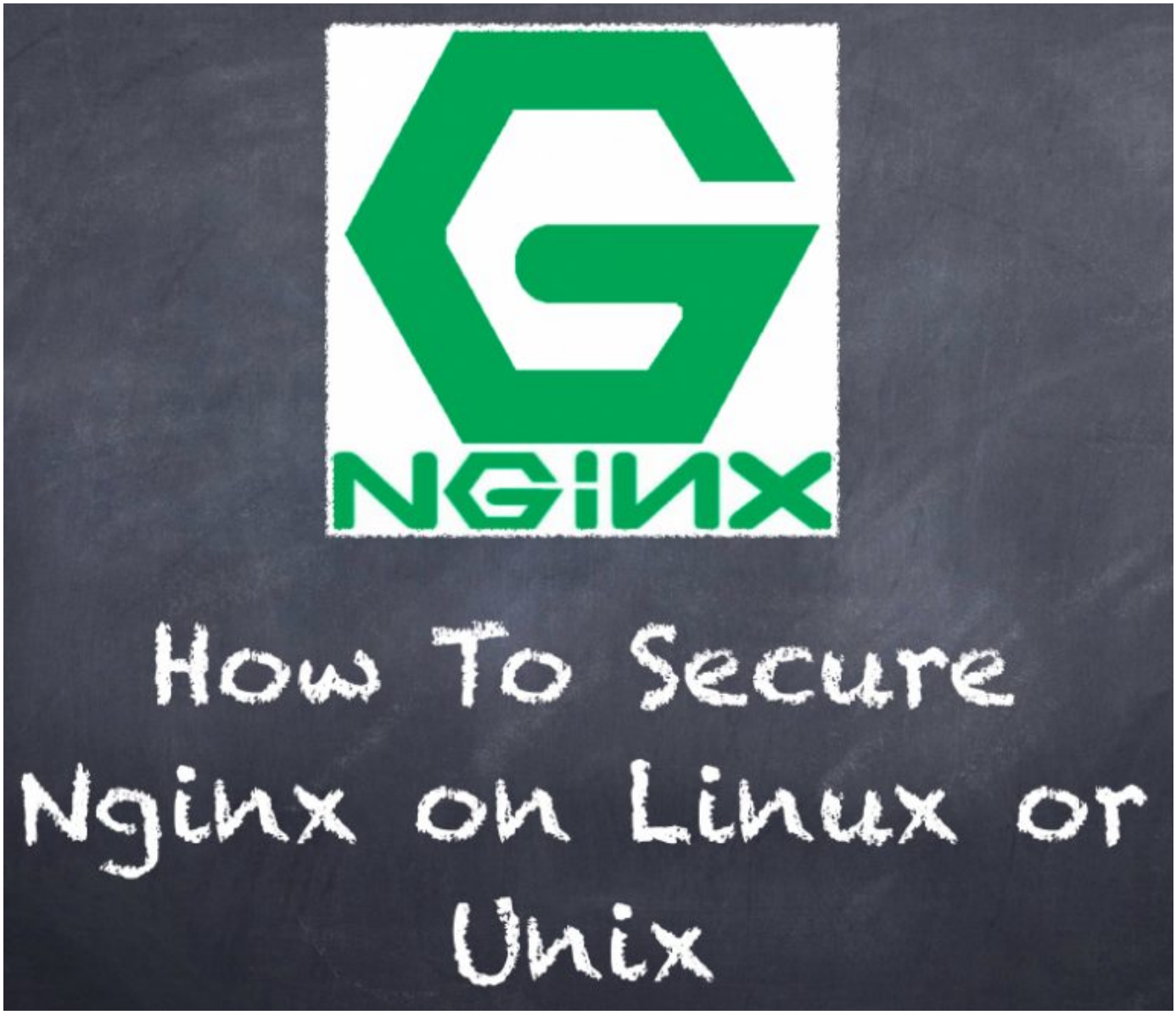
nixCraft
Linux Tips, Hacks, Tutorials, And Ideas In Blog Format

Top 25 Nginx Web Server Best Security Practices

last updated September 19, 2017 **in Debian Linux, Howto, Linux, RedHat/Fedora Linux, Security, Sys admin, Ubuntu Linux, UNIX**

Nginx is a lightweight, high-performance web server/reverse proxy and e-mail (IMAP/POP3) proxy. It runs on UNIX, GNU/Linux, BSD variants, Mac OS X, Solaris, and Microsoft Windows. According to Netcraft, 13.50% of all domains on the Internet use nginx web server. Nginx is one of a handful of servers written to address the C10K problem. Unlike traditional servers, Nginx doesn't rely on threads to handle requests. Instead, it uses a much more scalable event-driven (asynchronous) architecture. Nginx powers several high traffic web sites, such as WordPress, Hulu, Github, and SourceForge.



This page collects hints how to improve the security of nginx web servers running on Linux or UNIX-like operating systems.

Default Config Files and Nginx Port

- `/usr/local/nginx/conf/` or `/etc/nginx/`– The nginx server configuration directory and /usr/local/nginx/conf/nginx.conf is main configuration file.
- `/usr/local/nginx/html/` or `/var/www/html`– The default document location.
- `/usr/local/nginx/logs/` or `/var/log/nginx` – The default log file location.
- Nginx **HTTP default port** : TCP 80
- Nginx **HTTPS default port** : TCP 443

You can test nginx configuration changes as follows:

```
# /usr/local/nginx/sbin/nginx -t
```

OR

```
# nginx -t
```

Sample outputs:

```
the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok
configuration file /usr/local/nginx/conf/nginx.conf test is successful
```

To load config changes, type:

```
# /usr/local/nginx/sbin/nginx -s reload
```

OR

```
# nginx -s reload
```

To stop server, type:

```
# /usr/local/nginx/sbin/nginx -s stop
```

OR

```
# nginx -s stop
```

#1: Turn On SELinux

Security-Enhanced Linux (SELinux) is a Linux kernel feature that provides a mechanism for supporting access control security policies which provides great protection. It can stop many attacks before your system rooted. See how to turn on SELinux for CentOS / RHEL based systems.

Do Boolean Lockdown

Run the getsebool -a command and lockdown system:

```
getsebool -a | less
getsebool -a | grep off
getsebool -a | grep on
```

To secure the machine, look at settings which are set to 'on' and change to 'off' if they do not apply to your setup with the help of setsebool command. Set correct SE Linux booleans to maintain functionality and protection. Please note that SELinux adds 2-8% overheads to typical RHEL or CentOS installation.

#2: Allow Minimal Privileges Via Mount Options

Server all your webpages / html / php files via separate partitions. For example, create a partition called /dev/sda5 and mount at the /nginx. Make sure /nginx is mounted with noexec, nodev and nosetuid permissions. Here is my /etc/fstab entry for mounting /nginx:

```
LABEL=/nginx     /nginx          ext3    defaults,nosuid,noexec,nodev 1 2
```

Note you need to create a new partition using fdisk and mkfs.ext3 commands.

