
            ...
        }
        ...
    }
    ...
}</pre>
<p>ab -n 5 -c 1 http://www.weizhang.org/test.php</p>
<p>118.144.94.193 - - [22/Dec/2012:07:01:00 +0000] &quot;GET /test.php HTTP/1.0&quot; 2
0 11000 &quot;-&quot; &quot;ApacheBench/2.3&quot;<br />118.144.94.193 - - [22/Dec/201
:07:01:00 +0000] &quot;GET /test.php HTTP/1.0&quot; 200 11000 &quot;-&quot; &quot;Apa
heBench/2.3&quot;<br />118.144.94.193 - - [22/Dec/2012:07:01:01 +0000] &quot;GET /test.
hp HTTP/1.0&quot; 503 537 &quot;-&quot; &quot;ApacheBench/2.3&quot;<br />118.144.94
193 - - [22/Dec/2012:07:01:01 +0000] &quot;GET /test.php HTTP/1.0&quot; 503 537 &quot;-
quot; &quot;ApacheBench/2.3&quot;<br />118.144.94.193 - - [22/Dec/2012:07:01:01 +0000]
&quot;GET /test.php HTTP/1.0&quot; 503 537 &quot;-&quot; &quot;ApacheBench/2.3&quot
</p>
<p>设置brust=1和nodelay后允许第1秒处理两个请求。</p>
```