优化Nginx服务的安全配置

Nginx安全优化包括：删除不要的模块、修改版本信息、限制并发、拒绝非法请求、防止buffer溢出。

优化Nginx服务的安全配置

1） 删除不需要的模块

Nignx是模块化设计的软件，需要什么功能与模块以及不需要哪些模块，都可以在编译安装软件时自定义，使用--with参数可以开启某些模块，使用--without可以禁用某些模块。最小化安装永远都是对的方案！

下面是禁用某些模块的案例：

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY05/CASE/01/index.html)

1. [root@proxy ~]# tar -xf nginx-1.12.tar.gz
2. [root@proxy ~]# cd nginx-1.12
3. [root@proxy nginx-1.12]# ./configure \
4. >--without-http\_autoindex\_module \            //禁用自动索引文件目录模块
5. >--without-http\_ssi\_module
6. [root@proxy nginx-1.12]# make
7. [root@proxy nginx-1.12]# make install

[root@proxy ~]# tar -xf nginx-1.12.tar.gz

[root@proxy ~]# cd nginx-1.12

[root@proxy nginx-1.12]# ./configure \

>--without-http\_autoindex\_module \ //禁用自动索引文件目录模块

>--without-http\_ssi\_module

[root@proxy nginx-1.12]# make

[root@proxy nginx-1.12]# make install

2） 修改版本信息，并隐藏具体的版本号

默认Nginx会显示版本信息以及具体的版本号，这些信息给攻击者带来了便利性，便于他们找到具体版本的漏洞。

如果需要屏蔽版本号信息，执行如下操作，可以隐藏版本号。

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY05/CASE/01/index.html)

1. [root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf
2. … …
3. http{
4. server\_tokens off;                            //在http下面手动添加这么一行
5. … …
6. }
7. [root@proxy ~]# nginx -s reload
9. [root@proxy ~]# curl -I http://192.168.4.5 //查看服务器响应的头部信息

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

… …

http{

server\_tokens off; //在http下面手动添加这么一行

… …

}

[root@proxy ~]# nginx -s reload

[root@proxy ~]# curl -I http://192.168.4.5 //查看服务器响应的头部信息

3） 限制并发量

DDOS攻击者会发送大量的并发连接，占用服务器资源（包括连接数、带宽等），这样会导致正常用户处于等待或无法访问服务器的状态。

Nginx提供了一个ngx\_http\_limit\_req\_module模块，可以有效降低DDOS攻击的风险，操作方法如下：

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY05/CASE/01/index.html)

1. [root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf
2. … …
3. http{
4. … …
5. limit\_req\_zone $binary\_remote\_addr zone=one:10m rate=1r/s;
6. server {
7. listen 80;
8. server\_name localhost;
9. limit\_req zone=one burst=5;
10. }
11. }
13. //备注说明：
14. //limit\_req\_zone语法格式如下：
15. //limit\_req\_zone key zone=name:size rate=rate;
16. //上面案例中是将客户端IP信息存储名称为one的共享内存，内存空间为10M
17. //1M可以存储8千个IP信息，10M可以存储8万个主机连接的状态，容量可以根据需要任意调整
18. //每秒中仅接受1个请求，多余的放入漏斗
19. //漏斗超过5个则报错
20. [root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

… …

http{

… …

limit\_req\_zone $binary\_remote\_addr zone=one:10m rate=1r/s;

server {

listen 80;

server\_name localhost;

limit\_req zone=one burst=5;

}

}

//备注说明：

//limit\_req\_zone语法格式如下：

//limit\_req\_zone key zone=name:size rate=rate;

//上面案例中是将客户端IP信息存储名称为one的共享内存，内存空间为10M

//1M可以存储8千个IP信息，10M可以存储8万个主机连接的状态，容量可以根据需要任意调整

//每秒中仅接受1个请求，多余的放入漏斗

//漏斗超过5个则报错

[root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload

客户端使用ab测试软件测试效果：

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY05/CASE/01/index.html)

1. [root@client ~]# ab -c 100 -n 100 http://192.168.4.5/

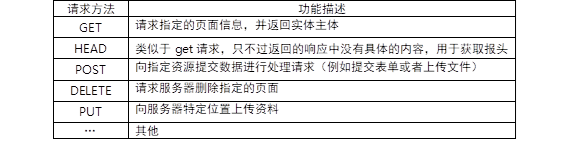
[root@client ~]# ab -c 100 -n 100 http://192.168.4.5/

4） 拒绝非法的请求

网站使用的是HTTP协议，该协议中定义了很多方法，可以让用户连接服务器，获得需要的资源。但实际应用中一般仅需要get和post。

具体HTTP请求方法的含义如表-1所示。

表-1 HTTP请求方法及含义



未修改服务器配置前，客户端使用不同请求方法测试：

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY05/CASE/01/index.html)

1. [root@client ~]# curl -i -X GET http://192.168.4.5            //正常
2. [root@client ~]# curl -i -X HEAD http://192.168.4.5            //正常
3. //curl命令选项说明：
4. //-i选项：访问服务器页面时，显示HTTP的头部信息
5. //-X选项：指定请求服务器的方法

[root@client ~]# curl -i -X GET http://192.168.4.5 //正常

[root@client ~]# curl -i -X HEAD http://192.168.4.5 //正常

//curl命令选项说明：

//-i选项：访问服务器页面时，显示HTTP的头部信息

//-X选项：指定请求服务器的方法

通过如下设置可以让Nginx拒绝非法的请求方法：

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY05/CASE/01/index.html)

1. [root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf
2. http{
3. server {
4. listen 80;
5. #这里，!符号表示对正则取反，~符号是正则匹配符号
6. #如果用户使用非GET或POST方法访问网站，则retrun返回错误信息
7. if ($request\_method !~ ^(GET|POST)$ ) {
8. return 444;
9. }
10. }
11. }
12. [root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload

[root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf

http{

server {

listen 80;

#这里，!符号表示对正则取反，~符号是正则匹配符号

#如果用户使用非GET或POST方法访问网站，则retrun返回错误信息

if ($request\_method !~ ^(GET|POST)$ ) {

return 444;

}

}

}

[root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload

修改服务器配置后，客户端使用不同请求方法测试：

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY05/CASE/01/index.html)

1. [root@client ~]# curl -i -X GET http://192.168.4.5            //正常
2. [root@client ~]# curl -i -X HEAD http://192.168.4.5            //报错

[root@client ~]# curl -i -X GET http://192.168.4.5 //正常

[root@client ~]# curl -i -X HEAD http://192.168.4.5 //报错

4） 防止buffer溢出

当客户端连接服务器时，服务器会启用各种缓存，用来存放连接的状态信息。

如果攻击者发送大量的连接请求，而服务器不对缓存做限制的话，内存数据就有可能溢出（空间不足）。

修改Nginx配置文件，调整各种buffer参数，可以有效降低溢出风险。

[copytextpop-up](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN202001/SECURITY/DAY05/CASE/01/index.html)

1. [root@proxy ~]# vim /usr/local/nginx/conf/nginx.conf
2. http{
3. client\_body\_buffer\_size 1K;
4. client\_header\_buffer\_size 1k;
5. client\_max\_body\_size 1k;
6. large\_client\_header\_buffers 2 1k;
7. … …
8. }
9. [root@proxy ~]# /usr/local/nginx/sbin/nginx -s reload