目录

01 系统加固

- Windows操作系统加固
- Linux操作系统加固

02 中间件加固

- IIS加固
- Apache加固
- Nginx加固

03 数据库加固

- Mysql加固
- Mongodb加固

Nginx 安全加固

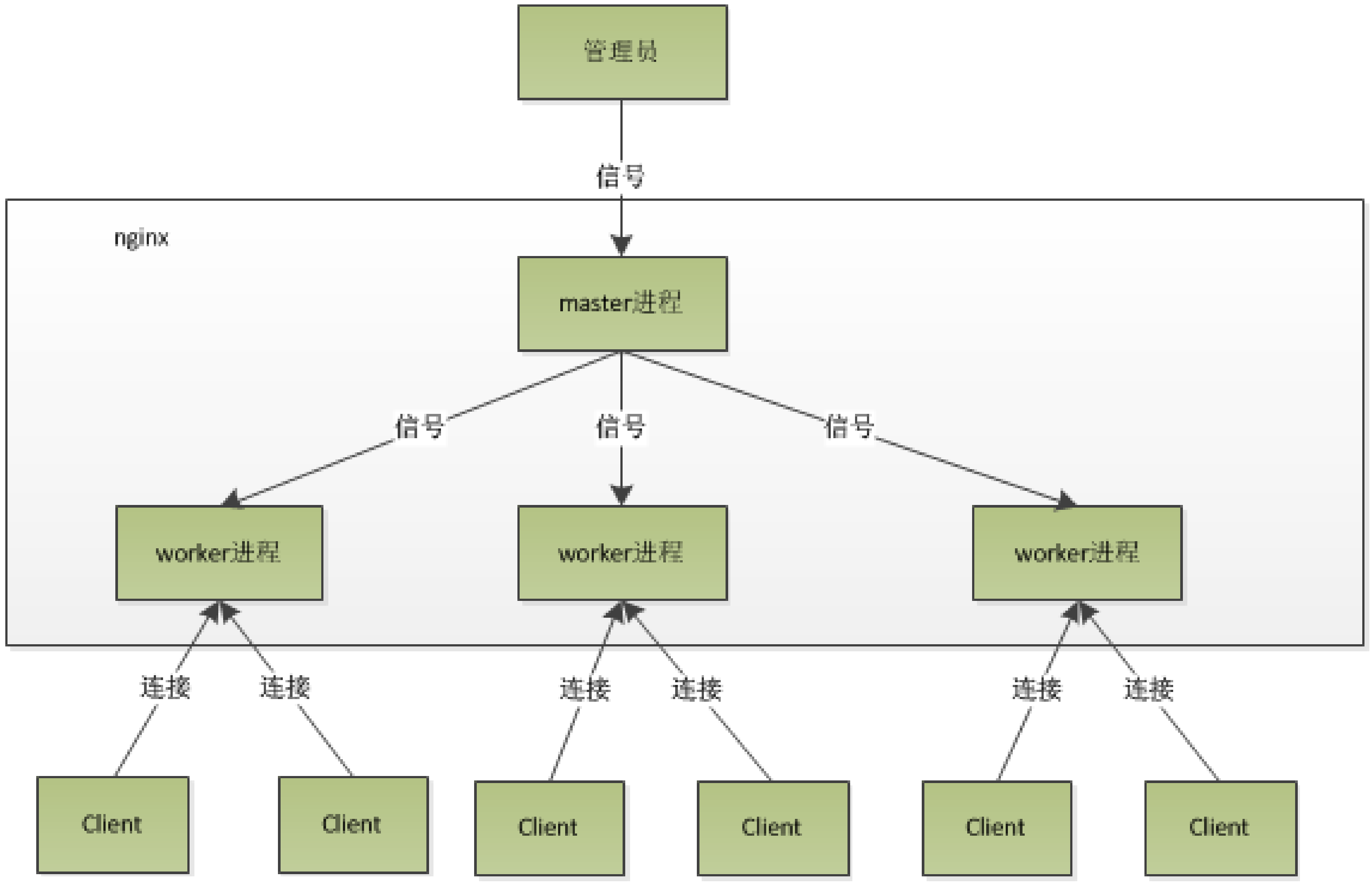
目录

01 Nginx 基础

02 Nginx 服务器安全加固

Nginx 基础

Nginx架构


管理员
信号
nginx
master进程
信号
信号
信号
worker进程
worker进程
worker进程
连接
连接
连接
连接
连接
连接
Client
Client
Client
Client
Client
Client

