DigitalOcean

Subscribe Share Contents

How To Secure Nginx with Let's Encrypt on Ubuntu 16.04

443

Updated October 27, 2017 1m NGINX LET'S ENCRYPT SECURITY UBUNTU 16.04

By: Mitchell Anicas

Not using **Ubuntu 16.04**? Choose a different version:

Introduction

ROLL TO TOP

providing a software client, Certbot, that attempts to automate most (if not all) of the required steps. Currently, the entire process of obtaining and installing a certificate is fully automated on both Apache and Nginx.

In this tutorial, you will use Certbot to obtain a free SSL certificate for Nginx on Ubuntu 16.04 and set up your certificate to renew automatically.

This tutorial uses the default Nginx configuration file instead of a separate server block file. We recommend creating new Nginx server block files for each domain because it helps to avoid some common mistakes and maintains the default files as a fallback configuration as intended. If you want to set up SSL using server blocks instead, you can follow this Nginx server blocks with Let's Encrypt tutorial.

Prerequisites

To follow this tutorial, you will need:

- One Ubuntu 16.04 server set up by following this initial server setup for Ubuntu 16.04 tutorial, including a sudo non-root user and a firewall.
- A fully registered domain name. This tutorial will use `example.com` throughout. You can purchase a domain name on Namecheap, get one for free on Freenom, or use the domain registrar of your choice.
- Both of the following DNS records set up for your server. You can follow this hostname tutorial for details on how to add them.
    - An A record with `example.com` pointing to your server's public IP address.
    - An A record with `www.example.com` pointing to your server's public IP address.
- Nginx installed by following How To Install Nginx on Ubuntu 16.04.

Step 1 — Installing Certbot

The first step to using Let's Encrypt to obtain an SSL certificate is to install the Certbot software on your server.

Certbot is in very active development, so the Certbot packages provided by Ubuntu tend to be outdated. However, the Certbot developers maintain a Ubuntu software repository with up-to-date versions, so we'll use that repository instead.

First, add the repository.

```
$ sudo add-apt-repository ppa:certbot/certbot
```

You'll need to press `ENTER` to accept. Then, update the package list to pick up the new repository's package information.

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

And finally, install Certbot's Nginx package with `apt-get`.

```
$ sudo apt-get install python-certbot-nginx
```

Certbot is now ready to use, but in order for it to configure SSL for Nginx, we need to verify some of Nginx's configuration.

Step 2 — Setting up Nginx

Certbot can automatically configure SSL for Nginx, but it needs to be able to find the correct `server` block in your config. It does this by looking for a `server_name` directive that matches the domain you're requesting a certificate for.

If you're starting out with a fresh Nginx install, you can update the default config file. Open it with `nano` or your favorite text editor.

```
$ sudo nano /etc/nginx/sites-available/default
```

Find the existing `server_name` line and replace the underscore, `_`, with your domain name:

/etc/nginx/sites-available/default

```
. . .
server_name example.com www.example.com;
. . .
```

Save the file and quit your editor.

Then, verify the syntax of your configuration edits.

```
$ sudo nginx -t
```

If you get any errors, reopen the file and check for typos, then test it again.

Once your configuration's syntax is correct, reload Nginx to load the new configuration.

```
$ sudo systemctl reload nginx
```

Certbot will now be able to find the correct `server` block and update it. Next, we'll update our firewall to allow HTTPS traffic.

Step 3 — Allowing HTTPS Through the Firewall

If you have the `ufw` firewall enabled, as recommended by the prerequisite guides, you'll need to adjust the settings to allow for HTTPS traffic. Luckily, Nginx registers a few profiles with `ufw` upon installation.

