

Установка nginx на Ubuntu Server из ИСХОДНИКОВ

Дано: Ubuntu Server 10.04.1 LTS x64. Версия из пакета nginx-0.7.65-1ubuntu2, а в природе есть уже и постарше (на момент написания заметки была nginx-1.0.3).

Задача: установить nginx, переложить на него отдачу статического контента (картинок, документов и прочего).

Так как в репозитории слишком уж старая версия, будем ставить руками. Поехали.

Примечание: есть PPA здесь - <https://launchpad.net/~nginx/+archive/stable>

Установка оригинального пакета

Для начала надо установить оригинальный пакет - в нем есть ценный сценарий запуска и конфигурационные файлы (они помогут сократить время настройки). Затем его удаляем чтобы не мешался.

```
# sudo apt-get install nginx
# sudo apt-get remove nginx
```

Сборка из исходников

Проверяем зависимости, какие же пакеты нужны для сборки.

```
# sudo aptitude show nginx | grep Depends
Depends: libc6 (>= 2.4), libpcre3 (>= 7.7), libssl0.9.8 (>= 0.9.8k-1),
zlib1g
```

Ставим нужные пакеты:

```
# sudo apt-get install libc6-dev libpcre3-dev libssl-dev zlib1g-dev
```

Качаем nginx, разворачиваем, заходим в папку:

```
# mkdir /usr/src/nginx
# cd /usr/src/nginx
# wget http://sysoev.ru/nginx/nginx-1.0.3.tar.gz
# tar xvf nginx-1.0.3.tar.gz
# cd nginx-1.0.3/
```

Задаем некоторые опции для сборки. Пояснение: я люблю когда все лежит по местам и вообще «красота». Бинарник поместим в /usr/local/sbin чтобы он не пересекался с бинарником из

пакета, вдруг таковой будет случайно установлен. Ну и `-with-http_ssl_module` пригодится. Работать nginx будет у нас под стандартным пользователем `www-data:www-data`

```
# ./configure --sbin-path=/usr/local/sbin \
--conf-path=/etc/nginx/nginx.conf \
--error-log-path=/var/log/nginx/nginx-error.log \
--http-log-path=/var/log/nginx/nginx-http.log \
--pid-path=/var/run/nginx.pid \
--with-http_ssl_module \
--with-http_stub_status_module \
--with-http_gzip_static_module \
--user=www-data \
--group=www-data
```

Просмотрим отчет, убедимся что все правильно:

Configuration summary

- + using system PCRE library
- + using system OpenSSL library
- + md5: using OpenSSL library
- + sha1 library is not used
- + using system zlib library

nginx path prefix: `"/usr/local/nginx"`
nginx binary file: `"/usr/local/sbin"`
nginx configuration prefix: `"/etc/nginx"`
nginx configuration file: `"/etc/nginx/nginx.conf"`
nginx pid file: `"/var/run/nginx.pid"`
nginx error log file: `"/var/log/nginx/nginx-error.log"`
nginx http access log file: `"/var/log/nginx/nginx-http.log"`
nginx http client request body temporary files: `"client_body_temp"`
nginx http proxy temporary files: `"proxy_temp"`
nginx http fastcgi temporary files: `"fastcgi_temp"`
nginx http uwsgi temporary files: `"uwsgi_temp"`
nginx http scgi temporary files: `"scgi_temp"`

Собираем

```
# make
```

Устанавливаем

```
# sudo make install
```

NAXSI

NAXSI = **NGINX ANTI XSS & SQL INJECTION** Проще говоря, это файрвол веб-приложений (WAF) для NGINX, помогающий в защите от XSS, SQL-инъекций, CSRF, Local & Remote file inclusions. Отличительными особенностями его являются быстрота работы и простота настройки. Это

делает его хорошей альтернативой например mod_security и апачу.