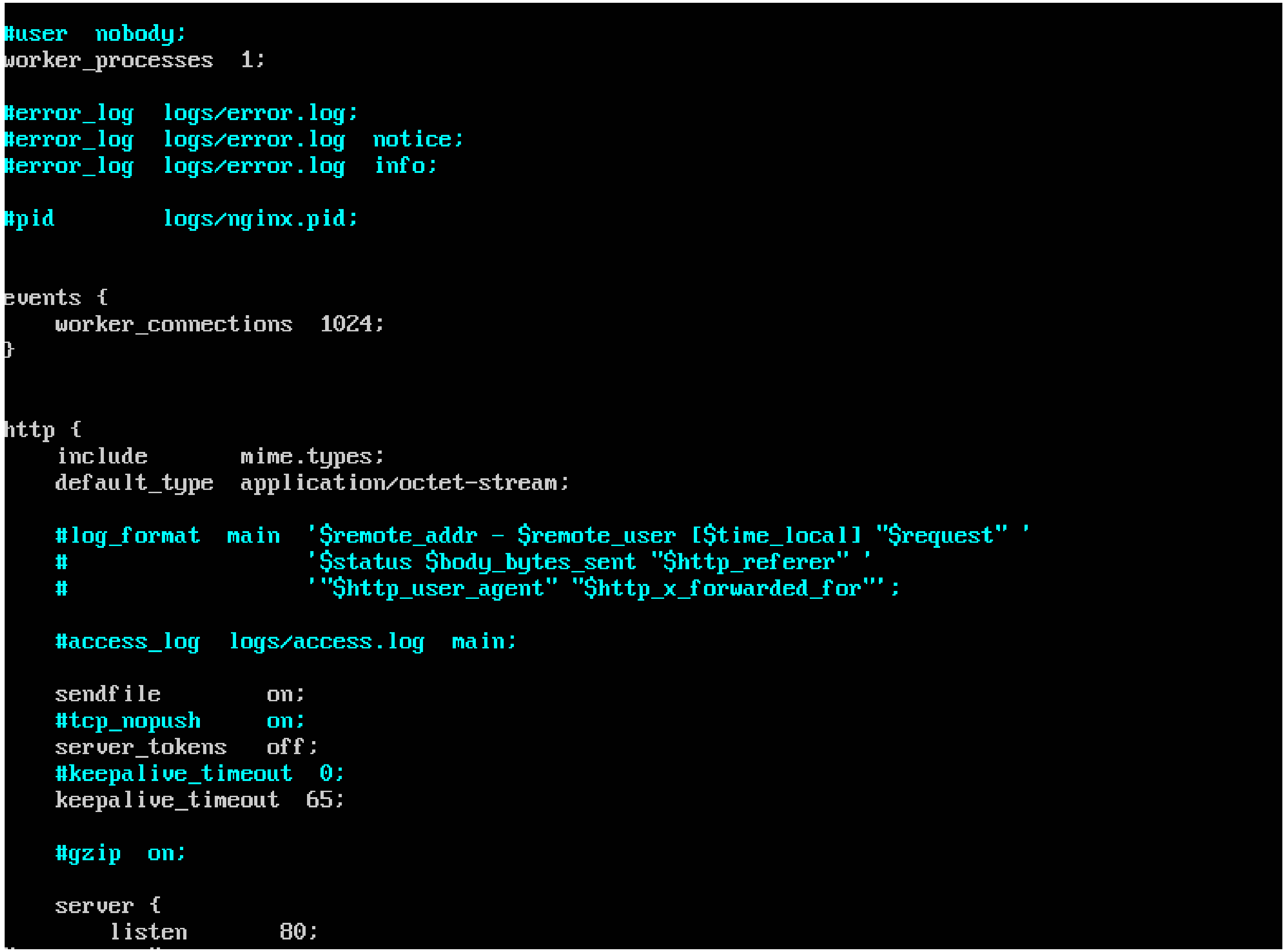
安装nginx

我们采用源码安装方式

```
apt-get install openssl libssl-dev          #解决依赖包openssl安装
apt-get install libpcre3 libpcre3-dev       #解决依赖包pcre安装
apt-get install zlib1g-dev                  #解决依赖包zlib安装
apt-get install build-essential             #解决依赖包gcc问题

./configure --prefix=/usr/local/nginx

 make && make install

 /usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx.conf
```

nginx.conf

配置文件是nginx的核心！！！
/usr/local/nginx/conf/nginx.conf

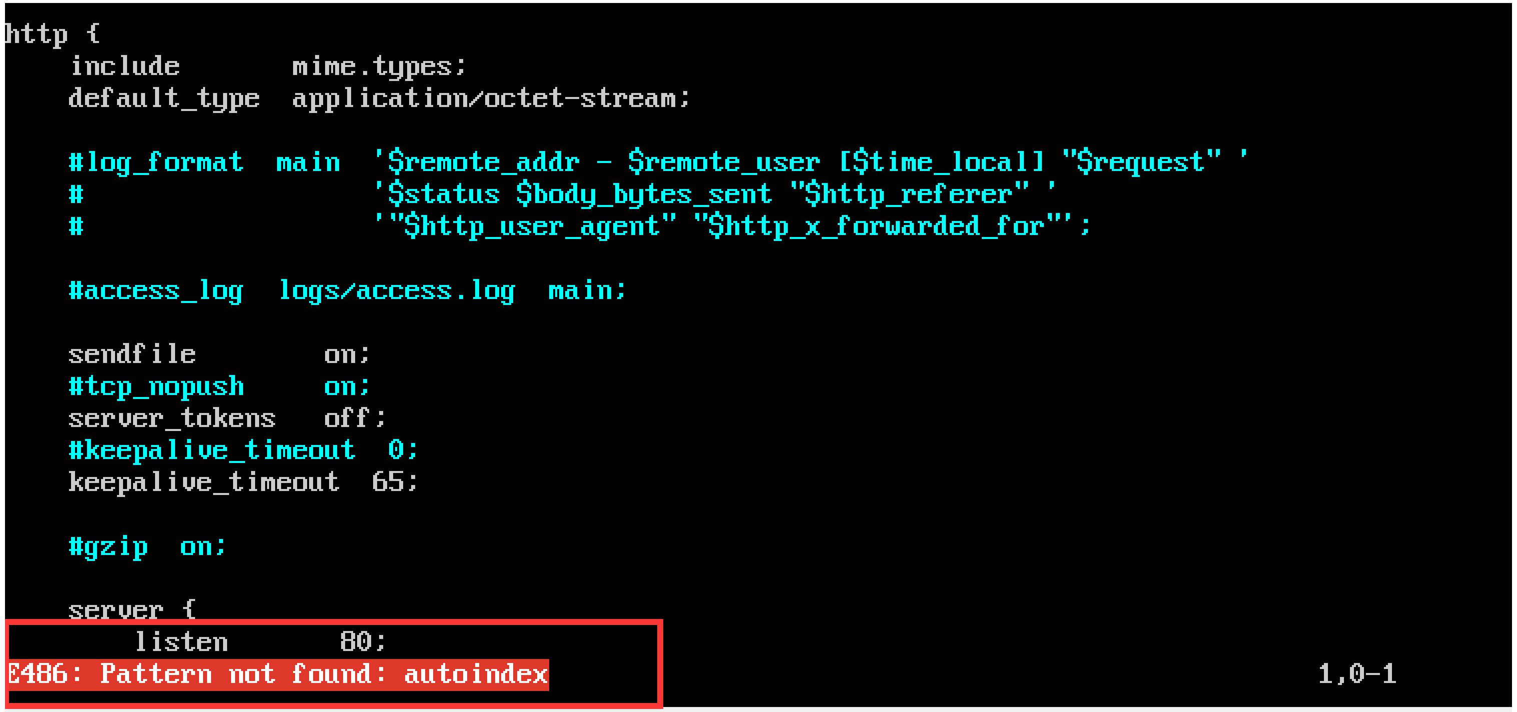


```
#user  nobody;
worker_processes  1;

#error_log  logs/error.log;
#error_log  logs/error.log  notice;
#error_log  logs/error.log  info;

#pid        logs/nginx.pid;


events {
    worker_connections  1024;
}


http {
    include       mime.types;
    default_type  application/octet-stream;

    #log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
    #                  '$status $body_bytes_sent "$http_referer" '
    #                  '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log  logs/access.log  main;

    sendfile        on;
    #tcp_nopush     on;
    server_tokens   off;
    #keepalive_timeout  0;
    keepalive_timeout  65;

    #gzip  on;

    server {
        listen       80;
```

Nginx 服务器安全加固

禁用autoindex

确保nginx.conf配置文件上禁用autoindex，即autoindex off或者没有配置autoindex



```
http {
    include       mime.types;
    default_type  application/octet-stream;

    #log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
    #                  '$status $body_bytes_sent "$http_referer" '
    #                  '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log  logs/access.log  main;

    sendfile        on;
    #tcp_nopush     on;
    server_tokens   off;
    #keepalive_timeout  0;
    keepalive_timeout  65;

    #gzip  on;

    server {
        listen       80;
E486: Pattern not found: autoindex                                          1,0-1
```

