Criação de certificados SSL

1 – Instalação do OpenSSL:
http://www.openssl.org/
http://gnuwin32.sourceforge.net/packages/openssl.htm

2 – Criação da chave privada da autoridade certificadora (CA), de 2048 Bits:
openssl genrsa 2048 > ca-key.pem

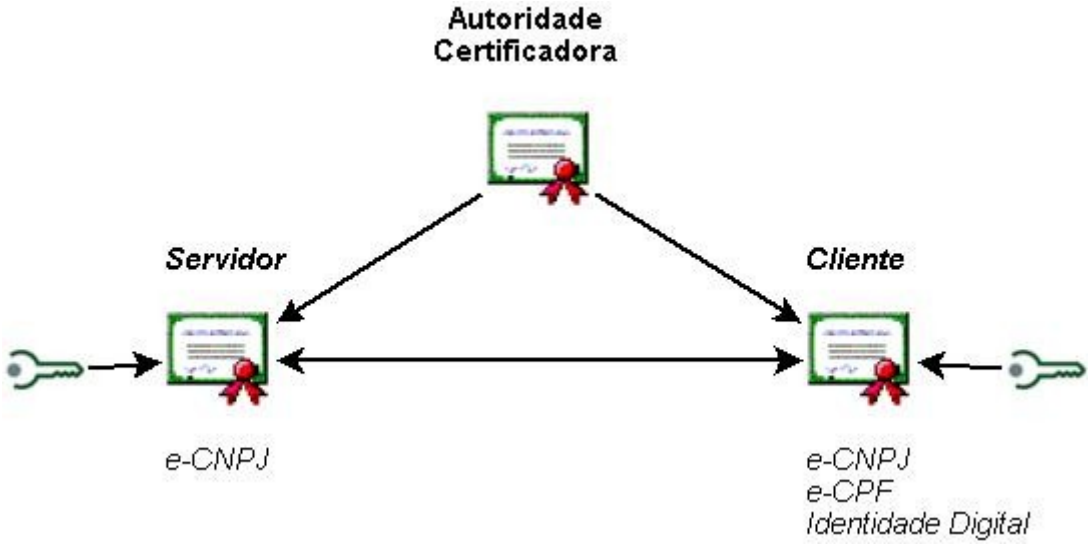
3 – Criação do certificado (chave pública, padrão X.509, validade de 1.000 dias e outras informações) da autoridade certificadora (CA), através da chave privada:
(Linux)
openssl req -new -x509 -nodes -days 1000 -key ca-key.pem > ca-cert.pem
(Windows)
openssl req -new -x509 -nodes -days 1000 -key ca-key.pem -config openssl.cnf > ca-cert.pem

Obs: O arquivo openssl.cnf para Windows, está disponível dentro do pacote Sources, e deverá ser copiado no mesmo diretório do comando **openssl**.

4 – Criação da chave privada do usuário e requisição de assinatura, através do CA:
(Linux)
openssl req -newkey rsa:2048 -days 1000 -nodes -keyout usuario-key.pem > usuario-req.pem
(Windows)
openssl req -newkey rsa:2048 -days 1000 -nodes -keyout usuario-key.pem -config openssl.cnf > usuario-req.pem

5 – Criação do certificado do usuário, através do CA e requisição de assinatura:
openssl x509 -req -in usuario-req.pem -days 1000 -CA ca-cert.pem -CAkey ca-key.pem -set_serial 01 > usuario-cert.pem

6 – Criação de um segundo usuário, através da mesma CA:
É só seguir os mesmos passos de 4 e 5. Assim os dois usuários poderão estabelecer uma conexão segura, certificado pela autoridade CA.



Acesso seguro ao MySQL 5.0

1 – Criar obter ou criar um par de chaves e certificados válidos:
Pode-se utilizar dois e-CNPJ ou criar as chaves e certificados, para haver a comunicação. Para o servidor MySQL será necessário o certificado da autoridade certificadora *(ca-cert.pem)*, um certificado *(server-cert.pem)* e uma chave privada *(server-key.pem)* válidos.
Para o cliente MySQL será necessário o certificado da autoridade certificadora *(ca-cert.pem)*, um certificado *(client-cert.pem)* e uma chave privada *(client-key.pem)* válidos, diferentes do servidor.

2 – Configuração do servidor:
(Linux)
Arquivo /etc/mysql/my.cnf:
[mysqld]
ssl-ca = /etc/mysql/ca-cert.pem
ssl-cert = /etc/mysql/server-cert.pem
ssl-key = /etc/mysql/server-key.pem

(Windows)
Arquivo C:\Arquivos de programas\MySQL\MySQL Server 5.0\my.ini:
[mysqld]
ssl-ca = "C:/Arquivos de programas/MySQL/MySQL Server 5.0/ssl/ca-cert.pem"
ssl-cert = "C:/Arquivos de programas/MySQL/MySQL Server 5.0/ssl/server-cert.pem"
ssl-key = "C:/Arquivos de programas/MySQL/MySQL Server 5.0/ssl/server-key.pem"

3 – Reiniciar o servidor e confirmar se está funcionando:

(Linux)
mysqladmin shutdown
mysqld&
mysql -u root -p
mysql> SHOW VARIABLES LIKE 'have_openssl';

```
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| have_openssl  | YES   |
+---------------+-------+
```

(Windows XP/2000)
net stop mysql
net start mysql
mysql -u root -p
mysql> SHOW VARIABLES LIKE 'have_openssl';

```
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| have_openssl  | YES   |
+---------------+-------+
```

4 – Criar um usuário com acesso seguro:
*mysql>GRANT ALL PRIVILEGES ON *.* TO 'usuario'@'%' REQUIRE CIPHER 'DHE-RSA-AES256-SHA' AND SUBJECT '/C=BR/ST=SP/L=Campinas/O=Cliente/CN=Cliente' AND ISSUER '/C=BR/ST=SP/L=Campinas/O=Autoridade/CN=Autoridade';FLUSH PRIVILEGES;*

Este comando cria um usuário com acesso quase total, de qualquer IP e sem senha. Mas seu certificado deverá seguir os parâmetros de *ISSUER*, a autoridade certificadora deverá seguir os parâmetros de *SUBJECT*, e ambos deverão ser capazes de utilizar a cifra definida em *CIPHER*.

5 – Acesso seguro:
mysql -u usuario --ssl-ca=ca-cert.pem --ssl-cert=client-cert.pem –ssl-key=client-key.pem

Os arquivos cert-cert.pem, client-cert.pem e client-key.pem deverão estar no mesmo diretório do comando **mysql**.
Observe que o arquivo ca-cert.pem é o mesmo para o servidor e o usuário.