#3: Linux /etc/sysctl.conf Hardening

You can control and configure Linux kernel and networking settings via /etc/sysctl.conf.

```
# Avoid a smurf attack
net.ipv4.icmp_echo_ignore_broadcasts = 1

# Turn on protection for bad icmp error messages
net.ipv4.icmp_ignore_bogus_error_responses = 1

# Turn on syncookies for SYN flood attack protection
net.ipv4.tcp_syncookies = 1

# Turn on and log spoofed, source routed, and redirect packets
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.default.log_martians = 1

# No source routed packets here
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.default.accept_source_route = 0

# Turn on reverse path filtering
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1

# Make sure no one can alter the routing tables
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
net.ipv4.conf.all.secure_redirects = 0
net.ipv4.conf.default.secure_redirects = 0

# Don't act as a router
net.ipv4.ip_forward = 0
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0


# Turn on execshild
kernel.exec-shield = 1
kernel.randomize_va_space = 1

# Tuen IPv6
net.ipv6.conf.default.router_solicitations = 0
net.ipv6.conf.default.accept_ra_rtr_pref = 0
net.ipv6.conf.default.accept_ra_pinfo = 0
net.ipv6.conf.default.accept_ra_defrtr = 0
net.ipv6.conf.default.autoconf = 0
```

```
net.ipv6.conf.default.dad_transmits = 0
net.ipv6.conf.default.max_addresses = 1

# Optimization for port usefor LBs
# Increase system file descriptor limit
fs.file-max = 65535

# Allow for more PIDs (to reduce rollover problems); may break some programs 32768
kernel.pid_max = 65536

# Increase system IP port limits
net.ipv4.ip_local_port_range = 2000 65000

# Increase TCP max buffer size setable using setsockopt()
net.ipv4.tcp_rmem = 4096 87380 8388608
net.ipv4.tcp_wmem = 4096 87380 8388608

# Increase Linux auto tuning TCP buffer limits
# min, default, and max number of bytes to use
# set max to at least 4MB, or higher if you use very high BDP paths
# Tcp Windows etc
net.core.rmem_max = 8388608
net.core.wmem_max = 8388608
net.core.netdev_max_backlog = 5000
net.ipv4.tcp_window_scaling = 1
```

See also:

- Linux Tuning The VM (memory) Subsystem
- Linux Tune Network Stack (Buffers Size) To Increase Networking Performance

#4: Remove All Unwanted Nginx Modules

You need to minimizes the number of modules that are compiled directly into the nginx binary. This minimizes risk by limiting the capabilities allowed by the webserver. You can configure and install nginx using only required modules. For example, disable SSI and autoindex module you can type:

```
# ./configure --without-http_autoindex_module --without-http_ssi_module
# make
# make install
```

Type the following command to see which modules can be turn on or off while compiling nginx server:

```
# ./configure --help | less
```

Disable nginx modules that you don't need.

(Optional) Change Nginx Version Header

Edit src/http/ngx_http_header_filter_module.c, enter:

```
# vi +48 src/http/ngx_http_header_filter_module.c
```

Find line

```
static char ngx_http_server_string[] = "Server: nginx" CRLF;
static char ngx_http_server_full_string[] = "Server: " NGINX_VER CRLF;
```

Change them as follows:

```
static char ngx_http_server_string[] = "Server: Ninja Web Server" CRLF;
static char ngx_http_server_full_string[] = "Server: Ninja Web Server" CRLF;
```

Save and close the file. Now, you can compile the server. Add the following in nginx.conf to turn off nginx version number displayed on all auto generated error pages:

```
server_tokens off
```

#5: Use mod_security (only for backend Apache servers)

mod_security provides an application level firewall for Apache. Install mod_security for all backend Apache web servers. This will stop many injection attacks.

#6: Install SELinux Policy To Harden The Nginx Webserver

By default SELinux will not protect the nginx web server. However, you can install and compile protection as follows. First, install required SELinux compile time support:

```
# yum -y install selinux-policy-targeted selinux-policy-devel
```

Download targeted SELinux policies to harden the nginx webserver on Linux servers from the project home page:

```
# cd /opt
# wget 'http://downloads.sourceforge.net/project/selinuxnginx/se-ngix_1_0_10.tar.gz?
```

```
use_mirror=nchc'
```

Untar the same:

```
# tar -zxvf se-ngix_1_0_10.tar.gz
```

Compile the same

```
# cd se-ngix_1_0_10/nginx
# make
```

Sample outputs:

```
Compiling targeted nginx module
/usr/bin/checkmodule:  loading policy configuration from tmp/nginx.tmp
/usr/bin/checkmodule:  policy configuration loaded
/usr/bin/checkmodule:  writing binary representation (version 6) to tmp/nginx.mod
Creating targeted nginx.pp policy package
rm tmp/nginx.mod.fc tmp/nginx.mod
```

Install the resulting nginx.pp SELinux module:

```
# /usr/sbin/semodule -i nginx.pp
```

#7: Restrictive Iptables Based Firewall

The following firewall script blocks everything and only allows:

- Incoming HTTP (TCP port 80) requests
- Incoming ICMP ping requests
- Outgoing ntp (port 123) requests
- Outgoing smtp (TCP port 25) requests

```
#!/bin/bash
IPT="/sbin/iptables"

#### IPS ######
# Get server public ip
SERVER_IP=$(ifconfig eth0 | grep 'inet addr:' | awk -F'inet addr:' '{ print $2}' | awk '
LB1_IP="204.54.1.1"
```

```
LB2_IP="204.54.1.2"

# Do some smart logic so that we can use damm script on LB2 too
OTHER_LB=""
SERVER_IP=""
[[ "$SERVER_IP" == "$LB1_IP" ]] && OTHER_LB="$LB2_IP" || OTHER_LB="$LB1_IP"
[[ "$OTHER_LB" == "$LB2_IP" ]] && OPP_LB="$LB1_IP" || OPP_LB="$LB2_IP"

### IPs ###
PUB_SSH_ONLY="122.xx.yy.zz/29"

#### FILES #####
BLOCKED_IP_TDB=/root/.fw/blocked.ip.txt
SPOOFIP="127.0.0.0/8 192.168.0.0/16 172.16.0.0/12 10.0.0.0/8 169.254.0.0/16 0.0.0.0/8 24
BADIPS=$( [[ -f ${BLOCKED_IP_TDB} ]] && egrep -v "^#|^$" ${BLOCKED_IP_TDB})

### Interfaces ###
PUB_IF="eth0"   # public interface
LO_IF="lo"      # loopback
VPN_IF="eth1"   # vpn / private net

### start firewall ###
echo "Setting LB1 $(hostname) Firewall..."

# DROP and close everything
$IPT -P INPUT DROP
$IPT -P OUTPUT DROP
$IPT -P FORWARD DROP

# Unlimited lo access
$IPT -A INPUT -i ${LO_IF} -j ACCEPT
$IPT -A OUTPUT -o ${LO_IF} -j ACCEPT

# Unlimited vpn / pnet access
$IPT -A INPUT -i ${VPN_IF} -j ACCEPT
$IPT -A OUTPUT -o ${VPN_IF} -j ACCEPT

# Drop sync
$IPT -A INPUT -i ${PUB_IF} -p tcp ! --syn -m state --state NEW -j DROP

# Drop Fragments
$IPT -A INPUT -i ${PUB_IF} -f -j DROP

$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags ALL FIN,URG,PSH -j DROP
$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags ALL ALL -j DROP

# Drop NULL packets
$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags ALL NONE -m limit --limit 5/m --limit-bur
$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags ALL NONE -j DROP

$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags SYN,RST SYN,RST -j DROP

# Drop XMAS
$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags SYN,FIN SYN,FIN -m limit --limit 5/m --li
$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP

# Drop FIN packet scans
$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags FIN,ACK FIN -m limit --limit 5/m --limit-
$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags FIN,ACK FIN -j DROP

$IPT  -A INPUT -i ${PUB_IF} -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j DROP
```

```
# Log and get rid of broadcast / multicast and invalid
$IPT  -A INPUT -i ${PUB_IF} -m pkttype --pkt-type broadcast -j LOG --log-prefix " Broadc
$IPT  -A INPUT -i ${PUB_IF} -m pkttype --pkt-type broadcast -j DROP

$IPT  -A INPUT -i ${PUB_IF} -m pkttype --pkt-type multicast -j LOG --log-prefix " Multic
$IPT  -A INPUT -i ${PUB_IF} -m pkttype --pkt-type multicast -j DROP

$IPT  -A INPUT -i ${PUB_IF} -m state --state INVALID -j LOG --log-prefix " Invalid "
$IPT  -A INPUT -i ${PUB_IF} -m state --state INVALID -j DROP

# Log and block spoofed ips
$IPT -N spooflist
for ipblock in $SPOOFIP
do
         $IPT -A spooflist -i ${PUB_IF} -s $ipblock -j LOG --log-prefix " SPOOF List Blo
         $IPT -A spooflist -i ${PUB_IF} -s $ipblock -j DROP
done
$IPT -I INPUT -j spooflist
$IPT -I OUTPUT -j spooflist
$IPT -I FORWARD -j spooflist

# Allow ssh only from selected public ips
for ip in ${PUB_SSH_ONLY}
do
        $IPT -A INPUT -i ${PUB_IF} -s ${ip} -p tcp -d ${SERVER_IP} --destination-port 22
        $IPT -A OUTPUT -o ${PUB_IF} -d ${ip} -p tcp -s ${SERVER_IP} --sport 22 -j ACCEPT
done

# allow incoming ICMP ping pong stuff
$IPT -A INPUT -i ${PUB_IF} -p icmp --icmp-type 8 -s 0/0 -m state --state NEW,ESTABLISHED
$IPT -A OUTPUT -o ${PUB_IF} -p icmp --icmp-type 0 -d 0/0 -m state --state ESTABLISHED,RE

# allow incoming HTTP port 80
$IPT -A INPUT -i ${PUB_IF} -p tcp -s 0/0 --sport 1024:65535 --dport 80 -m state --state
$IPT -A OUTPUT -o ${PUB_IF} -p tcp --sport 80 -d 0/0 --dport 1024:65535 -m state --state


# allow outgoing ntp
$IPT -A OUTPUT -o ${PUB_IF} -p udp --dport 123 -m state --state NEW,ESTABLISHED -j ACCEP
$IPT -A INPUT -i ${PUB_IF} -p udp --sport 123 -m state --state ESTABLISHED -j ACCEPT

# allow outgoing smtp
$IPT -A OUTPUT -o ${PUB_IF} -p tcp --dport 25 -m state --state NEW,ESTABLISHED -j ACCEPT
$IPT -A INPUT -i ${PUB_IF} -p tcp --sport 25 -m state --state ESTABLISHED -j ACCEPT

### add your other rules here ####

########################
# drop and log everything else
$IPT -A INPUT -m limit --limit 5/m --limit-burst 7 -j LOG --log-prefix " DEFAULT DROP "
$IPT -A INPUT -j DROP

exit 0
```

See the following tutorials for more info:

1. How to setup a UFW firewall on Ubuntu 16.04 LTS server

2. CentOS / Redhat Iptables Firewall Configuration Tutorial
3. Linux: 20 Iptables Examples For New SysAdmins

#8: Controlling Buffer Overflow Attacks

Edit nginx.conf and set the buffer size limitations for all clients.

```
# vi /usr/local/nginx/conf/nginx.conf
```

Edit and set the buffer size limitations for all clients as follows:

```
## Start: Size Limits & Buffer Overflows ##
 client_body_buffer_size  1K;
 client_header_buffer_size 1k;
 client_max_body_size 1k;
 large_client_header_buffers 2 1k;
## END: Size Limits & Buffer Overflows ##
```

Where,

1. **client_body_buffer_size 1k** – (default is 8k or 16k) The directive specifies the client request body buffer size.
2. **client_header_buffer_size 1k** – Directive sets the headerbuffer size for the request header from client. For the overwhelming majority of requests a buffer size of 1K is sufficient. Increase this if you have a custom header or a large cookie sent from the client (e.g., wap client).
3. **client_max_body_size 1k**– Directive assigns the maximum accepted body size of client request, indicated by the line Content-Length in the header of request. If size is greater the given one, then the client gets the error “Request Entity Too Large” (413). Increase this when you are getting file uploads via the POST method.
4. **large_client_header_buffers 2 1k** – Directive assigns the maximum number and size of buffers for large headers to read from client request. By default the size of one buffer is equal to the size of page, depending on platform this either 4K or 8K, if at the end of working request connection converts to state keep-alive, then these buffers are freed. 2x1k will accept 2kB data URI. This will also help combat bad bots and DoS attacks.