关闭服务器标记

如果开启的话（默认情况下）所有的错误页面都会显示服务器的版本和信息。nginx.conf配置如下：

```
http{
    include       naxsi_core.rules;
    include      mime.types;
    default_type  application/octet-stream;
    sendfile        on;
    server_tokens off;
    ... ...
```

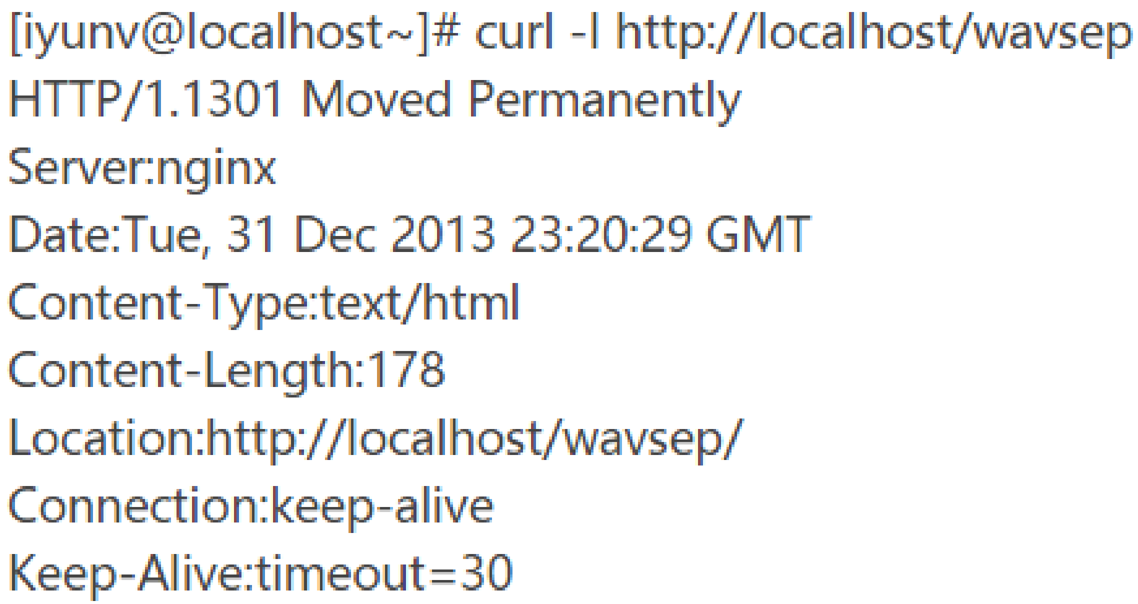
同时修改/usr/local/nginx/conf/fastcgi_params

将里面的

```
fastcgi_param SERVER_SOFTWARE nginx/$nginx_version;
```

修改为：

```
fastcgi_param SERVER_SOFTWARE nginx;
```



```
[iyunv@localhost~]# curl -I http://localhost/wavsep
HTTP/1.1301 Moved Permanently
Server:nginx
Date:Tue, 31 Dec 2013 23:20:29 GMT
Content-Type:text/html
Content-Length:178
Location:http://localhost/wavsep/
Connection:keep-alive
Keep-Alive:timeout=30
```

自定义缓存

设置自定义缓存以限制缓冲区溢出攻击。nginx.conf配置如下：

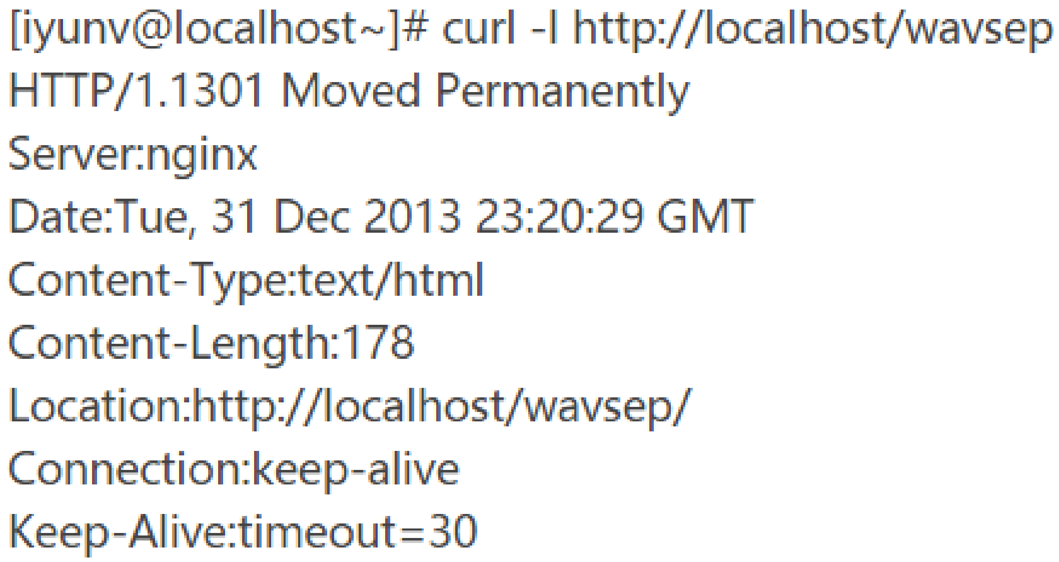
```
http{
  ... ...
  server{
     ... ...
     client_body_buffer_size  16K;
    client_header_buffer_size  1k;
     client_max_body_size  1m;
    large_client_header_buffers  4  8k;
```

注：上述的参数不是最优参数，仅供参考。

timeout设置

设置timeout设低来防御DOS攻击，nginx.conf配置如下：

```
http {
  ... ...
     client_body_timeout   10;
     client_header_timeout  30;
     keepalive_timeout     30  30;
     send_timeout          10;
```