You can see the current setting by typing:

```
$ sudo ufw status
```

It will probably look like this, meaning that only HTTP traffic is allowed to the web server:

```
Output
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
Nginx HTTP                 ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
Nginx HTTP (v6)            ALLOW       Anywhere (v6)
```

To additionally let in HTTPS traffic, we can allow the Nginx Full profile and then delete the redundant Nginx HTTP profile allowance:

```
$ sudo ufw allow 'Nginx Full'
$ sudo ufw delete allow 'Nginx HTTP'
```

Your status should look like this now:

```
$ sudo ufw status
```

```
Output
Status: active

To                         Action      From
--                         ------      ----
OpenSSH                    ALLOW       Anywhere
Nginx Full                 ALLOW       Anywhere
OpenSSH (v6)               ALLOW       Anywhere (v6)
Nginx Full (v6)            ALLOW       Anywhere (v6)
```

We're now ready to run Certbot and fetch our certificates.

Step 4 — Obtaining an SSL Certificate

Certbot provides a variety of ways to obtain SSL certificates, through various plugins. The Nginx plugin will take care of reconfiguring Nginx and reloading the config whenever necessary:

```
$ sudo certbot --nginx -d example.com -d www.example.com
```

This runs `certbot` with the `--nginx` plugin, using `-d` to specify the names we'd like the certificate to be valid for.

If this is your first time running `certbot`, you will be prompted to enter an email address and agree to the terms of service. After doing so, `certbot` will communicate with the Let's Encrypt server, then run a challenge to verify that you control the domain you're requesting a certificate for.

If that's successful, `certbot` will ask how you'd like to configure your HTTPS settings.

```
Output
Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access.
-------------------------------------------------------------------------------
1: No redirect - Make no further changes to the webserver configuration.
2: Redirect - Make all requests redirect to secure HTTPS access. Choose this for
new sites, or if you're confident your site works on HTTPS. You can undo this
```

```
change by editing your web server's configuration.
-------------------------------------------------------------------------------
Select the appropriate number [1-2] then [enter] (press 'c' to cancel):
```

Select your choice then hit `ENTER`. The configuration will be updated, and Nginx will reload to pick up the new settings. `certbot` will wrap up with a message telling you the process was successful and where your certificates are stored:

Output

```
IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at
   /etc/letsencrypt/live/example.com/fullchain.pem. Your cert will
   expire on 2017-10-23. To obtain a new or tweaked version of this
   certificate in the future, simply run certbot again with the
   "certonly" option. To non-interactively renew *all* of your
   certificates, run "certbot renew"
 - Your account credentials have been saved in your Certbot
   configuration directory at /etc/letsencrypt. You should make a
   secure backup of this folder now. This configuration directory will
   also contain certificates and private keys obtained by Certbot so
   making regular backups of this folder is ideal.
 - If you like Certbot, please consider supporting our work by:

   Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
   Donating to EFF:                    https://eff.org/donate-le
```

Your certificates are downloaded, installed, and loaded. Try reloading your website using `https://` and notice your browser's security indicator. It should indicate that the site is properly secured, usually with a green lock icon. If you test your server using the SSL Labs Server Test, it will get an **A** grade.

Let's finish by testing the renewal process.

Step 5 — Verifying Certbot Auto-Renewal

Let's Encrypt's certificates are only valid for ninety days. This is to encourage users to automate their certificate renewal process. The `certbot` package we installed takes care of this for us by running ‘certbot renew’ twice a day via a systemd timer. On non-systemd distributions this functionality is provided by a script placed in `/etc/cron.d`. This task runs twice a day and will renew any certificate that's within thirty days of expiration.

To test the renewal process, you can do a dry run with `certbot`:

```
$ sudo certbot renew --dry-run
```

If you see no errors, you're all set. When necessary, Certbot will renew your certificates and reload Nginx to pick up the changes. If the automated renewal process ever fails, Let's Encrypt will send a message to the email you specified, warning you when your certificate is about to expire.

Conclusion

In this tutorial, you installed the Let's Encrypt client `certbot`, downloaded SSL certificates for your domain, configured Nginx to use these certificates, and set up automatic certificate renewal. If you have further questions about using Certbot, their documentation is a good place to start.