You also need to control timeouts to improve server performance and cut clients. Edit it as follows:

```
 ## Start: Timeouts ##
  client_body_timeout   10;
  client_header_timeout 10;
  keepalive_timeout     5 5;
  send_timeout          10;
## End: Timeouts ##
```

1. **client_body_timeout 10;** – Directive sets the read timeout for the request body from client. The timeout is set only if a body is not get in one readstep. If after this time the client send nothing, nginx returns error “Request time out” (408). The default is 60.
2. **client_header_timeout 10;** – Directive assigns timeout with reading of the title of the request of client. The timeout is set only if a header is not get in one readstep. If after this time the client send nothing, nginx returns error “Request time out” (408).
3. **keepalive_timeout 5 5;** – The first parameter assigns the timeout for keep-alive connections with the client. The server will close connections after this time. The optional second parameter assigns the time value in the header Keep-Alive: timeout=time of the response. This header can convince some browsers to close the connection, so that the server does not have to. Without this parameter, nginx does not send a Keep-Alive header (though this is not what makes a connection “keep-alive”).
4. **send_timeout 10;** – Directive assigns response timeout to client. Timeout is established not on entire transfer of answer, but only between two operations of reading, if after this time client will take nothing, then nginx is shutting down the connection.

#9: Control Simultaneous Connections

You can use NginxHttpLimitZone module to limit the number of simultaneous connections for the assigned session or as a special case, from one IP address. Edit nginx.conf:

```
### Directive describes the zone, in which the session states are stored i.e. store in sl
### 1m can handle 32000 sessions with 32 bytes/session, set to 5m x 32000 session ###
       limit_zone slimits $binary_remote_addr 5m;

### Control maximum number of simultaneous connections for one session i.e. ###
### restricts the amount of connections from a single ip address ###
        limit_conn slimits 5;
```

The above will limits remote clients to no more than 5 concurrently “open” connections per remote ip address.

#10: Allow Access To Our Domain Only

If bot is just making random server scan for all domains, just deny it. You must only allow configured virtual domain or reverse proxy requests. You don’t want to display request using an IP address:

```
## Only requests to our Host are allowed i.e. nixcraft.in, images.nixcraft.in and w
      if ($host !~ ^(nixcraft.in|www.nixcraft.in|images.nixcraft.in)$ ) {
         return 444;
      }
##
```

#11: Limit Available Methods

GET and POST are the most common methods on the Internet. Web server methods are defined in RFC 2616. If a web server does not require the implementation of all available methods, they should be disabled. The following will filter and only allow GET, HEAD and POST methods:

```
## Only allow these request methods ##
     if ($request_method !~ ^(GET|HEAD|POST)$ ) {
         return 444;
     }
## Do not accept DELETE, SEARCH and other methods ##
```

More About HTTP Methods

- The GET method is used to request document such as http://www.cyberciti.biz/index.php.
- The HEAD method is identical to GET except that the server MUST NOT return a message-body in the response.
- The POST method may involve anything, like storing or updating data, or ordering a product, or sending E-mail by submitting the form. This is usually processed using the server side scripting such as PHP, PERL, Python and so on. You must use this if you want to upload files and process forms on server.

#12: How Do I Deny Certain User-Agents?

You can easily block user-agents i.e. scanners, bots, and spammers who may be abusing your server.

```
## Block download agents ##
     if ($http_user_agent ~* LWP::Simple|BBBike|wget) {
            return 403;
     }
##
```

Block robots called msnbot and scrapbot:

```
## Block some robots ##
     if ($http_user_agent ~* msnbot|scrapbot) {
            return 403;
     }
```

#12: How Do I Block Referral Spam?

Referer spam is dengerouns. It can harm your SEO ranking via web-logs (if published) as referer field refer to their spammy site. You can block access to referer spammers with these lines.

```
## Deny certain Referers ###
     if ( $http_referer ~* (babes|forsale|girl|jewelry|love|nudit|organic|poker|por
     {
         # return 404;
         return 403;
     }
##
```

#13: How Do I Stop Image Hotlinking?

Image or HTML hotlinking means someone makes a link to your site to one of your images, but displays it on their own site. The end result you will end up paying for bandwidth bills and make the content look like part of the hijacker's site. This is usually done on forums and blogs. I strongly suggest you block and stop image hotlinking at your server level itself.

```
# Stop deep linking or hot linking
location /images/ {
  valid_referers none blocked www.example.com example.com;
   if ($invalid_referer) {
     return   403;
   }
}
```

Example: Rewrite And Display Image

Another example with link to banned image:

```
valid_referers blocked www.example.com example.com;
 if ($invalid_referer) {
  rewrite ^/images/uploads.*\.(gif|jpg|jpeg|png)$ http://www.examples.com/banned.jp
 }
```

See also:

- HowTo: Use nginx map to block image hotlinking. This is useful if you want to block tons of domains.

#14: Directory Restrictions

You can set access control for a specified directory. All web directories should be configured on a case-by-case basis, allowing access only where needed.

Limiting Access By Ip Address

You can limit access to directory by ip address to /docs/ directory:

```
location /docs/ {
  ## block one workstation
  deny    192.168.1.1;

  ## allow anyone in 192.168.1.0/24
  allow   192.168.1.0/24;

  ## drop rest of the world
  deny    all;
}
```

Password Protect The Directory

First create the password file and add a user called vivek:

```
# mkdir /usr/local/nginx/conf/.htpasswd/
# htpasswd -c /usr/local/nginx/conf/.htpasswd/passwd vivek
```

Edit nginx.conf and protect the required directories as follows:

```
### Password Protect /personal-images/ and /delta/ directories ###
location ~ /(personal-images/.*|delta/.*) {
  auth_basic  "Restricted";
  auth_basic_user_file   /usr/local/nginx/conf/.htpasswd/passwd;
}
```

Once a password file has been generated, subsequent users can be added with the following command:

```
# htpasswd -s /usr/local/nginx/conf/.htpasswd/passwd userName
```

#15: Nginx SSL Configuration

HTTP is a plain text protocol and it is open to passive monitoring. You should use SSL to to encrypt your content for users:

1. HowTo: Create a Self-Signed SSL Certificate on Nginx For CentOS / RHEL
2. How to configure Nginx with free Let's Encrypt SSL certificate on Debian or Ubuntu Linux
3. How to install Letsencrypt free SSL/TLS for Nginx certificate on Alpine Linux
4. How to configure Nginx SSL/TLS passthrough with TCP load balancing
5. nginx: Setup SSL Reverse Proxy (Load Balanced SSL Proxy)

#16: Nginx And PHP Security Tips

PHP is one of the popular server side scripting language. Edit /etc/php.ini as follows:

```
# Disallow dangerous functions
disable_functions = phpinfo, system, mail, exec

## Try to limit resources  ##

# Maximum execution time of each script, in seconds
max_execution_time = 30

# Maximum amount of time each script may spend parsing request data
max_input_time = 60

# Maximum amount of memory a script may consume (8MB)
memory_limit = 8M

# Maximum size of POST data that PHP will accept.
post_max_size = 8M

# Whether to allow HTTP file uploads.
file_uploads = Off

# Maximum allowed size for uploaded files.
upload_max_filesize = 2M

# Do not expose PHP error messages to external users
display_errors = Off

# Turn on safe mode
safe_mode = On

# Only allow access to executables in isolated directory
safe_mode_exec_dir = php-required-executables-path

# Limit external access to PHP environment
safe_mode_allowed_env_vars = PHP_

# Restrict PHP information leakage
expose_php = Off

# Log all errors
log_errors = On

# Do not register globals for input data
register_globals = Off
```

```
# Minimize allowable PHP post size
post_max_size = 1K

# Ensure PHP redirects appropriately
cgi.force_redirect = 0

# Disallow uploading unless necessary
file_uploads = Off

# Enable SQL safe mode
sql.safe_mode = On

# Avoid Opening remote files
allow_url_fopen = Off
```