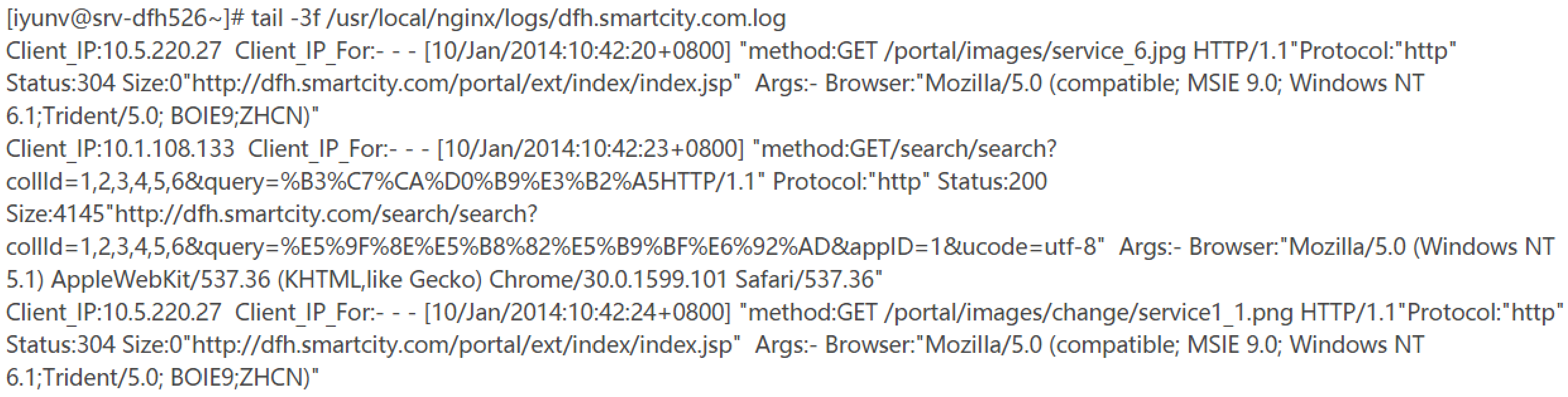


```
[iyunv@localhost~]# curl -I http://localhost/wavsep
HTTP/1.1301 Moved Permanently
Server:nginx
Date:Tue, 31 Dec 2013 23:20:29 GMT
Content-Type:text/html
Content-Length:178
Location:http://localhost/wavsep/
Connection:keep-alive
Keep-Alive:timeout=30
```

配置日志

鉴于日志的输出格式还未确定，目前暂时先使用Nginx默认的日志格式。nginx.conf配置如下：

```
http {
    ......
    log_format  main  '$remote_addr - $remote_user [$time_local]"$request" "$status $body_bytes_sent
"$http_referer"""$http_user_agent" "$http_x_forwarded_for"';
    access_log logs/ access.log  main;
```



```
[iyunv@srv-dfh526~]# tail -3f /usr/local/nginx/logs/dfh.smartcity.com.log
Client_IP:10.5.220.27  Client_IP_For:- - - [10/Jan/2014:10:42:20+0800] "method:GET /portal/images/service_6.jpg HTTP/1.1"Protocol:"http"
Status:304 Size:0"http://dfh.smartcity.com/portal/ext/index/index.jsp"  Args:- Browser:"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT
6.1;Trident/5.0; BOIE9;ZHCN)"
Client_IP:10.1.108.133  Client_IP_For:- - - [10/Jan/2014:10:42:23+0800] "method:GET/search/search?
collId=1,2,3,4,5,6&query=%B3%C7%CA%D0%B9%E3%B2%A5HTTP/1.1" Protocol:"http" Status:200
Size:4145"http://dfh.smartcity.com/search/search?
collId=1,2,3,4,5,6&query=%E5%9F%8E%E5%B8%82%E5%B9%BF%E6%92%AD&appID=1&ucode=utf-8"  Args:- Browser:"Mozilla/5.0 (Windows NT
5.1) AppleWebKit/537.36 (KHTML,like Gecko) Chrome/30.0.1599.101 Safari/537.36"
Client_IP:10.5.220.27  Client_IP_For:- - - [10/Jan/2014:10:42:24+0800] "method:GET /portal/images/change/service1_1.png HTTP/1.1"Protocol:"http"
Status:304 Size:0"http://dfh.smartcity.com/portal/ext/index/index.jsp"  Args:- Browser:"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT
6.1;Trident/5.0; BOIE9;ZHCN)"
```

限制访问

在目前的应用系统中值使用到POST和GET方法，所以除了它们之外，其他方式的请求均可拒绝。Nginx.conf配置如下：

```
server{
      … …
      if($request_method !~ ^(GET|HEAD|POST)$) {
            return404;
         }
      … …
```

限制访问IP

模块 ngx_http_access_module 允许限制某些IP地址的客户端访问。

如下范例：

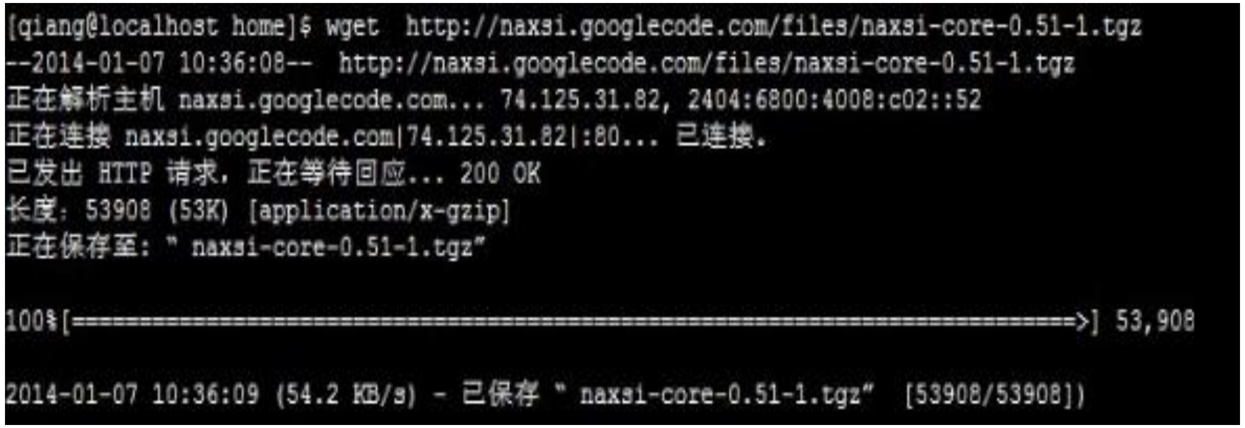
```
location/ {
    deny  192.168.1.1;
    allow 192.168.1.0/24;
    allow 10.1.1.0/16;
    allow 2001:0db8::/32;
    deny  all;
}
```

注：规则按照顺序依次检测，直到匹配到第一条规则。 在这个例子里，IPv4的网络中只有 10.1.1.0/16 和 192.168.1.0/24允许访问，但 192.168.1.1除外, 对于IPv6的网络，只有2001:0db8::/32允许访问。