By: Mitchell Anicas

Upvote (443) Subscribe Share

Editor:
Hazel Virdó

Related Tutorials

How to Retrieve Let's Encrypt SSL Wildcard Certificates using CloudFlare Validation on CentOS 7

How To Create a Kubernetes 1.11 Cluster Using Kubeadm on Ubuntu 18.04

How To Develop a Node.js TCP Server Application using PM2 and Nginx on Ubuntu 16.04

How To Install Nginx on Ubuntu 18.04 [Quickstart]

How To Serve Flask Applications with Gunicorn and Nginx on Ubuntu 18.04

201 Comments

Leave a comment...

Log In to Comment

frankis *March 31, 2016*

2 Thanks for sharing, can you please comment on the two following topics?

1. You are referring to initial server setup for Ubuntu 16.04 tutorial - which ends in 404.
2. Following the (your) 14.04 security guides the root account has no remote ssh access. I am therefore logging in with "remoteuser" (just an example). onced logged-in I do require to enter a password for being able to execute commands with sudo permission.

Question 1: Have there been changes to 14.04 and if so: when will the referenced article be available?

Question 2: Shouldn´t nginx also run with its own permissions? If so, does that require any tweaks to the above guide?

Question 3: Under which permission are the cron jobs executed following the above guide? If root, would it make sense to have cron jobs run under a different account? And if that is the case and assuming nginx also have its own permissions under which it is running, does that somehow require additional tweaks to you guide?

iamkingsleyf *April 27, 2016*

0 Hello;

Can i use this on a subdomain? ads.mysite.com?

hackzilla *May 3, 2016*

1 yeah,

./letsencrypt-auto certonly -a webroot --webroot-path=/var/www/html -d example.com -d www.example.com -d ads.example.com

or treat it like a separate domain.

iamkingsleyf *May 3, 2016*

0 i want it as a separate domain

stphnlwlsh *June 29, 2016*

1 webroot-path should be the same for the sub-domain as the TLD?

iamkingsleyf *April 27, 2016*

0 Few questions please.

Can i do this for each site on the server?

hackzilla *May 3, 2016*

0 yes, just repeat the bits with example.com for each domain

chgraham *May 8, 2016*

This doesn't seem to work with a 512 MB droplet (i.e., I noticed a "cannot allocate virtual memory"

0 This doesn't seem to work with a 512 MB droplet (i.e., I noticed a "cannot allocate virtual memory error when running the './letsencrypt-auto ...' command).

This error was followed shortly thereafter by an InsecurePlatformWarning https://urllib3.readthedocs.io/en/latest/security.html#insecureplatformwarning

