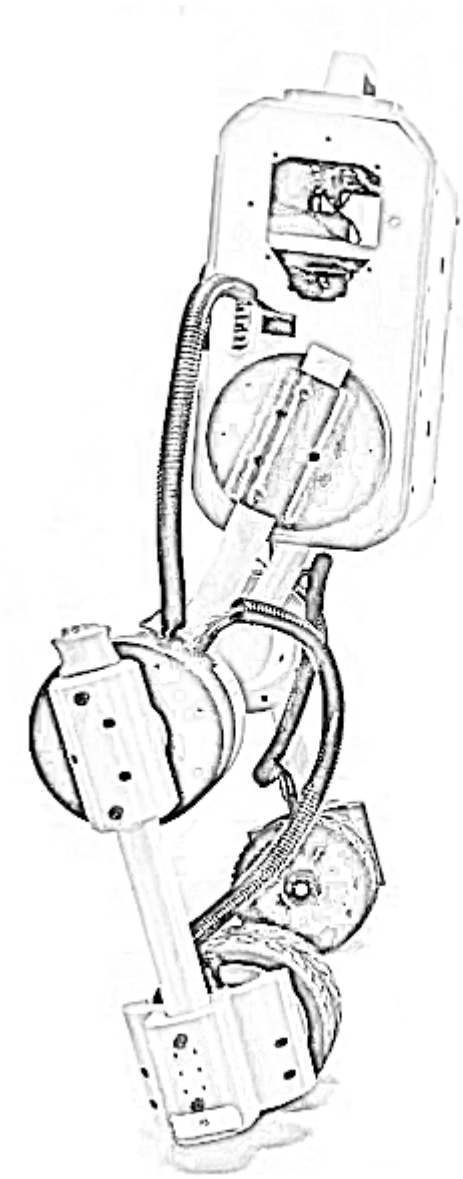

Blog

Ideas

Demos

Publications

Robotics

Talks

古人の跡を求めず、
古人の求めしところを求めよ

Improve your Nginx SSL configuration

This post is mostly a rehash of good advices I found on Ted's blog (Avoir une bonne configuration SSL avec nginx, in French). In a nutshell: go and check your SSL configuration with the Quarlys SSL Server Test.

One way to make dragnet surveillance as expensive as possible is to enable HTTPS by default on all our websites, reducing the amount of cleartext data flowing in the Internet pipes. This is the reason why my Nginx configuration suggests to every visitor the use of HTTPS via:

```
add_header Strict-Transport-Security max-age=63072000;
```

It actually goes as far as redirecting all HTTP traffic to HTTPS ;)

```
server {
    listen 80;
    server_name <my_server_name>;
    return 301 https://$host$request_uri;
}
```

Note that, when you do that, you need to be careful not to lose your SSL keys, as it is impossible to downgrade HTTPS visitors to HTTP without the proper certificates (e.g. once the website is fully referenced in HTTPS). Finally, my default SSL config also included:

```
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;  # drop SSLv3 (POODLE vulnerability)
ssl_session_cache shared:SSL:10m;
ssl_session_timeout 10m;
```

OK, now I had that nice green lock icon showing up in my web browser, but it turns out it was not enough. As I discovered with the Quarlys SSL Server Test, some SSL ciphers like RC4 are vulnerable, and SSL 3 is broken (my initial test score was B-...) Mozilla published a convenient SSL configuration generator, which I used to generate a stronger ciphers list for my server:

```
ssl_prefer_server_ciphers on;
ssl_ciphers 'ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-
```

My next vulnerability according to the test were my weak Diffie-Hellman keys. Diffie-Hellman key exchange is a protocol providing the pretty cool property that, even if some attackers get their hands on your server's private key, it will be exponentially hard for them to decipher the communication between the server and its clients. However, the default key size in OpenSSL is 1024 bits, which seems breakable with the computing power of a nation-state. So, you let's generate some better parameters.

First, generate your DH parameters with OpenSSL:

```
cd /etc/ssl/certs
openssl dhparam -out dhparam.pem 4096
```

Then, add the following to your Nginx configuration:

```
ssl_dhparam /etc/ssl/certs/dhparam.pem;
```

And *voilà*, your configuration just got better. Another feature you may want to enable is OCSP stapling. Finally, a little icing on the cake:

```
add_header X-Frame-Options DENY;
```

This option tells browsers that my website should not be displayed inside a `<frame>`, an `<iframe>` or an `<object>`. Just in case someone goes phishing.

Discussion

There are no comments yet. Feel free to leave a reply using the form below.

Post a comment

You can use Markdown with `$\LaTeX$` formulas in your comment.

Name Your name or nickname

Website Your personal website (optional)

Comment Your question or comment

You agree to the publication of your comment on this page under the CC BY 4.0 license.

Post via email Your email address will not be published.