Tips and Tricks to Secure Your Nginx Web Server

Nginx is an open source, lightweight, high-performance the fastest growing web server around the world. Nginx runs on Linux, Windows, Mac OS, and Solaris operating system. NGINX continues to rise in popularity, so means more and more NGINX deployments need to be secured.

In this tutorial, we will explain some popular Nginx server security tips and tricks.

On this page

Requirements

- A server running Ubuntu 18.04 or Debian 9.
- A root password is set up on your server.

Install Nginx

First, you will need to install Nginx to your system. You can install it by running the following command:

```
apt-get install nginx -y
```

Once the Nginx has been installed, you can check the status of Nginx with the following command:

```
systemctl status nginx
```

You should see the following output:

```
? nginx.service - A high performance web server and a reverse proxy se
rver
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor
preset: enabled)
   Active: active (running) since Sun 2019-03-10 02:43:14 UTC; 4min 40
s ago
     Docs: man:nginx(8)
  Process: 2271 ExecStop=/sbin/start-stop-daemon --quiet --stop --retr
y QUIT/5 --pidfile /run/nginx.pid (code=exited, status=0/SUCCESS)
  Process: 2281 ExecStart=/usr/sbin/nginx -g daemon on; master_process
on; (code=exited, status=0/SUCCESS)
  Process: 2274 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; maste
r_process on; (code=exited, status=0/SUCCESS)
 Main PID: 2285 (nginx)
    Tasks: 2 (limit: 1111)
   CGroup: /system.slice/nginx.service
           ??2285 nginx: master process /usr/sbin/nginx -g daemon on;
master_process on;
           ??2290 nginx: worker process

Mar 10 02:43:14 ubuntu1804 systemd[1]: Starting A high performance web
server and a reverse proxy server...
Mar 10 02:43:14 ubuntu1804 systemd[1]: nginx.service: Failed to parse
PID from file /run/nginx.pid: Invalid argument
Mar 10 02:43:14 ubuntu1804 systemd[1]: Started A high performance web
server and a reverse proxy server.
```

Update Nginx

You will need to update your Nginx web server as there are many performance enhancement, new features and security fixes are being added. Most modern Linux distributions will not come with the latest version of nginx into their default package lists. So you will need to upgrade the latest version of nginx via a package manager. You can update your Nginx web server with the following command:

```
apt-get update -y
apt-get install nginx --reinstall -y
```

Prevent Information Disclosure

First, you will need to prevent the Nginx to disclose their version information.

By default, Nginx shows its name and version in the HTTP headers.

You can check it with the following command:

```
curl -I http://localhost
```

You should see the following output:

```
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Sat, 09 Mar 2019 15:28:01 GMT
Content-Type: text/html
Content-Length: 10918
Last-Modified: Fri, 01 Feb 2019 16:05:17 GMT
Connection: keep-alive
ETag: "5c546e3d-2aa6"
Accept-Ranges: bytes
```

In the above output, you should see the Nginx and operating system version.

You can hide this information by editing /etc/nginx/nginx.conf file:

```
nano /etc/nginx/nginx.conf
```

Add the server_tokens off line inside http configuration part:

```
http {

        ##
        # Basic Settings
        ##
        server_tokens off;
```

Save and close the file, when you are finished. Then, restart Nginx web server to apply the changes:

```
systemctl restart nginx
```

Now, run the curl command again:

```
curl -I http://localhost
```

You should see the following output:

```
HTTP/1.1 200 OK
Server: nginx
Date: Sat, 09 Mar 2019 15:33:31 GMT
Content-Type: text/html
Content-Length: 10918
Last-Modified: Fri, 01 Feb 2019 16:05:17 GMT
Connection: keep-alive
ETag: "5c546e3d-2aa6"
Accept-Ranges: bytes
```

Restrict the IPs from the Access

Nginx comes with a simple module called ngx_http_access_module to allow or deny a specific IP address.

If you want to allow Nginx form 172.16.0.0/16 and deny from other subnets. Then, open /etc/nginx /sites-enabled/default file:

```
nano /etc/nginx/sites-enabled/default
```

Make the following changes inside server block:

```
server {
        listen 80 default_server;
        listen [::]:80 default_server;

        allow 172.16.0.0/16;
        deny  all;
```

Save and close the file, when you are finished. Then, restart Nginx to apply these changes:

```
systemctl restart nginx
```

Now, try to access your Nginx server from other IP address range like 192.168.0.102.

