Nginx

Collection of Nginx-related posts on my Debian server, might create a separate Book for Nginx in the future.

- Missing /etc/nginx folder
- Hide Nginx version
- Generate DH parameters (dhparam.pem)

Missing /etc/nginx folder

Not entirely sure how this might have happened, but I once SSHed into one of my servers to find out that the entire `/etc/nginx` folder has dissapeared. I tried looking elsewhere or using locate, but the configuration folder was nowhere to be found. Just like that the `/var/log/nginx` folder has dissapeared as well. It might have happened during an upgrade, but after some thorough searching, I still couldn't figure out when or why it happened.

Strangely, all services (websites) using Nginx were running fine and the entire service was active and running as well, still happily pointing to the non-existent config file:

```
$ systemctl status nginx
● nginx.service - nginx - high performance web server
     Loaded: loaded (/lib/systemd/system/nginx.service; disabled; vendor preset: enabled)
    Drop-In: /etc/systemd/system/nginx.service.d
             └─override.conf
     Active: active (running) since Wed 2021-09-15 21:40:43 CEST; 3 days ago
       Docs: https://nginx.org/en/docs/
   Main PID: 98 (nginx)
        CPU: 20.643s
     CGroup: /system.slice/nginx.service
             ├─ 98 nginx: master process /usr/sbin/nginx -c /etc/nginx/nginx.conf
             └─101 nginx: worker process

Sep 15 21:40:43 hostname systemd[1]: Starting nginx - high performance web server...
Sep 15 21:40:43 hostname systemd[1]: Started nginx - high performance web server.
```

```
$ cd /etc/nginx
-bash: cd: /etc/nginx: No such file or directory
```

However, if the server or Nginx randomly restarted, it wouldn't be able to start again and all hell would break loose.

Fortunately, I have backed up my config file, which is really the only thing that matters, the rest can be rebuilt.

Stop Nginx

```
$ sudo systemctl stop nginx
```

Try to restart it, and just like I expected:

```
$ sudo systemctl restart nginx
Job for nginx.service failed because the control process exited with error code.
See "systemctl status nginx.service" and "journalctl -xe" for details.

$ sudo systemctl status nginx
● nginx.service - nginx - high performance web server
     Loaded: loaded (/lib/systemd/system/nginx.service; disabled; vendor preset: enabled)
    Drop-In: /etc/systemd/system/nginx.service.d
             └─override.conf
     Active: failed (Result: exit-code) since Sun 2021-09-19 16:25:01 CEST; 5s ago
       Docs: https://nginx.org/en/docs/
    Process: 22419 ExecStart=/usr/sbin/nginx -c /etc/nginx/nginx.conf (code=exited, status=1/FAILURE)
        CPU: 11ms

Sep 19 16:25:01 hostname systemd[1]: Starting nginx - high performance web server...
Sep 19 16:25:01 hostname nginx[22419]: nginx: [alert] could not open error log file: open()
"/var/log/nginx/error.log" fa>
Sep 19 16:25:01 hostname nginx[22419]: 2021/09/19 16:25:01 [emerg] 22419#22419: open()
"/etc/nginx/nginx.conf" failed (2:>
Sep 19 16:25:01 hostname systemd[1]: nginx.service: Control process exited, code=exited, status=1/FAILURE
Sep 19 16:25:01 hostname systemd[1]: nginx.service: Failed with result 'exit-code'.
Sep 19 16:25:01 hostname systemd[1]: Failed to start nginx - high performance web server.
```

Uninstall and reinstall Nginx

```
$ sudo apt purge nginx*
```

```
$ sudo apt install nginx
```

Fix configuration

Navigate to `/etc/nginx/conf.d` and rename the config file like before.

```
$ cd /etc/nginx/conf.d
$ sudo mv default.conf proxy.conf
```

I'm copying the config through SSH session, so just `$ sudo vi proxy.conf` and `Shift+Ins` and `Esc` and `ZZ` to save and quit.

Test config, final fixes

Test the configuration:

```
$ sudo nginx -t


nginx: [emerg] BIO_new_file("/etc/nginx/dhparam.pem") failed (SSL: error:02001002:system library:fopen:No
such file or directory:fopen('/etc/nginx/dhparam.pem','r') error:2006D080:BIO routines:BIO_new_file:no such file)
nginx: configuration file /etc/nginx/nginx.conf test failed
```

Together with the config folder, my `dhparam.pem` file was also deleted, so we have to generate it again.