See also:

- PHP Security: Limit Resources Used By Script
- PHP.INI settings: Disable exec, shell_exec, system, popen and Other Functions To Improve Security

#17: Run Nginx In A Chroot Jail (Containers) If Possible

Putting nginx in a chroot jail minimizes the damage done by a potential break-in by isolating the web server to a small section of the filesystem. You can use traditional chroot kind of setup with nginx. If possible use FreeBSD jails, XEN, Linux containers on a Debian/Ubuntu, LXD on a Fedora, or OpenVZ virtualization which uses the concept of containers.

#18: Limits Connections Per IP At The Firewall Level

A webserver must keep an eye on connections and limit connections per second. This is serving 101. Both pf and iptables can throttle end users before accessing your nginx server.

Linux Iptables: Throttle Nginx Connections Per Second

The following example will drop incoming connections if IP make more than 15 connection attempts to port 80 within 60 seconds:

```
/sbin/iptables -A INPUT -p tcp --dport 80 -i eth0 -m state --state NEW -m recent --set
/sbin/iptables -A INPUT -p tcp --dport 80 -i eth0 -m state --state NEW -m recent --updat
service iptables save
```

BSD PF: Throttle Nginx Connections Per Second

Edit your /etc/pf.conf and update it as follows. The following will limits the maximum number of connections per source to 100. 15/5 specifies the number of connections per second or span of seconds i.e. rate limit the number of connections to 15 in a 5 second span. If anyone breaks our rules add them to our abusive_ips table and block them for making any further connections. Finally, flush keyword kills all states created by the matching rule which originate from the host which exceeds these limits.

```
webserver_ip="202.54.1.1"
table <abusive_ips> persist
block in quick from <abusive_ips>
pass in on $ext_if proto tcp to $webserver_ip port www flags S/SA keep state (max-s
```

Please adjust all values as per your requirements and traffic (browsers may open multiple connections to your site). See also:

1. Sample PF firewall script.
2. Sample Iptables firewall script.

#19: Configure Operating System to Protect Web Server

Turn on SELinux as described above. Set correct permissions on /nginx document root. The nginx runs as a user named nginx. However, the files in the DocumentRoot (/nginx or /usr/local/nginx/html) should not be owned or writable by that user. To find files with wrong permissions, use:

```
# find /nginx -user nginx
# find /usr/local/nginx/html -user nginx
```

Make sure you change file ownership to root or other user. A typical set of permission /usr/local/nginx/html/

```
# ls -l /usr/local/nginx/html/
```

Sample outputs:

```
-rw-r--r-- 1 root root 925 Jan  3 00:50 error4xx.html
-rw-r--r-- 1 root root  52 Jan  3 10:00 error5xx.html
-rw-r--r-- 1 root root 134 Jan  3 00:52 index.html
```

You must delete unwated backup files created by vi or other text editor:

```
# find /nginx -name '.?*' -not -name .ht* -or -name '*~' -or -name '*.bak*' -or -name
'*.old*'
# find /usr/local/nginx/html/ -name '.?*' -not -name .ht* -or -name '*~' -or -name '*.bak*' -
or -name '*.old*'
```

Pass -delete option to find command and it will get rid of those files too.

#20: Restrict Outgoing Nginx Connections

The crackers will download file locally on your server using tools such as wget. Use iptables to block outgoing connections from nginx user. The ipt_owner module attempts to match various characteristics of the packet creator, for locally generated packets. It is only valid in the OUTPUT chain. In this example, allow vivek user to connect outside using port 80 (useful for RHN access or to grab CentOS updates via repos):

```
/sbin/iptables -A OUTPUT -o eth0 -m owner --uid-owner vivek -p tcp --dport 80 -m state -
```

Add above rule to your iptables based shell script. Do not allow nginx web server user to connect outside.

#21: Keep your software up to date

You must keep your software and kernel up to date all time. Apply patch as per your version or distro. If you are using a Debian/Ubuntu Linux use apt-get command/apt command to apply patches:

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

If you are using a RHEL/CentOS/Oracle/Scientific Linux, use yum command:

```
$ sudo yum update
```

If you are using an Alpine Linux use apk command:

```
# apk update
# apk upgrade
```

#22 : Avoid clickjacking

Add the following in your nginx.conf or virtual domain to avoid clickjacking:

```
add_header X-Frame-Options SAMEORIGIN;
```

#23 : Disable content-type sniffing on some browsers

Add the following in your nginx.conf or virtual domain:

```
add_header X-Content-Type-Options nosniff;
```

#23 : Enable the Cross-site scripting (XSS) filter

Add the following in your nginx.conf or virtual domain:

```
add_header X-XSS-Protection "1; mode=block";
```

#24: Force HTTPS

There is no reason to use HTTP. Force everyone to use HTTPS by deault:

1. How To: Nginx Redirect All HTTP Request To HTTPS Rewrite 301 Rules
2. How to redirect non-www to www HTTP / TLS /SSL traffic on Nginx

#25: Monitoring nginx

The Stub Status allows to see total number of client requests and other info. See how to enable it for more info:

1. nginx: See Active connections / Connections Per Seconds
2. HowTo: Enable Nginx Status Page

Bounce Tip: Watching Your Logs & Auditing

Check the Log files. They will give you some understanding of what attacks is thrown against the server and allow you to check if the necessary level of security is present or not.

```
# grep "/login.php??" /usr/local/nginx/logs/access_log
# grep "...etc/passwd" /usr/local/nginx/logs/access_log
# egrep -i "denied|error|warn" /usr/local/nginx/logs/error_log
```

The auditd service is provided for system auditing. Turn it on to audit service SELinux events, authetication events, file modifications, account modification and so on. As usual disable all services and follow our "Linux Server Hardening" security tips.

Conclusion

Your nginx server is now properly harden and ready to server webpages. However, you should be consulted further resources for your web applications security needs. For example, wordpress or any other third party apps has its own security requirements.

REFERENCES:

- 20 Linux Server Hardening Security Tips
- HowTo: Setup ~~nginx reverse proxy~~ and HA cluser with the help of keepalived.
- nginx wiki – The official nginx wiki.
- OpenBSD specific Nginx installation and security how to.