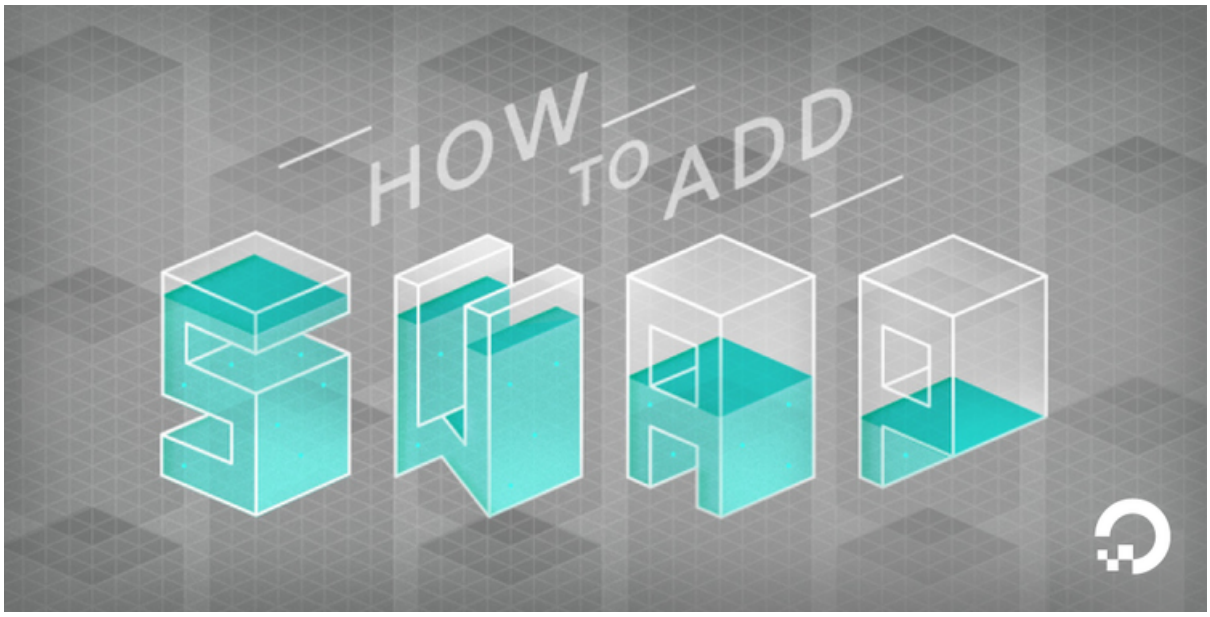
...and then the command terminated in failure.

andynazay153 *June 11, 2016*

0 It does if it does not have anything else cluttering it...

Danje *August 6, 2016*

2 Try adding a system Swap File.
https://www.digitalocean.com/community/tutorials/how-to-add-swap-space-on-ubuntu-16-04



How To Add Swap Space on Ubuntu 16.04

by Justin Ellingwood

One of the easiest way of increasing the responsiveness of your server and guarding against out of memory errors in your applications is to add some swap space. In this guide, we will cover how to add a swap file to an Ubuntu 16.04 server. <$>[warning] [label...

luis02lopez *May 24, 2017*

1 Swap in not recommendable in DigitalOcean due to the SSD disks...

stooj *May 20, 2016*

3 Thanks for the great tutorial.

I was able to install the certificate using the letsencrypt version in the 16.04 repos.
So, instead of cloning the git repo, I was able to just `apt-get install letsencrypt`

Consequently, all the `./letsencrypt` commands become `sudo letsencrypt`

du5rte *July 20, 2016*

1 how would you use `./letsencrypt/letsencrypt-auto` ? or is it all the same?

knnleow *May 23, 2016*

0 thank for sharing this. just implemented on my web server.
one thing will like to suggest is to include the ".ini" file.
takes me some googling to get this syntax out.

most people do not have GUI especially running inside a docker container and this will be helpful

```
 # vi /opt/letsencrypt/LE_server01.example.com.ini
      rsa-key-size = 2048

     server = https://acme-v01.api.letsencrypt.org/directory
     text = True
     agree-tos = True
     verbose = True

     authenticator = webroot
    email = username01@example.com
    domains = server01.example.com
:wq!

# cd /opt/leteencrypt
# ./letsencrypt-auto certonly -c LE_server01.example.com.ini
```

Fightface *June 1, 2016*

After doing this I got this error when checking if there were any syntax errors.

0

```
nginx: [emerg] BIO_new_file("/etc/ssl/certs/dhparam.pem") failed (SSL: error:02001002
```

Then after that happened I closed Putty and reopened it, and suddenly I couldn't connect anymore, making it so I couldn't revert my changes. Help would be appreciated.

PS: Getting timed out gave me this error: Network error: Connection timed out
This happened right after I had done this.

---

philippe92df865 *September 7, 2016*

1 if you issue the command sudo openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048, it should go out.

---

brine *February 16, 2017*

1 This little obscure line is VERY important if you're on a LEMP/16.04 install. The dhparam.pem does *not* exist, so you need to generate it. As the above states, run:

```
sudo openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048
```

This will generate the file and fix the error above.