Generate secure `dhparam.pem`. This will take a loooooong time, especially on VPSes with a single core like in my case.

```
$ openssl dhparam -out dhparam.pem 4096
```

Test config again, it should be fine now:

```
$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Restart the service and check status to see if everything is working.

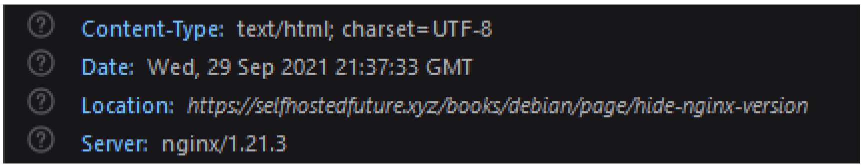
```
$ sudo systemctl restart nginx
```

> Moral of the story – always backup at least your config files

Hide Nginx version

Test if your website sends `Server` header

When you make a request to a Nginx-powered website, by default, every response will contain Nginx's server version in a `Server` header. You can test this by opening *developer options* in your browser (`F12` in Firefox) and looking at the requests in the *Network* tab.



```
Content-Type: text/html; charset=UTF-8
Date: Wed, 29 Sep 2021 21:37:33 GMT
Location: https://selfhostedfuture.xyz/books/debian/page/hide-nginx-version
Server: nginx/1.21.3
```

Other method is to use `curl`:

```
$ curl -IL https://selfhostedfuture.xyz
```

The output shows a similar result:

```
HTTP/1.1 200 OK
Server: nginx/1.21.3
Content-Type: text/html; charset=UTF-8
...
```

Hide Nginx version from `Server` header

Even though exposing the server's Nginx version isn't a huge security threat, it makes it easier for attackers to find exploits and vulnerabilities specifically for the given version. This is especially important if you forget updating your server for a while.

> Hiding Nginx's version is *Security Through Obscurity*, since more advanced attackers are able to find what they want anyway

Edit Nginx configuration

Open the Nginx configuration file in `/etc/nginx/nginx.conf`:

```
$ sudo vi /etc/nginx/nginx.conf
```

By default, your config will look something like this (settings here can be overwritten by configuration in `/etc/nginx/conf.d`):

```
user  nginx;
worker_processes  auto;

error_log  /var/log/nginx/error.log notice;
pid        /var/run/nginx.pid;


events {
    worker_connections  1024;
}


http {
    include       /etc/nginx/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  /var/log/nginx/access.log  main;

    sendfile        on;
    #tcp_nopush     on;

    keepalive_timeout  65;

    #gzip  on;

    include /etc/nginx/conf.d/*.conf;
}
```

Focus on the `http` server block and add the `server_tokens` directive and set it to `off`:

```
server_tokens off;
```

Like this:

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

    server_tokens off;
    ....
```

Test configuration

Before restarting, check that you haven't made a mistake in the configuration:

```
$ sudo nginx -t
```

Proceed if the output looks like this:

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

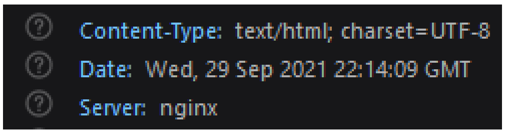
Restart Nginx

Restart Nginx for the changes to take effect:

```
$ sudo systemctl restart nginx
```

Confirm the change

Look into the *Network* tab again, or use `curl` to see whether Nginx still reports the version. It is also possible to completely hide the fact that the website is Nginx powered, but that is for another post.



```
Content-Type: text/html; charset=UTF-8
Date: Wed, 29 Sep 2021 22:14:09 GMT
Server: nginx
```

```
$ curl -IL https://selfhostedfuture.xyz
```

```
HTTP/1.1 200 OK
Server: nginx
Content-Type: text/html; charset=UTF-8
```

Generate DH parameters (dhparam.pem)

One of the things we can do to improve the security of our website is to generate our own DH parameters. What these parameters mean is decently explained HERE.

Genreate dhparam.pem

```
$ openssl dhparam -out dhparam.pem 4096
```

If you are generating directly into the `/etc/nginx` directory (only writable by root), you can use `sudo`, or if you don't want to elevate `openssl` for no reason, just generate the file to a writable location and copy it to `/etc/nginx` later on.

```
$ sudo openssl dhparam -out dhparam.pem 4096
```

> Heads up – This will take a long time, especially on less powerful servers/VMs.

Add it to Nginx config

Open Nginx configuration file and add the path to `dhparam.pem` file under the `server` block.

```
$ sudo vi /etc/nginx/conf.d/your_config_file.conf
```

```
...
ssl_dhparam /etc/nginx/dhparam.pem;
...
```

Test the configuration to avoid mistakes (like forgetting `;`) etc.

```
$ sudo nginx -t
```

Restart Nginx to apply the change:

```
$ sudo systemctl restart nginx
```