Next, check the Nginx log with the following command:

```
tail -f /var/log/nginx/error.log
```

You should get access forbidden in the following output:

```
2019/03/09 16:13:01 [error] 11589#11589: *1 access forbidden by rule,
client: 192.168.0.102, server: _, request: "GET /test/ HTTP/1.1", hos
t: "172.16.0.122"
```

Secure Nginx with TLS

TLS (Transport Layer Security) is the successor to SSL (Secure Socket Layer). It provides stronger and more efficient HTTPS and contains more enhancements such as Forward Secrecy, compatibility with modern OpenSSL cipher suites, and HSTS. This tutorials shows how to enable a self-signed SSL Certificate in Nginx. If you want to use a let's Encrypt certificate instead, take a look here: https://www.howtoforge.com/tutorial/nginx-with-letsencrypt-ciphersuite/

First, create a directory for SSL with the following command:

```
mkdir /etc/nginx/ssl/
```

Next, generate a key and a certificate with the following command:

```
cd /etc/nginx/ssl/
```

First, generate key with the following command:

```
openssl genrsa -aes256 -out nginx.key 1024
```

You should see the following output:

```
Generating RSA private key, 1024 bit long modulus
...++++++
.............................++++++
e is 65537 (0x010001)
Enter pass phrase for nginx.key:
Verifying - Enter pass phrase for nginx.key:
```

Next, generate csr with the following command:

```
openssl req -new -key nginx.key -out nginx.csr
```

Provide all the information as shown below:

```
Generating RSA private key, 1024 bit long modulus
...++++++
.............................++++++
e is 65537 (0x010001)
Enter pass phrase for nginx.key:
Verifying - Enter pass phrase for nginx.key:
root@ubuntu1804:~# openssl req -new -key nginx.key -out nginx.csr
Enter pass phrase for nginx.key:
You are about to be asked to enter information that will be incorporat
ed
into your certificate request.
What you are about to enter is what is called a Distinguished Name or
a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:IN
State or Province Name (full name) [Some-State]:Gujarat
Locality Name (eg, city) []:Junagadh
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IT
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:HITESH
Email Address []:admin@example.com
```

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:admin
An optional company name []:IT
```

Next, sign the certificate with the following command:

```
openssl x509 -req -days 365 -in nginx.csr -signkey nginx.key -out ng
inx.crt
```

You should see the following output:

```
Signature ok
subject=C = IN, ST = Gujarat, L = Junagadh, O = IT, OU = IT, CN = HITE
SH, emailAddress = admin@example.com
Getting Private key
Enter pass phrase for nginx.key:
```

Next, open Nginx default virtual host file and define the certificate:

```
nano /etc/nginx/sites-enabled/default
```

Make the following changes:

```
server {
        listen 192.168.0.100:443 ssl;
        root /var/www/html;
        index index.html index.htm index.nginx-debian.html;
        server_name _;
        ssl_certificate /etc/nginx/ssl/nginx.crt;
        ssl_certificate_key /etc/nginx/ssl/nginx.key;
        ssl_protocols       TLSv1 TLSv1.1 TLSv1.2;
```

Save and close the file, when you are finished. Then, restart Nginx server to apply these changes:

```
systemctl restart nginx
```

Password Protect The Directory

When setting up an Nginx web server, you can also protect a specific directory with a password. You can do this using .htpasswd file.

To do so, create the passwd file and add the user to it with the following command:

```
mkdir /etc/nginx/.htpasswd
htpasswd -c /etc/nginx/.htpasswd/passwd admin
```

You should see the following output:

```
New password:
Re-type new password:
Adding password for user admin
```

Next, create a test directory inside Nginx web root with the following command:

```
mkdir /var/www/html/test
```

Next, give ownership to www-data user with the following command:

```
chown -R www-data:www-data /var/www/html/test
```

Next, open Nginx default virtual host file with the following command:

```
nano /etc/nginx/sites-enabled/default
```

Next, protect test directory as shown below:

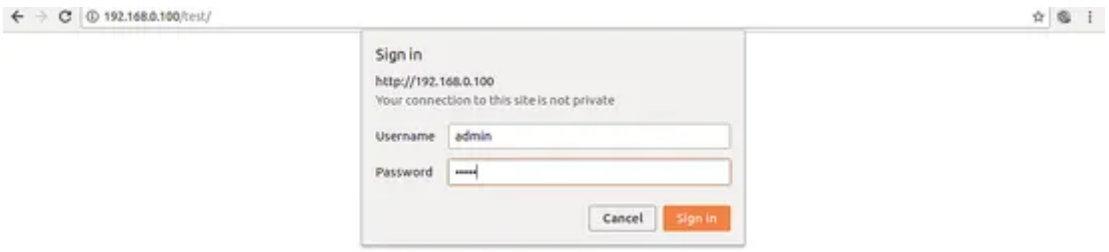
```
        location /test {

        auth_basic  "Restricted";
        auth_basic_user_file   /etc/nginx/.htpasswd/passwd;
```

Save and close the file, when you are finished. Then, restart Nginx service to apply these changes:

```
systemctl restart nginx
```

Next, open your web browser and type the URL http://your-server-ip/test. You will be prompt to enter username and password to access the test directory as shown in the following page:



Congratulations! you have successfully secured your Nginx server on Ubuntu 18.04 server. I hope this will help you to protect your application hosted on the Nginx web server. Feel free to ask me if you

have any questions. For more information, you can refer to the Nginx security doc.

About Hitesh Jethva

Over 8 years of experience as a Linux system administrator. My skills include a depth knowledge of Redhat/Centos, Ubuntu Nginx and Apache, Mysql, Subversion, Linux, Ubuntu, web hosting, web server, Squid proxy, NFS, FTP, DNS, Samba, LDAP, OpenVPN, Haproxy, Amazon web services, WHMCS, OpenStack Cloud, Postfix Mail Server, Security etc.

view as pdf | print

Share this page:

Follow

Suggested articles

7 Comment(s)

Add comment

Name *

Email *

B I

p

I'm not a robot

reCAPTCHA
Privacy - Terms

Submit comment

Comments

By: Petter Neumann Reply

These days, wouden't the more practical way be to use Let's encrypt / Certboot for HTTPS ?

By: till Reply

LE is nice and there are tutorials for it https://www.howtoforge.com/tutorial/nginx-with-letsencrypt-ciphersuite/ and https://www.howtoforge.com/tutorial/install-letsencrypt-and-secure-nginx-in-debian-9/ But LE does not work in all environments, e.g. you can't get a LE cert in your local network.

By: Not me Reply

This is NOT securing an nginx server. SSL is not security, it is for privacy only.
Using Let's Encrypt would be much better. To accomplish that, acme.sh is much easier to use for requesting and deploying the certs. There are many how-tos for that.
I hoped to learn about deploying u2f for 2FA with this article and for methods to dynamically block myphpadmin requests. BTW, if you use myphpadmin, please only allow localhost connections - for the users to ssh into the machine with a tunnel first. Actually, it would be good to block all .php requests from any outside country, IMHO.
Also blocking WP admin requests and putting those source IPs into a gulag would be helpful.
Or perhaps how to prevent brute force attacks against the normally terrible end-user passwords with fail2ban? A nice regex for this would be good.
Security is more than just blocking internet access to an internet service.

By: till Reply

The tutorial shows how to restrict access by IP, how to prevent that Nginx version information is shown, how to protect the website with a password and how to secure the server with a self-signed SSL cert. All these things are important to secure an Nginx server. We covered Let's encrypt certificates for Nginx servers already here https://www.howtoforge.com/tutorial/nginx-with-letsencrypt-ciphersuite/ and https://www.howtoforge.com/tutorial/install-letsencrypt-and-secure-nginx-in-debian-9/

By: Warren Reply

I recommend looking into ngx_http_geoip_module, to limit which country your reverse proxy reponds to.It should go without saying that this could likely be circumvented by using a vpn.

By: CTan Reply

I have a question here. Anyway, for the nginx.key (which is a private key) to be protected? Because everyone can see when you open the conf file.

By: shailendra S Reply

Hi ,
my requirement is i need to secure my application throug one off the custom API which accept user id and password.
So first my custom API will call once it is success then only the main application should be call.

Please suggets how can we achive through Nginx configraution file.
i tried with ngx_http_auth_request_module with subrequest ,but i dont know how to call my api which is having post method.how i will get user id and password from client and pass to my sub API.

Please suggest.