SHARE ON Facebook Twitter

Posted by: Vivek Gite

The author is the creator of nixCraft and a seasoned sysadmin, DevOps engineer, and a trainer for the Linux operating system/Unix shell scripting. Get the **latest tutorials on SysAdmin, Linux/Unix and open source topics via RSS/XML feed** or weekly email newsletter.

64 comment

Leo March 6, 2010 at 11:51 am

Very nice post...

tip top March 6, 2010 at 12:31 pm

Can you add Apache specific security tips?

nixCraft March 6, 2010 at 1:58 pm

Apache? Y'days technology? Just kidding .. I will add when I've some free time but no ETA.

MC.Spring March 6, 2010 at 2:25 pm

Very good job

Thanks for share!

Robin March 6, 2010 at 5:56 pm

I find it ironic that you refer to apache as yesterday's technology, indirectly implying that you need to be able to handle massive loads on your server, and then set a very restrictive source throttling for PF of a crazy 15 connections per 5 seconds – do you even know how many connections a browser, even with keepalive, will spit out when digesting f.e. an html file with 50 tags in it? Obviously not. Also, you should auto-flush that blocked table regularly, because you will have _a lot_ of false positives with a restrictive rule like that. Very bad advice there.

I also find it odd that you haven't gotten up to speed on the removal of safe_mode in PHP, and, that you in your SSL certificate generation make use of 3DES, which is anything but secure. Also pretty bad advice.

Kenny Hendrick November 3, 2015 at 5:53 am

Hello,

I read your response to something I guess needs to be learned (my site is somehow somebody else's site).

In any event, I was wondering if you ever were able to take the time and write good instructions on hardening the nginx webserver.

I did a httrack mirror of my site and received 3 of my domains on a server set for "default"! LOL

Anyway, if you've any good hints/pointers, I'd appreciate it.

Craig February 29, 2016 at 2:06 pm

Sorry i realise this is an old post but i think the guy is just trying to provide some helpful hints to optimizing his install. You would be daft to copy and paste anyones "quick start" configs to run your own production server and expect everything to run perfectly. His post does not say this is the definitive guide to setting up nginx security and if you read his comment about Apache being yesterdays technology he later posts he was only joking.

Ali October 13, 2016 at 12:28 am

I felt absolutely same way Robin. Thanks for expressing my thoughts ☺

People should not believe whatever they see on the internet. Second hand knowledge is very dangerous. Especially if you're running mission critical systems.

Juan Giordana March 6, 2010 at 11:56 pm

There's a nice nginx howto that may complement these tips at https://calomel.org/nginx.html

Ayman Fekri March 7, 2010 at 6:09 am

very Good post.
But :why u consider mail () as dangerous functions ?

> **Felipe** June 5, 2014 at 5:21 am
>
> I think it's because mail injection:
>
> http://securephpwiki.com/index.php/Email_Injection

Emin March 7, 2010 at 2:02 pm

Re: #10

I find it much more clean and convenient to simply create a default website with blank webpage (or return error if preferred) that will respond to all non-matched queries.

Amr El-Sharnoby March 7, 2010 at 6:27 pm

Hello, Thanks a lot
I've already implemented nginx on multiple servers to serve more than 200 TB of data monthly ..yes Terabyte not Gigabye, I know it.
Here is some comments I've;
Re:#17: Run Nginx In A Chroot Jail (Containers) If Possible;
You CAN, of course, use traditional chroot kind of setup with nginx. It's just a little bit tricky, I'm already setting it up with php fastcgi server chroot'ed too. you can contact me if you need the steps.

Re:#18: Limits Connections Per IP At The Firewall Level;
You can use something like the following in nginx; – this is already what I use on heavily loaded servers with many visitors behind proxies –

limit_req_zone $binary_remote_addr zone=ratezone:20m rate=16r/s;
limit_req zone=ratezone burst=160 nodelay;

I believe that nginx can do it better than iptables, specially under a DDoS attack, because the iptables recent module have a maximum memory limit of 8MB, as I can remeber it, and after that it's either completely fail or drop everything … nginx will do always behave better.

Re: #20: Restrict Outgoing Nginx Connections;
I think that It's better to do that using selinux policy … if you use seedit , you can add some line like this to the nginx_t.sp ..

allownet -protocol tcp -port 21,25,80,110,143,443 client;

Thanks a lot

js&c March 8, 2010 at 12:16 pm

@Amr,

Can you share your instructions on chrooting Nginx in a chroot jail?

nixCraft March 8, 2010 at 1:40 pm

@js&c,

You can chroot nginx using chroot command under CentOS / RHEL or any Linux disro as follows. You need to copy /usr/local/nginx to your $D. Next copy /etc/{passwd,group,hosts,resolv.conf,php.ini} to $D/etc, You need to copy required libs to $D. Once done copy /lib64/* to $D too. Copy php-cgi to $D/usr/bin. Finally, copy required php modules such gd, php-mysql to $D/usr/lib64/php/modules directory. Run php-cgi in $D using the following syntax

```
/usr/bin/spawn-fcgi -c $D -a 192.168.1.10 -p 9000 -P /var/run/php-cgi.fastcgi.pid -
```

Where,
D=/jail.dir

You need to place /dev/null and a few more entries in $D/dev. Do not add hard disk and/or any other block device entries in $D/dev. This is the main problem with chroot and it can be easily escaped if proper care is not taken, hence I recommend proper tools.

Update nginx.conf and point fastcgi to 192.168.1.10:9000. Once done start nginx as

```
chroot $D /usr/local/nginx/sbin/nginx
```

HTH

> **edogawaconan** August 9, 2010 at 9:08 am
>
> There's no need to chroot both php-cgi and nginx in one place. Additionally, php-fpm has chroot functionality built-in.

robert March 15, 2010 at 2:25 pm

Hi,

It's great to see the complete step by step on hardening nginx web server.

Would you consider in writing something like that for lighttpd web server? ☺

> **nixCraft** April 12, 2010 at 8:23 am
>
> Yes, both Apache and Lighttpd are on my TODO lists. So stay tuned.

Vamsi Krishna March 15, 2010 at 3:29 pm

Thank you very much sir ☺

Alok Kumar April 14, 2010 at 1:11 pm

nice article, quite an informative

vinod April 17, 2010 at 5:59 am

quite nice article.. but I have not understood yet why selinux is important ☺ … I have been a FreeBSD guy and now started working on CentOS too…

I've setup a video streaming server, using Nginx and php-fpm… (this server transmits @ 3-4 mbps at average) I am seeing a lot of erros like "connection to upstream timed out " etc , which throws a " Bad Gateway" at times. After a lot of googling I increased the time out of fcgi and that seem to alleviate the issue, but I am seeing such entries in the logs often. I assume the issue is with nginx getting failed to communicate with PHP engine…

I wonder if the error is common and do we have hotfix for the issue ? I doubt if that is an issue with any compiled module ?

