通过 Nginx 和 Nginx Plus 阻止 DDoS 攻击

作者：Vanessa
原文链接：https://ld246.com/article/1454399153862
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

分布式拒绝服务攻击（DDoS）指的是通过多台机器向一个服务或者网站发送大量看似合法的数包使其网络阻塞、资源耗尽从而不能为正常用户提供正常服务的攻击手段。随着互联网带宽的增加和关工具的不断发布，这种攻击的实施难度越来越低，有大量IDC托管机房、商业站点、游戏服务商一饱受DDoS攻击的困扰，那么如何缓解甚至解决DDoS呢？最近Rick Nelson在Nginx的官方博客上发表了一篇文章，介绍如何通过Nginx和Nginx Plus缓和DDoS攻击。
Rick Nelson首先介绍了DDoS攻击的一些特点,例如攻击的流量通常来源于一些固定的IP地址，一个IP地址会创建比真实用户多得多的连接和请求；同时由于流量全部是由机器产生的，其速率要比类用户高的多。此外，进行攻击的机器其User-Agent头也不是标准的值，Referer头有时也会被设置能够与攻击关联起来的值。针对这些特点，Rick Nelson认为Nginx和Nginx Plus有很多能够通过调或控制流量来应对或者减轻DDoS攻击的特性。
限制请求率
将Nginx和Nginx Plus可接受的入站请求率限制为个适合真实用户的值。例如，通过下面的配置让一个真正的用户每两秒钟才能访问一次登录页面：</>
<pre><code>limit_req_zone $binary_remote_addr zone=one:10m rate=30r/m;

server {

...

location /login.html {

limit_req zone=one;

...

}

}

</code></pre>

在该配置中，<code>limit_req_zone</code>指令配置了一个名为<code>one</code>的共内存<code>zone</code>用来存储<code>$binary_remote_addr</code>的请求状态，locatio块中/login.html的<code>limit_req</code>指令引用了共享内存<code>zone</code>。
限制连接的数量
将某个客户端IP地址所能打开的连接数限制为实用户的合理值。例如，限制每一个IP对网站/store部分打开的连接数不超过10个：
<pre><code>limit_conn_zone $binary_remote_addr zone=addr:10m;

server {

...

location /store/ {

limit_conn addr 10;

...

}

}

</code></pre>

该配置中，<code>limit_conn_zone</code>指令配置了一个名为<code>addr</code>的共内存<code>zone</code>用来存储<code>$binary_remote_addr</code>的请求，location块中/tore/的<code>limit_conn</code>指令引用了共享内存<code>zone</code>，并将最大连接数置为10.

原文链接：通过 Nginx 和 Nginx Plus 阻止 DDoS 攻击

关闭慢连接
关闭那些一直保持打开同时写数据又特别频繁的连，因为它们会降低服务器接受新连接的能力。Slowloris就是这种类型的攻击。对此，可以通过<code>client_body_timeout</code>和<code>client_header_timeout</code>指令控制请求体或者请头的超时时间，例如，通过下面的配置将等待时间控制在5s之内：
<pre><code>server {
    client_body_timeout 5s;
    client_header_timeout 5s;
    ...
}
</code></pre>
设置IP黑名单
如果能识别攻击者所使用的客户端IP地址，那么过<code>deny</code>指令将其屏蔽，让Nginx和Nginx Plus拒绝来自这些地址的连接或请求。例，通过下面的指令拒绝来自123.123.123.3、123.123.123.5和123.123.123.7的请求：
<pre><code>location / {
    deny 123.123.123.3;
    deny 123.123.123.5;
    deny 123.123.123.7;
    ...
}
</code></pre>
设置IP白名单
如果允许访问的IP地址比较固定，那么通过<code>allow</code>和<code>deny</code>指令让网站或者应用程序只接受来自于某个IP地址或者某个P地址段的请求。例如，通过下面的指令将访问限制为本地网络的一个IP段：
<pre><code>location / {
    allow 192.168.1.0/24;
    deny all;
    ...
}
</code></pre>
通过缓存削减流量峰值
通过启用缓存并设置某些缓存参数让Ngix和Nginx Plus吸收攻击所产生的大部分流量峰值。例如，通过<code>proxy_cache_use_stale</code>e>指令的<code>updating</code>参数告诉Nginx何时需要更新过期的缓存对象，避免因重复发更新请求对后端服务器产生压力。另外，<code>proxy_cache_key</code>指令定义的键通常会包嵌入的变量，例如默认的键<code>$scheme$proxy_host$request_uri</code>包含了三个变量，果它包含<code>$query_string</code>变量，那么攻击者可以通过发送随机的<code>query_strin</code>值来耗尽缓存，因此，如果没有特别原因，不要在该键中使用<code>$query_string</cod>变量。
阻塞请求
配置Nginx和Nginx Plus阻塞以下类型的请求：

以某个特定URL为目标的请求
User-Agent头中的值不在正常客户端范围之内的请求
Referer头中的值能够与攻击关联起来的请求
其他头中存在能够与攻击关联在一起的值的请求

例如，通过下面的配置阻塞以/foo.php为目标的攻击：
<pre><code>location /foo.php {
    deny all;
}
</code></pre>
或者通过下面的配置阻塞已识别出的User-Agent头的值是foo或者bar的DDoS攻击：
<pre><code>location / {
    if ($http_user_agent ~* foo|bar) {
        return 403;
    }

```
    ...
}
```

限制对后端服务器的连接数
通常Nginx和Nginx Plus实例能够理比后端服务器多得多的连接数，因此可以通过Nginx Plus限制到每一个后端服务器的连接数。例如通过下面的配置限制Nginx Plus和每一台后端服务器之间建立的连接数不多于200个：

```
upstream website {
    server 192.168.100.1:80 max_conns=200;
    server 192.168.100.2:80 max_conns=200;
    queue 10 timeout=30s;
}
```

另外，Rick Nelson还提到了如何处理基于范围的攻击和如何处理高负载的问题，以及如何使用Nginx Plus Status模块发现异常的流量模式，定位DDoS攻击。

转自：http://www.infoq.com/cn/news/2016/01/Nginx-AntiDDoS