集成Naxsi模块

Naxsi模块的集成，是基于Nginx已经部署了或已经存在系统中。

第一步：下载naxsi
Wget http://naxsi.googlecode.com/files/naxsi-core-0.51-1.tgz
注：如果不能上网可以事先下载，再上传到服务器中。



```
[qiang@localhost home]$ wget  http://naxsi.googlecode.com/files/naxsi-core-0.51-1.tgz
--2014-01-07 10:36:08--  http://naxsi.googlecode.com/files/naxsi-core-0.51-1.tgz
正在解析主机 naxsi.googlecode.com... 74.125.31.82, 2404:6800:4008:c02::52
正在连接 naxsi.googlecode.com|74.125.31.82|:80... 已连接.
已发出 HTTP 请求，正在等待回应... 200 OK
长度： 53908 (53K) [application/x-gzip]
正在保存至: " naxsi-core-0.51-1.tgz"

100%[=============================================================================>] 53,908

2014-01-07 10:36:09 (54.2 KB/s) - 已保存 " naxsi-core-0.51-1.tgz"  [53908/53908])
```

集成Naxsi模块

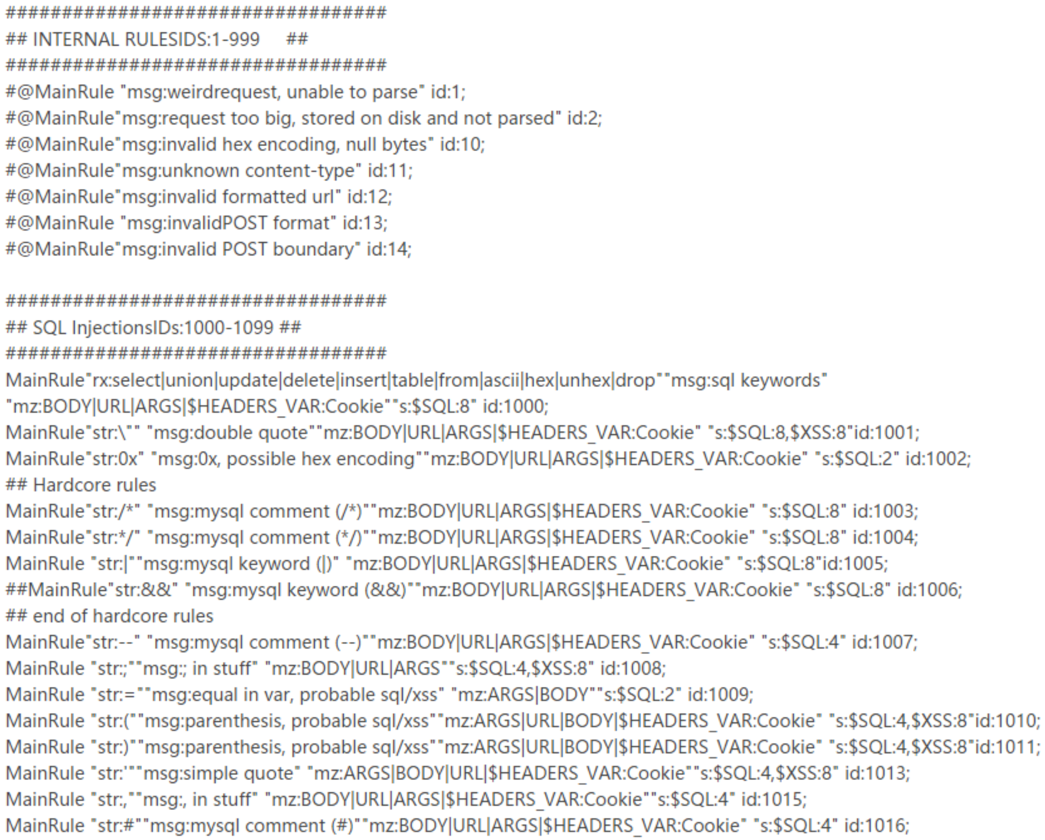
第二步：解压naxsi

[qiang@localhost install]$ tar -zxvfnaxsi-core-0.51-1.tgz

第三步：切换到naxsi-core-0.51-1目录，并复制其配置文件到nginx.conf同目录下

[qiang@localhostnaxsi_config]$ cp naxsi_core.rules /etc/nginx/naxsi_core.rules

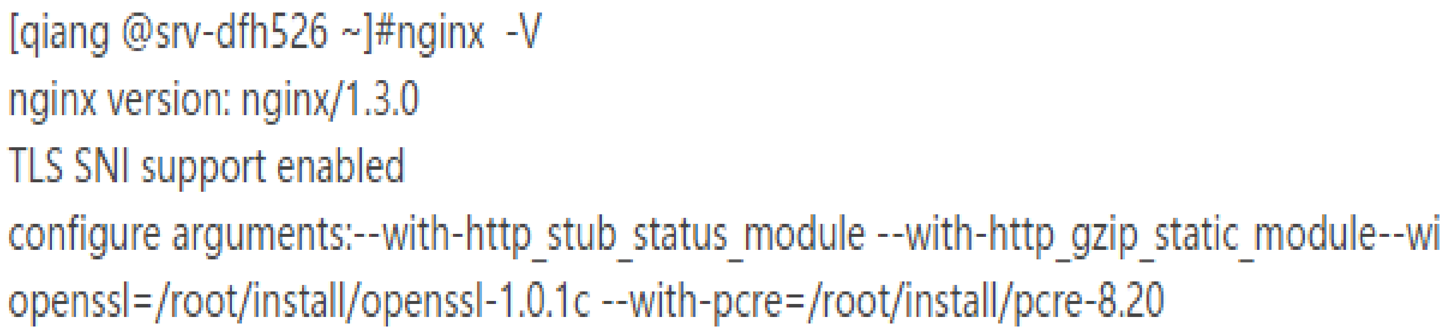
修改naxsi_core.rules的配置如下：



```
##################################
## INTERNAL RULESIDS:1-999     ##
##################################
#@MainRule "msg:weirdrequest, unable to parse" id:1;
#@MainRule"msg:request too big, stored on disk and not parsed" id:2;
#@MainRule"msg:invalid hex encoding, null bytes" id:10;
#@MainRule"msg:unknown content-type" id:11;
#@MainRule"msg:invalid formatted url" id:12;
#@MainRule "msg:invalidPOST format" id:13;
#@MainRule"msg:invalid POST boundary" id:14;

##################################
## SQL InjectionsIDs:1000-1099 ##
##################################
MainRule"rx:select|union|update|delete|insert|table|from|ascii|hex|unhex|drop""msg:sql keywords"
"mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie""s:$SQL:8" id:1000;
MainRule"str:\"" "msg:double quote""mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie" "s:$SQL:8,$XSS:8"id:1001;
MainRule"str:0x" "msg:0x, possible hex encoding""mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie" "s:$SQL:2" id:1002;
## Hardcore rules
MainRule"str:/*" "msg:mysql comment (/*)""mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie" "s:$SQL:8" id:1003;
MainRule"str:*/" "msg:mysql comment (*/)""mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie" "s:$SQL:8" id:1004;
MainRule "str:|""msg:mysql keyword (|)" "mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie" "s:$SQL:8"id:1005;
##MainRule"str:&&" "msg:mysql keyword (&&)""mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie" "s:$SQL:8" id:1006;
## end of hardcore rules
MainRule"str:--" "msg:mysql comment (--)""mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie" "s:$SQL:4" id:1007;
MainRule "str:;""msg:; in stuff" "mz:BODY|URL|ARGS""s:$SQL:4,$XSS:8" id:1008;
MainRule "str:=""msg:equal in var, probable sql/xss" "mz:ARGS|BODY""s:$SQL:2" id:1009;
MainRule "str:(""msg:parenthesis, probable sql/xss""mz:ARGS|URL|BODY|$HEADERS_VAR:Cookie" "s:$SQL:4,$XSS:8"id:1010;
MainRule "str:)""msg:parenthesis, probable sql/xss""mz:ARGS|URL|BODY|$HEADERS_VAR:Cookie" "s:$SQL:4,$XSS:8"id:1011;
MainRule "str:'""msg:simple quote" "mz:ARGS|BODY|URL|$HEADERS_VAR:Cookie""s:$SQL:4,$XSS:8" id:1013;
MainRule "str:,""msg:, in stuff" "mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie""s:$SQL:4" id:1015;
MainRule "str:#""msg:mysql comment (#)""mz:BODY|URL|ARGS|$HEADERS_VAR:Cookie" "s:$SQL:4" id:1016;
```