Thanks!
Vinod !

ruo91 April 25, 2010 at 2:41 pm

Very good!!

edogawaconan August 9, 2010 at 9:15 am

#10 should be done using server {} block.

```
server {
listen 80 default;
return 403;
}
```

#11, I doubt nginx supports any other methods

And #12... I doubt there's any spambots left running using non-common user agent.

Additionally, running php-cgi and nginx daemons as different user is recommended. Setting owner of the files to root and making it non-group/world writable except for some directories used by php (in which should be set to php-owned and not group/world writable) is also recommended.

Ahmed October 24, 2010 at 2:52 pm

#12: How Do I Block Referral Spam?
Please remove that ! It can't make anything just getting CPU load average from 3 to 8 it makes nginx slower and it's not good for seo also.

v13 November 23, 2010 at 8:47 am

nice nginx security practices

Dan Connor March 24, 2011 at 7:20 am

I ran into some issues reducing keepalive_time that far. The Nginx default is 75 seconds, so reducing it to 5 is definitely extreme. I had to increase it when using a backend dynamic threaded application server such as PHP5-FPM or it would end up producing intermittent 503 errors when spinning up new threads under load. See http://danconnor.com/503-errors-with-nginx-and-php5-fpm.html

At least thus far it seems like the keepalive parameter was the culprit.

dl May 27, 2011 at 12:04 am

There is a small problem with .htaccess protection rules. If you use domain.com/adm/ it works however if you use domain.com/adm/index.php it is not working. I even tried defining php conf inside but still not working

here are the two rules i used

```
location ~ /(adm/.*|install/.*) {
    auth_basic            "Restricted";
    auth_basic_user_file  /path/to/htpasswd;
        location ~ .php$ {
                fastcgi_index                                   index.php;
                fastcgi_pass                                127.0.0.1:9000;
                include                                         fastcgi.cor
        }
}
location ~ /(adm/.*|install/.*) {
    auth_basic            "Restricted";
    auth_basic_user_file  /path/to/htpasswd;
}
```

How can i make sure domain.com/adm/index.php and rest of php files inside adm directory are also password protected

Bangon Kali June 26, 2011 at 5:28 pm

Thank you very much! These are very helpful!

gunt October 6, 2011 at 2:47 pm

Hi,
Thank you for this post. It really help me a lot.
I need your help with the hotlinking part, could you please tell me exactly which file do I need to edit to stop the bad guys using my images.
I'll appreciate your help 'cause I can't find anywhere this info!
thanks

jake October 18, 2011 at 12:26 pm

THANK YOU SOOOOOOOOOOO MUCH.

My server has suffered from socket port exhaustion for 2 years now.
Ive tried every sysctl variable and a hundred configurations from various linux administrators, and only YOUR sysctl.conf file did the trick.

Im not sure why, ive used all these parameters before, but it finally fixed the problem on centos and now I can run a load test for hours and never suffer from port exhaustion.
YOU ARE THE MAN!

SuilAmhain November 26, 2011 at 2:35 am

Hi,

Just a quick question on point #7 Restrictive Iptables Based Firewall

I'm trying to get an understanding of iptables and in doing so your output rules confuse me.
Plese see below example:

> $IPT -A INPUT -i ${PUB_IF} -s ${ip} -p tcp -d ${SERVER_IP} –destination-port 22 -j ACCEPT
> $IPT -A OUTPUT -o ${PUB_IF} -d ${ip} -p tcp -s ${SERVER_IP} –sport 22 -j ACCEPT

I interpret that as being
Accept a SSH in to server from IP as defined in $PUB_SSH_ONLY.
Allow a ssh connection out to an IP as defined in $PUB_SSH_ONLY.

Why do you need the output rule?
Is it simply to allow a SSH connection to an IP defined in $PUB_SSH_ONLY or
is the output required as part of a handshaking process?

I know I should experiment and see but I'm curious to the answer and that I may have a gap in my iptables understanding.

Thanks,
SuilAmhain

Arthur January 23, 2012 at 3:19 pm

TIP:
I use NGinx as reverse proxy to several Apache nodes.
And I had some "Bad Gateway" problems when using IPv6.

After try open a page using an IPv6 (AAAA) name,
nginx crashed with "Bad Gateway" message and Apache gives "Segmentation Fault".

After dig a little bit, I discover that the problem were in the (lack of) IPv6 Linux Module.
A “modprobe ipv6” solved my problem.
=D

Phoenix May 12, 2012 at 8:44 am

The “client_header_buffer_size” etc — are they just per-request, or are they the overall buffer size of whatever Nginx maintains? It’s unclear.

soton junior June 13, 2012 at 1:24 pm

Very2 Nice Post ...
like it ..
^_^

parsigate July 17, 2012 at 10:29 am

I was looking for it... will try to implement some of them

Nesousx August 29, 2012 at 12:10 pm

You could also use NAXSI in order to secure your Nginx config. I am currently testing it my website, looks really good so far.

Eugene August 31, 2012 at 8:41 am

Usefull tips , thank you!

Steph September 7, 2012 at 8:45 am

This is by far one of the best articles I found ...
Thank you for sharing this !

Frans October 19, 2012 at 5:21 pm

Impressive collection of tips and tricks! Very nicely done!

Ankit Srivastava February 26, 2013 at 9:36 pm

Very nice post.

Thank you for sharing this !

Phil April 2, 2013 at 8:44 pm

The typical way nginx and many other web servers detail how to be configured has the master process (what's bound to port 80) running as root. My preference is to have the web server bind to a high port like 8080 (and then be able to run all of the web servers processes as a restricted user). Then simply redirect connections to 80 to 8080 either on the local machine (using for example iptables) or in what ever gear you have between the world and the web server.

@edogawaconan touched on this with the very good advice of running the web server and server side scripts as different users.

And then you can get into matters of never installing apps like wordpress that include a public and private code base as a single installation (when I've done this you need to tweak some apps a little so some of the restrictions are possible, wordpress for example requires a little coding knowledge, but it's easy enough to do). Separate out the public component from the private and setup the public component with exceptionally limited access ie. read only access to most if not all of the database, read only access to most if not all of the file system, no permission to connect back out to the internet and limited permission to connect to local services... whereas private installation consisting of admin area ideally only available over VPN or SSH tunnel has fewer restrictions but still limited to just what it needs to get the job done.

Alejandro April 11, 2013 at 7:11 pm

Hi, great post, very useful, one question What are LB1_IP="204.54.1.1" and LB2_IP="204.54.1.2"? If I use the script I have to leave this ips or I have to change it?

> **Gerard** November 4, 2013 at 8:41 am
>
> @Alejandro: I was wondering the same thing and then it hit me. Looking at the logic in the script they are boad balancer (cluster node) IP's. This way you can use this script on the other node in the farm as well.