---

gfelot33 *June 2, 2016*

0 Thanks for this tutorial.
Just a quick question. I generate my certif for my website and two subdomain.
But I wasn't able to setup my nginx blocks with ssl.

i tried to add a 301 to an other file like we did with this tutorial but I cannot add a other name than "default_server" to redirect the block to an other.

Here a snippet :
mainWebServer

```
server {
        listen       80;
        listen       [::]:80;
        server_name      gfelot.xyz www.gfelot.xyz;
        return           301 https://$host$request_uri;
```

```
}

server {
       listen 443 ssl http2 default_server;
       listen [::]:443 ssl http2 default_server;
       include snippets/ssl-gfelot.xyz.conf;
       include snippets/ssl-params.conf;

       server_name gfelot.xyz;
[...]
```

transmissionWebServer

```
server {
       listen  80;
       listen [::]:80;
       server_name dl.gfelot.xyz

       access_log off;
       error_log /var/log/nginx/dl.gfelot.xyz.log;

       location / {
                proxy_pass http://127.0.0.1:9091/web/;
                proxy_set_header Connection "";
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_pass_header X-Transmission-Session-Id;
                }

        location /rpc {
                proxy_pass http://127.0.0.1:9091/rpc;
                proxy_set_header Connection "";
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_pass_header X-Transmission-Session-Id;
                }
}
```

plexWebServer

```
upstream plex-upstream {
         server localhost:32400;
}

server {
       listen 80;
       listen [::]:80;
       server_name plex.gfelot.xyz;

       location / {
                if ($http_x_plex_device_name = '') {
                   rewrite ^/$ http://$http_host/web/index.html;
        }
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_redirect off;
        proxy_set_header Host $http_host;
        proxy_pass http://plex-upstream;
       }
}
```

jonathan *June 3, 2016*

4 Hi @manicas - great tutorial, but they've changed all the names and now it's called Certbot; there's also an nginx plugin now, too!

See https://certbot.eff.org/docs/using.html#nginx :)

ppkrauss *March 13, 2017*

0 I preffer to use "standard Certbot"... How to use (please an example!) the `renew` command?

gfelot33 *June 8, 2016*

0 Hey guys ! If I want to add an other certif after setting everything up once for an other subdomain, do I have to

```
./letsencrypt-auto certonly -a webroot --webroot-path=/var/www/html -d example.com -d
```

or just

```
./letsencrypt-auto certonly -a webroot --webroot-path=/var/www/html -d sub2.example.c
```

Thanks !

gfelot33 *June 8, 2016*

0 It seems to be the first one.
But I got an error when I try to expend my certificat.

```
[...]
Domain: gfelot.xyz
   Type:   connection
   Detail: Could not connect to
   http://gfelot.xyz/.well-known/acme-challenge/QuCDclLfdtw3IgHEz9f-tvTy7a_7NQZJuf

   To fix these errors, please make sure that your domain name was
   entered correctly and the DNS A record(s) for that domain
   contain(s) the right IP address. Additionally, please check that
   your computer has a publicly routable IP address and that no
   firewalls are preventing the server from communicating with the
   client. If you're using the webroot plugin, you should also verify
   that you are serving files from the webroot path you provided.
```

gosha *June 10, 2017*

0 Had the same problem -- just change nginx location to /.well-know/acme-challenge
I describe my approach in this article https://900913.ru/2017/06/09/kak-dobavit-ssl-na-ubuntu-server-16-04-i-vyshe/ -- you can read my configs.

andynazay153 *June 11, 2016*

1 This does not seem to work if trying to use http2, can you create a tutorial showing how?

keytouch *June 15, 2016*

2 You can add lets encrypt directly in 16.04 using `apt-get install letsencrypt`

---

squarology *June 16, 2016*

0 super happy that I got this to work! perfectly explained.

DaveTeu *June 16, 2016*

2 I encountered the follwing problem

[warn] "ssl_stapling" ignored, issuer certificate not found

SSL is working fine. I'm trying to get HTTP2 working on firefox.