集成Naxsi模块

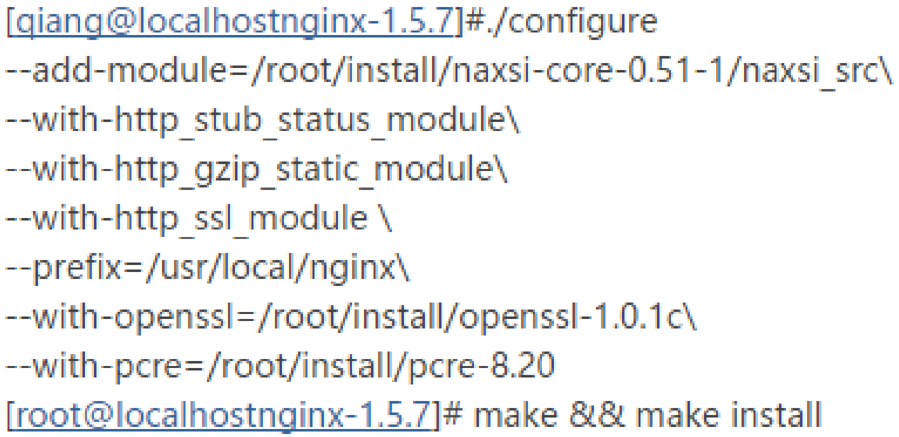
第四步：编译安装Nginx
查看系统原来编译Nginx的参数：



```
[qiang @srv-dfh526 ~]#nginx  -V
nginx version: nginx/1.3.0
TLS SNI support enabled
configure arguments:--with-http_stub_status_module --with-http_gzip_static_module--wi
openssl=/root/install/openssl-1.0.1c --with-pcre=/root/install/pcre-8.20
```

集成Naxsi模块

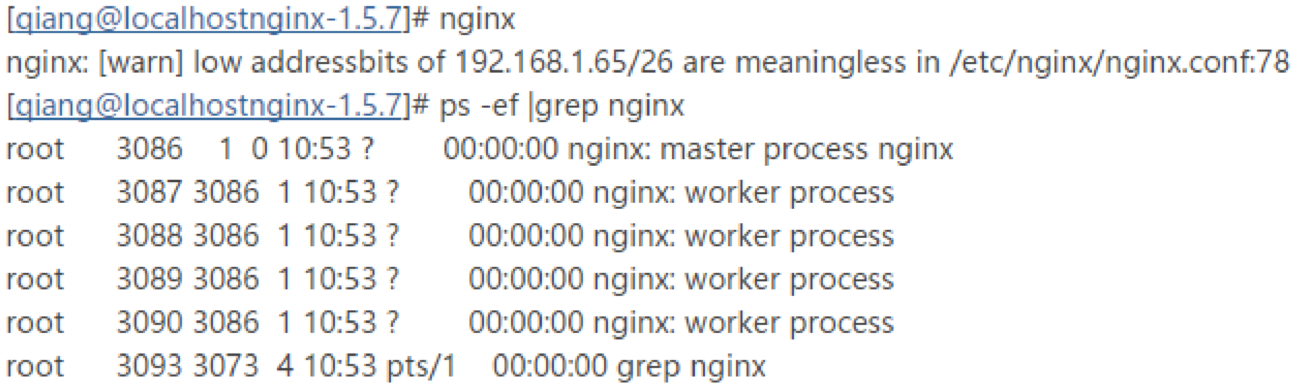
在原来的编译参数的首行加入--add-module=/root/install/naxsi-core-0.51-1/naxsi_src



```
[qiang@localhostnginx-1.5.7]#./configure
--add-module=/root/install/naxsi-core-0.51-1/naxsi_src\
--with-http_stub_status_module\
--with-http_gzip_static_module\
--with-http_ssl_module \
--prefix=/usr/local/nginx\
--with-openssl=/root/install/openssl-1.0.1c\
--with-pcre=/root/install/pcre-8.20
[root@localhostnginx-1.5.7]# make && make install
```