Статья по теме: <http://habrahabr.ru/post/145241/>

Установка

Из репозиториев:

```
# sudo apt-get install nginx-naxsi
```

Из исходников:

```
# wget http://nginx.org/download/nginx-x.x.xx.tar.gz
# wget http://naxsi.googlecode.com/files/naxsi-x.xx.tar.gz
# tar xvzf nginx-x.x.xx.tar.gz
# tar xvzf naxsi-x.xx.tar.gz
# cd nginx-x.x.xx/
# ./configure --add-module=../naxsi-x.xx/naxsi_src/ [тут ваши опции для nginx]
# sudo make
# sudo make install
```

Настройка

Набор правил: http://code.google.com/p/naxsi/source/browse/trunk/naxsi_config/naxsi_core.rules

Раскомментируем в конфигурации nginx включение базовых запретительных правил

```
include /etc/nginx/naxsi_core.rules;
```

Теперь добавим в конфигурацию виртуального хоста желаемые настройки (рекомендую вынести их в отдельный файл и подключать через include):

```
LearningMode;
SecRulesEnabled;
DeniedUrl "/RequestDenied";

#include "/etc/nginx/mynaxsi.rules";

## check rules
CheckRule "$SQL >= 8" BLOCK;
CheckRule "$RFI >= 8" BLOCK;
CheckRule "$TRAVERSAL >= 4" BLOCK;
CheckRule "$EVADE >= 4" BLOCK;
CheckRule "$XSS >= 8" BLOCK;
```

Разберем подробнее, что значат эти команды:

- LearningMode — включен режим обучения. Запросы не блокируются, формируется вайт-лист.
- SecRulesEnabled — NAXSI включен для данной локации. Если захотим выключить для другой локации (например, защищенной внутренней зоны), то делаем в ней SecRulesDisabled.
- DeniedURL — URL редиректа для запрещенных запросов.
- CheckRule — проверка “штрафных очков” запроса по категориям.
- /etc/nginx/mynaxsi.rules — сгенерированные правила (пока не сгенерили — закомментированные).

Подготавливаемся к запуску

Все, сборка закончена, теперь надо подготовиться к запуску

Создаем папку для логов:

```
# sudo mkdir /var/log/nginx
```

Правим скрипт запуска: в файле /etc/init.d/nginx меняем строку DAEMON на

```
DAEMON=/usr/local/sbin/nginx
```

Все, можно запускаться и пробовать:

```
# sudo service nginx start
```

Если конфигурационный файл лежит на месте, все должно заработать.

Конфигурируем

Преамбула: у меня живет несколько виртуальных хостов в Апаче, это необходимо учесть при настройке. Конфигурационные файлы живут в /etc/apache2/sites-available - так проще их настраивать.

Для того, чтобы в логах Апача писались реальные IP-адреса, а не 127.0.0.1 (на котором он будет висеть), требуется установить модуль rpf:

```
# sudo apt-get install libapache2-mod-rpaf
# sudo a2enmod rpaf
```

Проверяем что у нас в /etc/apache2/mods_available/rpaf.conf

```
# cat rpaf.conf
<IfModule mod_rpaf.c>
RPAFenable On
RPAFsethostname On
#RPAFproxy_ips 127.0.0.1
```

```
RPAFproxy_ips 127.0.0.1 realip [realip1]
</IfModule>
```

Объяснялки: Для того, чтобы прикрутить несколько виртуалхостов апача делается следующее:

- Каждый виртуалхост апача должен висеть на адресе 127.0.0.1:номер_порта, именно туда и подключается nginx, получив запрос
- В /etc/apache2/ports.conf заводится требуемое кол-во портов (по 1 на виртуалхост)
- В конфиге каждого виртуалхоста при помощи директивы VirtualHost выставляется требуемый адрес и порт
- В конфиге nginx прописывается каждый домен

Примеры - внимание, это очень простые болванки, а не рабочие файлы!

/etc/apache2/ports.conf :

```
#Listen 80
Listen 127.0.0.1:8081
Listen 127.0.0.1:8082
```

/etc/apache2/sites-available/site1 :

```
<VirtualHost 127.0.0.1:8081>
# Это нужно если стоит модуль mpm_itk_module - для запуска каждого виртуалхоста
# под своим пользователем
<IfModule mpm_itk_module>
    AssignUserId sitelcom sitelcom
</IfModule>

    ServerName sitel.com
    ServerAlias www.sitel.com
    ServerAdmin admin@sitel.com

    DocumentRoot /var/www/sitel.com
    <Directory /var/www/sitel.com>
        Options FollowSymLinks MultiViews
        AllowOverride All
        Order allow,deny
        allow from all
    </Directory>

    ErrorLog /var/log/apache2/sitel.com-error.log
    CustomLog /var/log/apache2/sitel.com-access.log combined

    # Possible values include: debug, info, notice, warn, error, crit,
    # alert, emerg.
    LogLevel warn
</VirtualHost>
```

