Vanja Komadinovic

~ interesting stuff

Client and Server side SSL with NodeJS

11 *Thursday* Aug 2011

Posted by Vanja Komadinovic in development, javascript, node.js

≈ 2 Comments

Tags

authentication, both, CA, client side, https, javascript, nodejs, server side, ssl, tis

My idea was to create simple nodeJS server which will work behind SSL, this is pretty simple to do, but I also want to authenticate clients with SSL certificate. I looked a bit on net and found what I think it's solution.

We need one Certificate Authority certificate which will be used for signing of all other certificates. After that we will create server and users certificates. In code for server we will specify that certificates are required and that trusted CA certificate is same certificate used for all users signing. Rest is magic 😃

Create CA certificate, self sign and of course test:

```
openssl genrsa -des3 -out ca.key 2048
openssl req -new -x509 -days 365 -key ca.key -out ca.crt
openssl x509 -in ca.crt -text -noout
```

Create server certificate, request signing, sign with our CA and test:

```
openssl genrsa -out server.key 1024
openssl req -new -key server.key -out server.csr
openssl x509 -req -in server.csr -out server.crt -CA ca.crt -CAkey ca.key -CAcreateserial -days 365
openssl x509 -in server.crt -text -noout
```

Create userA certificate, request signing, sign with our CA and test:

```
openssl genrsa -out userA.key 1024
openssl req -new -key userA.key -out userA.csr
openssl x509 -req -in userA.csr -out userA.crt -CA ca.crt -CAkey ca.key -CAcreateserial -days 365
openssl x509 -in userA.crt -text -noout
```

Do same thing for one more user:

```
openssl genrsa -out userB.key 1024
openssl req -new -key userB.key -out userB.csr
openssl x509 -req -in userB.csr -out userB.crt -CA ca.crt -CAkey ca.key -CAcreateserial -days 365
openssl x509 -in userB.crt -text -noout
```

Clean up:

```
rm *.csr
mkdir keys certs ca
mv ca.* ca/
mv *.key keys/
mv *.crt certs/
```

If we want to use browser instead of nodeJS client for connection to server certificates must be transferred to p12 format (p12 contains both key and certificate)

```
openssl pkcs12 -export -in certs/userA.crt -inkey keys/userA.key -name "User A BusyWait test cert" -out userA.p12
open userA.p12
```

Now, we need server to handle https requests. We will create simple server on port 8000 that will return `Hello World, name` for all clients with correct certificate (certificate signed by CA that sever trusts, in this case this will be our CA).

```
var sys = require("sys");
var fs = require("fs");
var https = require("https");

var options = {
  key: fs.readFileSync("keys/server.key"),
  cert: fs.readFileSync("certs/server.crt"),
```

```
  ca: fs.readFileSync("ca/ca.crt"),
  requestCert: true,
  rejectUnauthorized: true
};

https.createServer(options, function (req, res) {
  res.writeHead(200);
  sys.puts("request from: " + req.connection.getPeerCertificate().subject.CN);
  res.end("Hello World, " + req.connection.getPeerCertificate().subject.CN + "\n");
}).listen(8000);

sys.puts("server started");
```

Code for client is also pretty simple:

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

var options = {
        host: 'localhost',
        port: 8000,
        path: '/test',
        method: 'GET',
        key: fs.readFileSync("keys/userB.key"),
        cert: fs.readFileSync("certs/userB.crt"),
        ca: fs.readFileSync("ca/ca.crt")
};

var req = https.request(options, function(res) {
        console.log("statusCode: ", res.statusCode);
        console.log("headers: ", res.headers);

        res.on('data', function(d) {
        process.stdout.write(d);
        });
});

req.end();

req.on('error', function(e) {
        console.error(e);
});
```

To try this, download sample code from git, link bellow, and run `node server.js`. Server should be started. In other terminal start client `node client.js`. To change user certificate used change code in client for certificate files.

All code can be found on my GitHub.

Some useful links on net:
Van's Apache SSL/TLS mini-HOWTO
Accessing the client certificate in TLS/HTTPS
NodeJS https documentation
NodeJS tls documentation

2 thoughts on "Client and Server side SSL with NodeJS"

1. Pingback: HTTPS client for iOS « Vanja Komadinovic
2. Roi *said:*

   May 16, 2012 at 14:52

   when I ran the code on my nodejs (6.1.7) it seams that the https client code doesn't verify the server certificate against the CA

   Reply