集成Naxsi模块

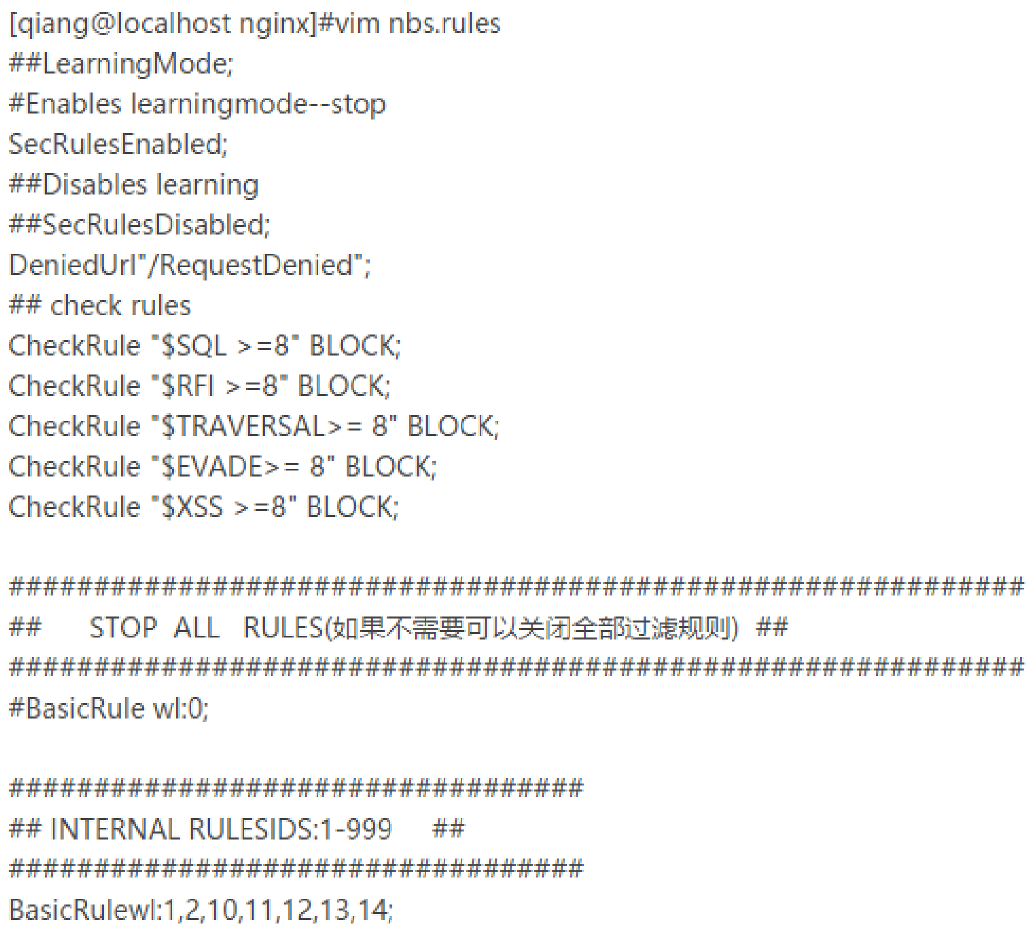
第五步：验证nginx是否安装成功



```
[qiang@localhostnginx-1.5.7]# nginx
nginx: [warn] low addressbits of 192.168.1.65/26 are meaningless in /etc/nginx/nginx.conf:78
[qiang@localhostnginx-1.5.7]# ps -ef |grep nginx
root      3086    1  0 10:53 ?        00:00:00 nginx: master process nginx
root      3087 3086  1 10:53 ?        00:00:00 nginx: worker process
root      3088 3086  1 10:53 ?        00:00:00 nginx: worker process
root      3089 3086  1 10:53 ?        00:00:00 nginx: worker process
root      3090 3086  1 10:53 ?        00:00:00 nginx: worker process
root      3093 3073  4 10:53 pts/1    00:00:00 grep nginx
```

集成Naxsi模块

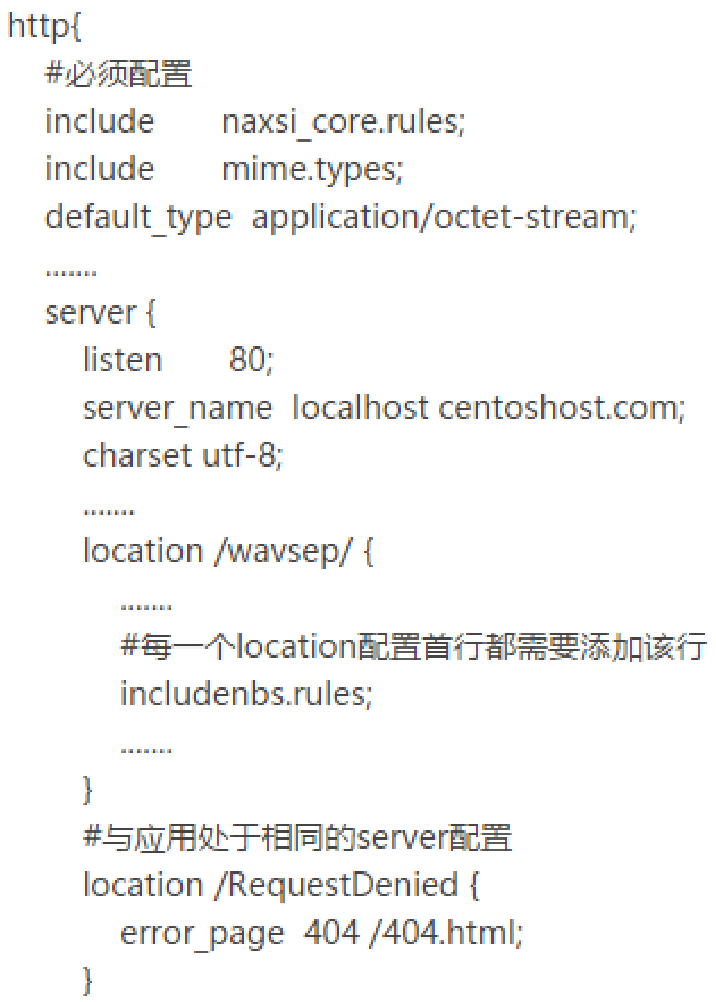
切换目录到与nginx.conf同目录下，新建nbs.rules文件



```
[qiang@localhost nginx]#vim nbs.rules
##LearningMode;
#Enables learningmode--stop
SecRulesEnabled;
##Disables learning
##SecRulesDisabled;
DeniedUrl"/RequestDenied";
## check rules
CheckRule "$SQL >=8" BLOCK;
CheckRule "$RFI >=8" BLOCK;
CheckRule "$TRAVERSAL>= 8" BLOCK;
CheckRule "$EVADE>= 8" BLOCK;
CheckRule "$XSS >=8" BLOCK;

############################################################
##      STOP  ALL   RULES(如果不需要可以关闭全部过滤规则)  ##
############################################################
#BasicRule wl:0;

##################################
## INTERNAL RULESIDS:1-999     ##
##################################
BasicRulewl:1,2,10,11,12,13,14;
```

