NodeJS and HTTPS

Markus Veijola

March 2014

opiframe

Introduction

- **Hypertext Transfer Protocol Secure (HTTPS)** is a communications protocol for secure communication over a computer network, with especially wide deployment on the Internet.
- Technically, it is not a protocol in and of itself; rather, it is the result of simply layering the Hypertext Transfer Protocol (HTTP) on top of the SSL/TLS protocol, thus adding the security capabilities of SSL/TLS to standard HTTP communications.
- The main motivation for HTTPS is to prevent wiretapping and man-in-the-middle attacks.

Introduction

- X.509 certificates are used to guarantee one is talking to the partner with whom one wants to talk. As a consequence, certificate authorities and a public key infrastructure are necessary to verify the relation between the owner of a certificate and the certificate, as well as to generate, sign, and administer the validity of certificates.

Introduction

- First of all we need associated SSL certificates for our HTTPS web server.
- The recommended way is to get your certificate signed by a Certificate Authority, but for testing purposes we will sign it ourselves.
- Windows users will need Cygwin tool to generate these files (or use IIS server). You can find Cygwin from here: https://www.cygwin.com/
- After downloading install the Cygwin package.

Generating The Keys

- Open Cygwin terminal.
- We need to install openssl library.
- To do this first install apt-cyg with next commands:

```
svn --force export http://apt-cyg.googlecode.com/svn/trunk/ /bin/
chmod +x /bin/apt-cyg
```

Generating The Keys

- Then install opessl:

apt-cyg install openssl

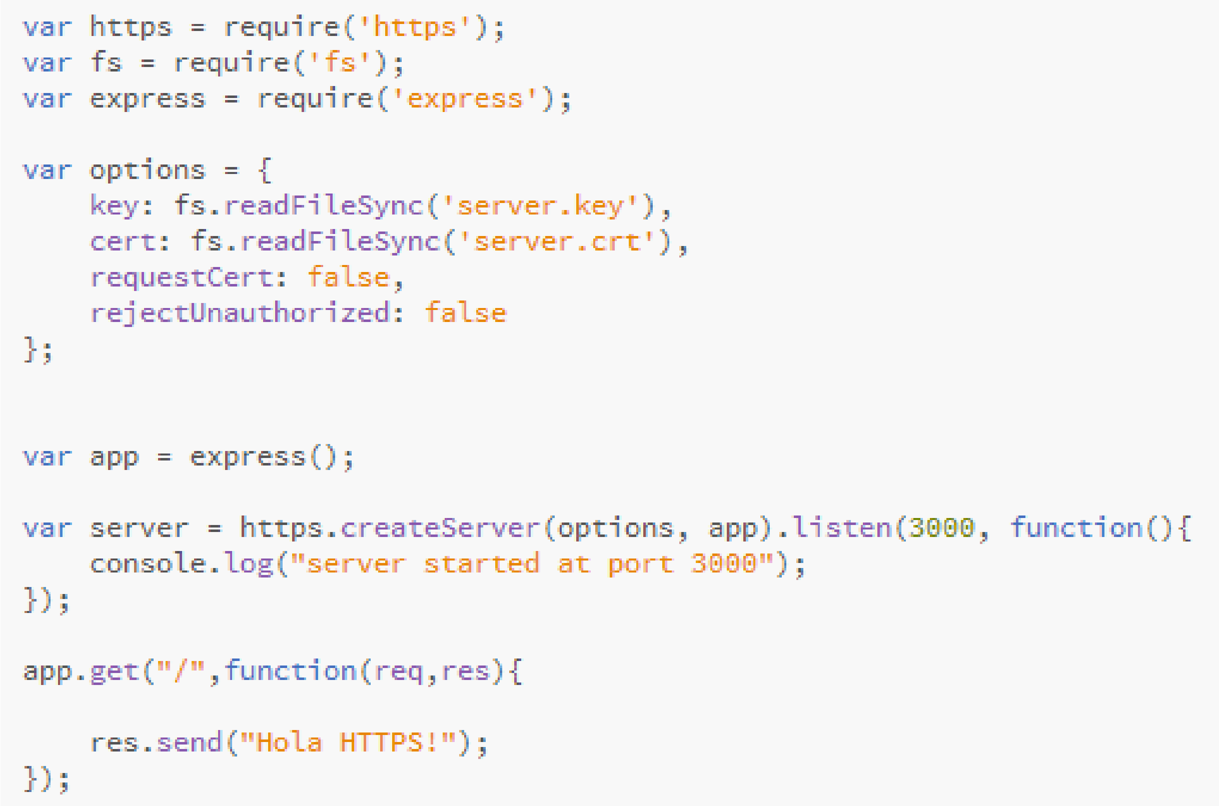
- Then generate the keys with next commands:

```
openssl genrsa -des3 -out server.key 1024
openssl req -new -key server.key -out server.csr
cp server.key server.key.org
openssl rsa -in server.key.org -out server.key
openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```

Generating The Keys

- Now create a working directory for your HTTPS server.
- Copy the generated files from C:\cygwin64\home\Opiframe to your working directory.
- In your working directory create app.js file with next content.

Testing



```
var https = require('https');
var fs = require('fs');
var express = require('express');

var options = {
    key: fs.readFileSync('server.key'),
    cert: fs.readFileSync('server.crt'),
    requestCert: false,
    rejectUnauthorized: false
};


var app = express();

var server = https.createServer(options, app).listen(3000, function(){
    console.log("server started at port 3000");
});

app.get("/",function(req,res){

    res.send("Hola HTTPS!");
});
```

Testing

- Start the server.
- Open browser and enter next url: https://localhost:3000
- You should see that browser is complaining about certificate, but make an exception rule for this domain (because this is just for testing purposes).
- Then you should see the text “Hola HTTPS!” in browser window.