/etc/apache2/sites-available/site2 :

```
<VirtualHost 127.0.0.1:8082>
# Это нужно если стоит модуль mpm_itk_module - для запуска каждого виртуалхоста
# под своим пользователем
<IfModule mpm_itk_module>
    AssignUserId site2com site2com
</IfModule>

    ServerName site2.com
    ServerAlias www.site2.com
    ServerAdmin admin@site2.com

    DocumentRoot /var/www/site2.com
    <Directory /var/www/site2.com>
        Options FollowSymLinks MultiViews
        AllowOverride All
        Order allow,deny
        allow from all
    </Directory>

    ErrorLog /var/log/apache2/site2.com-error.log
    CustomLog /var/log/apache2/site2.com-access.log combined

    # Possible values include: debug, info, notice, warn, error, crit,
    # alert, emerg.
    LogLevel warn
</VirtualHost>
```

/etc/nginx/nginx.conf :

```
user www-data;
worker_processes 3;

#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;
tcp_nodelay on;

#keepalive_timeout 0;
keepalive_timeout 65;

gzip on;
gzip_min_length 1000;
gzip_buffers 16 8k;
gzip_proxied expired no-cache no-store private auth;
gzip_types text/plain application/xml application/x-pointplus
application/x-javascript text/css;
gzip_comp_level 5;
gzip_disable "MSIE [1-6]\.(?!.*SV1)"; # do not compress for IE1-6

# sitel.com
server {
    listen 80;
    server_name www.sitel.com sitel.com;

    #charset koi8-r;

    #access_log logs/host.access.log main;

    location / {
        proxy_pass http://127.0.0.1:8081/;
        proxy_redirect off;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_headers_hash_max_size 1024;
        proxy_headers_hash_bucket_size 256;
        client_max_body_size 40m;
        client_body_buffer_size 256k;

        proxy_connect_timeout 120;
        proxy_send_timeout 120;
        proxy_read_timeout 120;

        proxy_buffer_size 64k;
        proxy_buffers 4 64k;
        proxy_busy_buffers_size 64k;
        proxy_temp_file_write_size 64k;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        # Отдача картинок
        # 1. Если картинка лежит в папке product, забрать ее через Apache - там
```

```
скриптом наложится watermark
    location ~* "product"
    {
        root /var/www/site1.com;
        proxy_pass http://127.0.0.1:8081;
        proxy_redirect off;
        access_log off;
    }
    # 2. Все остальные картинки отдаем напрямую
    location ~*
    \.(jpg|jpeg|gif|png|ico|zip|tar|tgz|gz|bz2|rar|doc|xls|exe|pdf|ppt|txt)$
    {
        root /var/www/site1.com;
        access_log off;
    }
}
# site2.com
server {
    listen 80;
    server_name www.site2.com site2.com;

    #charset koi8-r;

    #access_log logs/host.access.log main;

    location / {
        proxy_pass http://127.0.0.1:8082/;
        proxy_redirect off;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_headers_hash_max_size 1024;
        proxy_headers_hash_bucket_size 256;
        client_max_body_size 40m;
        client_body_buffer_size 256k;

        proxy_connect_timeout 120;
        proxy_send_timeout 120;
        proxy_read_timeout 120;

        proxy_buffer_size 64k;
        proxy_buffers 4 64k;
        proxy_busy_buffers_size 64k;
        proxy_temp_file_write_size 64k;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        location ~*
        \.(jpg|jpeg|gif|png|ico|zip|tar|tgz|gz|bz2|rar|doc|xls|exe|pdf|ppt|txt)$
        {
            root /var/www/site1.com;
            access_log off;
        }
    }
}
```



```
}  
}  
}
```

Ссылки

- Домашняя страничка nginx - <http://sysoev.ru/>

From:
<https://wiki.rtzra.ru/> - RTzRa's hive

Permanent link:
<https://wiki.rtzra.ru/software/nginx/installation>

Last update: **2017/05/09 18:34**