集成Naxsi模块

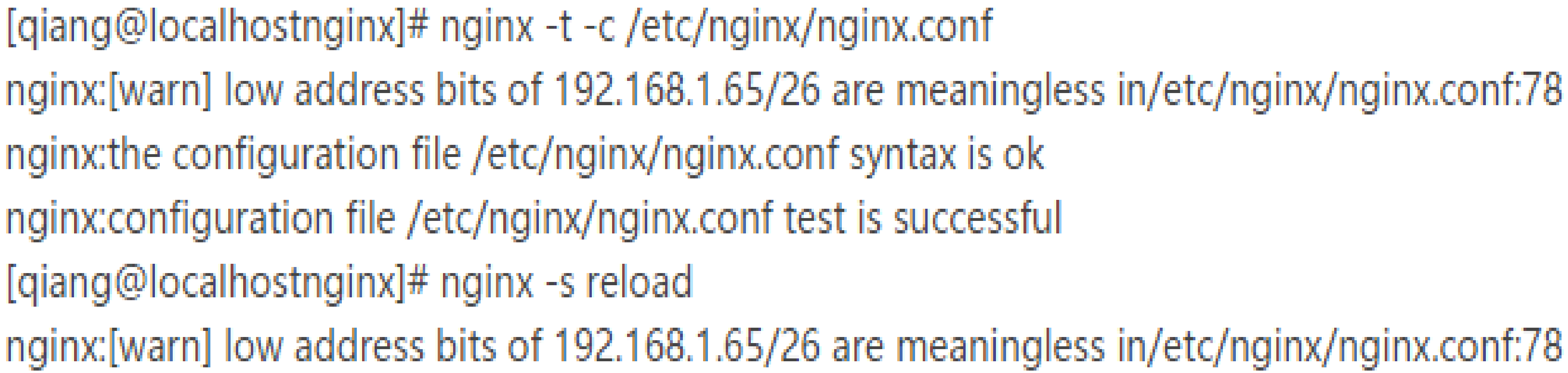
第七步：配置nginx.conf



```
http{
    #必须配置
    include       naxsi_core.rules;
    include       mime.types;
    default_type  application/octet-stream;
    .......
    server {
        listen       80;
        server_name  localhost centoshost.com;
        charset utf-8;
        .......
        location /wavsep/ {
            .......
            #每一个location配置首行都需要添加该行
            includenbs.rules;
            .......
        }
        #与应用处于相同的server配置
        location /RequestDenied {
            error_page  404 /404.html;
        }
```

集成Naxsi模块

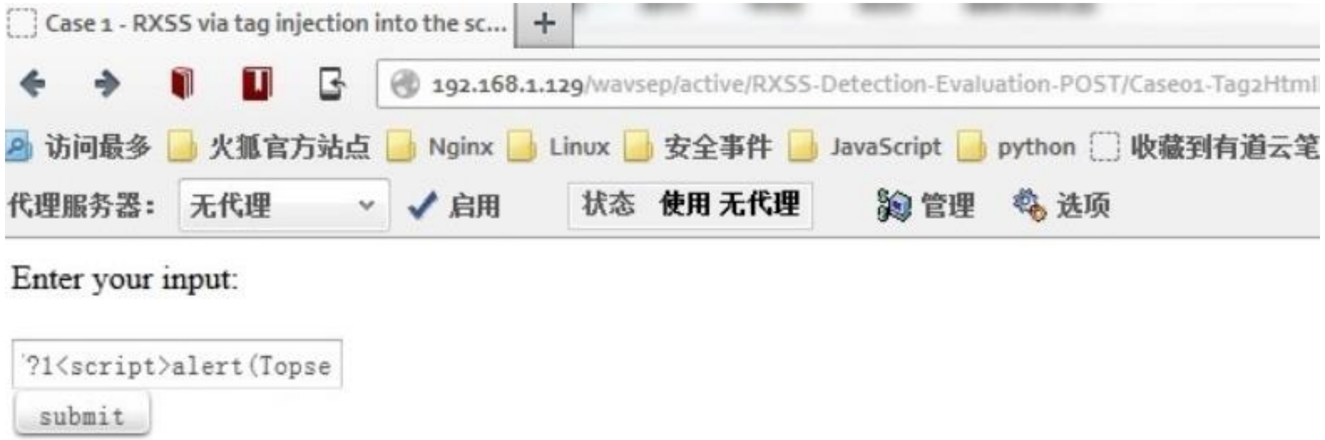
第八步：重启nginx



```
[qiang@localhostnginx]# nginx -t -c /etc/nginx/nginx.conf
nginx:[warn] low address bits of 192.168.1.65/26 are meaningless in/etc/nginx/nginx.conf:78
nginx:the configuration file /etc/nginx/nginx.conf syntax is ok
nginx:configuration file /etc/nginx/nginx.conf test is successful
[qiang@localhostnginx]# nginx -s reload
nginx:[warn] low address bits of 192.168.1.65/26 are meaningless in/etc/nginx/nginx.conf:78
```

集成Naxsi模块

第九步：测试拦截规则是否启用
上述的规则仅过滤“<”、“>”。
测试XSS注入



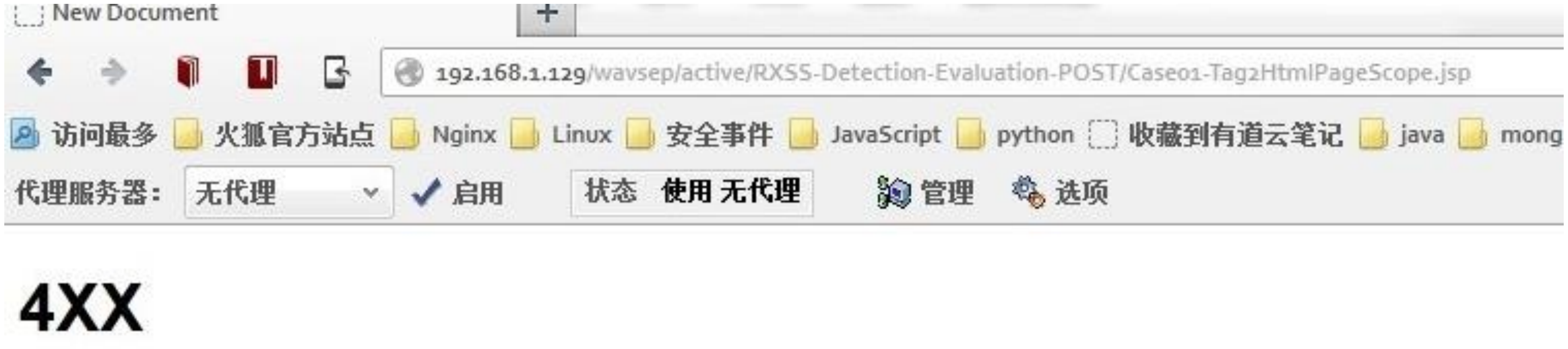
Case 1 - RXSS via tag injection into the sc...
+
192.168.1.129/wavsep/active/RXSS-Detection-Evaluation-POST/Case01-Tag2Html
访问最多
火狐官方站点
Nginx
Linux
安全事件
JavaScript
python
收藏到有道云笔
代理服务器：
无代理
启用
状态 使用 无代理
管理
选项
submit

Enter your input:

'?1<script>alert(Topse

集成Naxsi模块

结果：



4XX

THANK YOU