I'm assuming you know some clustering terms here ☺

Regards,

Gerard.

Marcus August 19, 2013 at 6:48 pm

Hello,

it is possible to integrate the blocking rules of sshguard in this great script? What needs to be changed in order to do this?

Or is this thought of mine completely wrong?

I thought the extra features would result in a very good addition to the script ...
Or is the script so good and comprehensive that sshguard is unnecessary?

Specifically it comes to supplements for sshguard that must be entered in iptables,
as described under sshguard.net/docs/setup/firewall/netfilter-iptables

best regards,
Marcus

Rajkumar kathane September 26, 2013 at 10:21 am

hello
thank for sharing post

Nicholas October 20, 2013 at 4:40 pm

Hi, About hot-linking. couple of questions.

```
"# Stop deep linking or hot linking
location /images/ {
  valid_referers none blocked www.example.com example.com;
   if ($invalid_referer) {
```

```
        return   403;
    }
}"
```

1. Do i renamed the /'images/' to say "imgs" if that is where I store my images?
2. The http://www.example.com example.com – Are these my passthru targets or are they the domains i intend to block?

nixCraft October 23, 2013 at 5:04 pm

> 1. Do i renamed the /â€™images/â€™ to say â€œimgsâ€ if that is where I store my images?

Yes.

> 2. The http://www.example.com example.com â€“ Are these my passthru targets or are they the domains i intend to block?

Replace example.com,www.example.com with your domain name i.e. your own domain names that are allowed to hotlink.

john December 2, 2013 at 10:02 pm

It sure would be nice if i could just do something like sudo .... and have everything you just put here.

Thanks for the share.

ura soul January 13, 2014 at 4:35 pm

the limit_zone directive has been deprecated – the following is the new syntax:

limit_conn_zone $binary_remote_addr zone=one:5m;
limit_conn one 5;

ura soul January 13, 2014 at 4:38 pm

where should #10 and #11 be inserted into the nginx config files?
so far i have found they are not accepted by nginx unless i put them inside a server element..
so i created a new server element within my site's config file (since these changes apply to the entire server and not just to a specific website config)..
and nginx accepts them.. but the site slowed down a lot.

e.g. i used:

```
server {
        listen 80;
        listen 443 ssl;
## Only requests to our Host are allowed
      if ($host !~ ^(mysite.com|www.mysite.com)$ ) {
         return 444;
      }
## Only allow these request methods ##
     if ($request_method !~ ^(GET|HEAD|POST)$ ) {
         return 444;
     }
## Do not accept DELETE, SEARCH and other methods ##
##
}
```

NIX Craft January 13, 2014 at 7:02 pm

Try in `server {}` section.

ken February 11, 2014 at 7:17 am

#11 should probably return a 405 instead of 444.

MVN March 14, 2014 at 5:53 pm

In #6: Install SELinux Policy To Harden The Nginx Webserver there is a problem.

semodule -i nginx.pp
libsepol.link_modules: Tried to link in a non-MLS module with an MLS base. (No such file or directory). libsemanage.semanage_link_sandbox: Link packages failed (No such file or directory). semodule: Failed!

One needs to make a change to the Makefile

Change

```
TYPE ?= $(NAME)-$(NTYPE)
```

to

```
TYPE ?= $(NTYPE)
```

And everything works. No idea what this is about – found it on a Russian web page.

dave April 13, 2014 at 1:48 am

hello,
I'm install nginx from dotdeb.
If I use chroot jail for nginx, when I upgrade to new nginx version, should I need to repeat the step on "traditional chroot kind"?

Thank you

> **NIX Craft** April 13, 2014 at 1:40 pm
>
> Yes.

Ariden May 11, 2014 at 2:39 pm

Hello,
In “#7: Restrictive Iptables Based Firewall” and “#18: Limits Connections Per IP At The Firewall Level” you did rules for iptables port 80.

Why don’t you put point 18 directly in the poit 7 ?
Is it ok if we add these lines in iptables rules of point 7?
$IPT -A INPUT -p tcp –dport 80 -i ${PUB_IF} -m state –state NEW -m recent –set
$IPT -A INPUT -p tcp –dport 80 -i ${PUB_IF} -m state –state NEW -m recent –update –seconds 60 –hitcount 15 -j DROP

thx

Ariden May 11, 2014 at 2:48 pm

In point 20: “Restrict Outgoing Nginx Connections”

I have got this message :

> Bad value for “–uid-owner” option: “vivek”
> Try `iptables -h’ or ‘iptables –help’ for more information.

when i try it.

How can i resolve it ?

thx

> **NIX Craft** May 12, 2014 at 4:10 am
>
> Replace UID vivek with your own UID.

morphix September 26, 2014 at 12:01 am

I use different server blocks for my various domains/sites and then in the default nginx.conf config have a ‘default’ site to capture stuff not directed at my domain(s):

I have it giving a 403 error:

```
server {
return 403;
}
```

This covers any listening ports i have (eg. HTTP (80) and HTTPS (SSL) to just give a 403.

TrÃ¹m WP February 9, 2015 at 11:53 pm

I like #4, #8, #10 and #16 tips in this article. I have used Nginx for a long times but I haven't known using these tips for it. Thank again for good tutorial. ☺

Jonathan March 10, 2016 at 5:06 pm

Let me get this straight, you're recommending people /increase/ their server security by compiling and running unsigned binaries from sourceforge on their servers?

wget 'http://downloads.sourceforge.net/project/selinuxnginx/se-ngix_1_0_10.tar.gz?use_mirror=nchc'

W.M. March 26, 2016 at 9:44 am

What do you suggest regarding the user setting inside nginx.conf? I serve websites at my system under the folder /srv/html

The /srv/html folder is owned by a normal Linux user (not root) e.g. jack. I need to be able to read / write to that folder by jack and at the same time make it possible for nginx/php-fpm to write (download) files to that folder. What I have done is to set nginx user to jack (inside nginx.conf). My question is this safe? or are there any better alternatives? Thanks.

Eugen June 22, 2016 at 10:41 am

How much RAM must have server for these parameters of kernel tuning ?

```
# Increase TCP max buffer size setable using setsockopt()
net.ipv4.tcp_rmem = 4096 87380 8388608
net.ipv4.tcp_wmem = 4096 87380 8388608
```

```
# Increase Linux auto tuning TCP buffer limits
# min, default, and max number of bytes to use
# set max to at least 4MB, or higher if you use very high BDP paths
# Tcp Windows etc
net.core.rmem_max = 8388608
net.core.wmem_max = 8388608
net.core.netdev_max_backlog = 5000
net.ipv4.tcp_window_scaling = 1
```

maclighiche August 4, 2017 at 8:31 am

very nice

Have a question? Post it on our forum!

Tagged as: mac os x, nginx, nginx security, reverse proxy, reverse proxy security, Updated tips 0