---

tridnguyen *May 11, 2017*

0 Is there an answer for why this is the case?

---

geeksmanga *June 15, 2017*

0 do you have :
listen 443 ssl http2 default*server;*
*listen [::]:443 ssl http2 default*server;

---

fdaciuk *June 17, 2016*

0 Works fine for me! Thanks for sharing <3

NitinKarwasra *June 28, 2016*

0 Can i install this on 512 mb ram server to use with wordpress

---

stphnlwlsh *June 29, 2016*

0 Yes.

---

stphnlwlsh *June 29, 2016*

3 For those wondering about doing this for a subdomain, I just got this working properly.

`toplevel.com` == your top level domain
`sub.toplevel.com` == your sub domain

1. When running the `certonly` step implement your top level and subdomains like this if they are separate directories like mine. Obviously, you'll want this all on one line. I broke it out a little bit so it would be easier to see what's happening.

```
letsencrypt certonly --webroot
-w /var/www/mytoplevel.com/html/
    -d www.mytoplevel.com
    -d mytoplevel.com
-w /var/www/sub.mytoplevel.com/html
    -d sub.mytoplevel.com
```

1. Setup server blocks appropriately. This is just how I have mine setup, and it's working correctly. I followed this guide before running `letsencrypt`

Default Site

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

        root /var/www/html;

        # Add index.php to the list if you are using PHP
        index index.php index.html index.htm index.nginx-debian.html;

        server_name _;

        # other configuration below
}
```

Top level domain in /etc/nginx/sites-available/toplevel.com

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

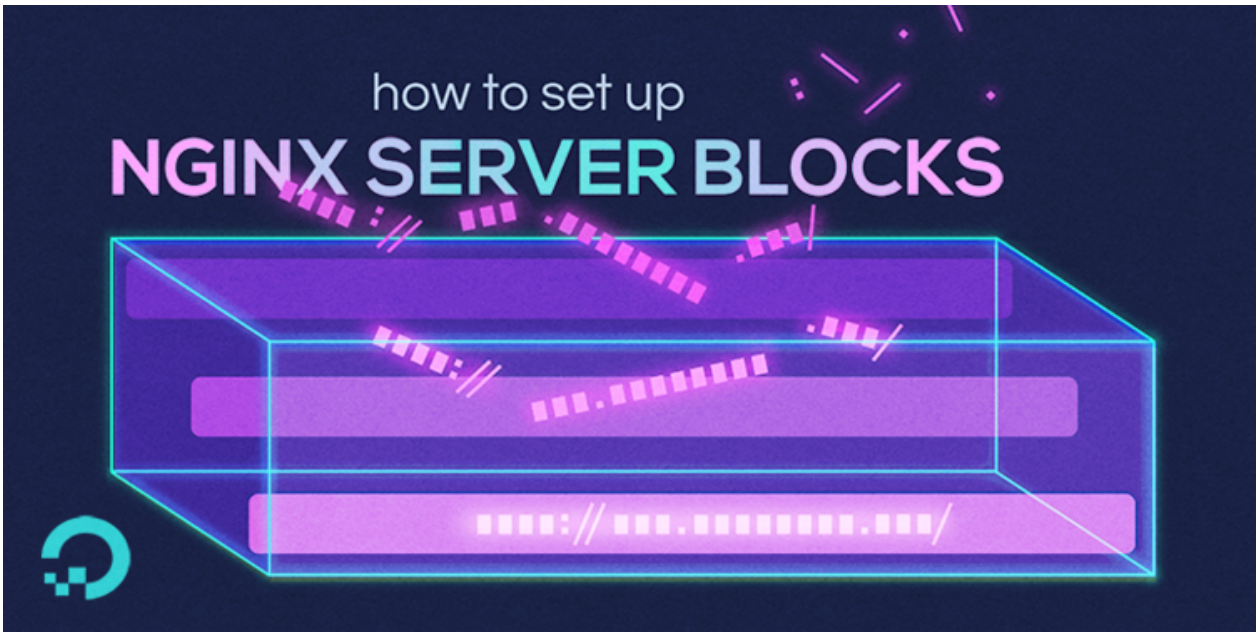
```
        server_name toplevel.com www.toplevel.com;
        return 301 https://$server_name$request_uri;
}

server {
        # SSL configuration

        listen 443 ssl http2 default_server;
        listen [::]:443 ssl http2 default_server;
        include snippets/ssl-toplevel.com.conf;
        include snippets/ssl-params.conf;

        root /var/www/toplevel.com/html;

        # Add index.php to the list if you are using PHP
        index index.php index.html index.htm index.nginx-debian.html;

        server_name toplevel.com www.toplevel.com;

        # other configuration below
}
```

Sub domain in /etc/nginx/sites-available/sub.toplevel.com

```
server {
        listen 80;
        listen [::]:80;
        server_name sub.toplevel.com;
        return 301 https://$server_name$request_uri;
}

server {
        # SSL configuration

        listen 443 ssl http2;
        listen [::]:443 ssl http2;
        include snippets/ssl-toplevel.com.conf;  # NOTE that this is the same file as
        include snippets/ssl-params.conf;

        root /var/www/sub.toplevel.com/html;

        # Add index.php to the list if you are using PHP
        index index.php index.html index.htm index.nginx-debian.html;
```

```
        server_name sub.toplevel.com;

        # other configuration below
}
```

1. I am not an Nginx or encryption expert. I got it working by trying to piece things together from the different tutorials. Any questions you have might be better answered by those that are more experienced than I.



How To Set Up Nginx Server Blocks (Virtual Hosts) on Ubuntu 16.04

by Justin Ellingwood

When using the Nginx web server, server blocks (similar to the virtual hosts in Apache) can be used to encapsulate configuration details and host more than one domain off of a single server. In this guide, we'll discuss how to configure server blocks in Nginx on an Ubuntu...

SummonD *September 21, 2016*

0 Thank you os much. I have problem with this issue and this comment help me. ^ ^

mckennatim *December 29, 2016*

0 do they each get a `location ~ /.well-known {allow all;}`

didimitrie *May 29, 2017*

0 For the sake of posterity, my renewals were failing without that. So I guess the answer would be yes :)

jfandradea *June 15, 2017*

0 Thanks for showing this configuration part. Now my server runs ok with this cleaner example.

gurabli *June 30, 2016*

11 Watch out to the ssl-params.conf where you set X-Frame-Options to DENY. It will completely disable opening a page in frame or iframe, result will be that many reverse proxies, for example Deluge WebUI will not work, just as ownCloud will be limited too.

A common and recommended setting is to set it to SAMEORIGIN, that will still be pretty much safe to prevent Clickjacking, but it will allow functionality.

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

alieus *July 22, 2016*

2 Also, a lot of functions in the WordPress admin panel don't work.

searchwrkcom *August 27, 2016*

0 Thank you! I had an issue with a Gravity Forms admin iframe window displaying a blank screen. Changing DENY to SAMEORIGIN solved the problem.

rejkpp *November 24, 2016*

0 Wow. This `add_header X-Frame-Options DENY;` took me days to find and figure out. It limited functionality of OptimizePress on WordPress.

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

Thanks for the tip. Cheers :)

aarshaw *July 27, 2016*

0 Has anyone else received this error:

```
Had a problem while installing Python packages:
```

When executing the command?

```
./letsencrypt-auto certonly -a webroot --webroot-path=/var/www/html -d example.com -d
```

Followed the steps exactly up to this point, not sure where I'm going wrong...

Load More Comments

Community Tutorials Questions Projects Tags Newsletter RSS

Distros & One-Click Apps Terms, Privacy, & Copyright Security Report a Bug Write for DOnations Shop