Connect and Replicate Securely: How to use MySQL with SSL

Sheeri K. Cabral, MySQL Team Lead
cabral@pythian.com

Is SSL Enabled?

```
mysql> SHOW VARIABLES LIKE '%ssl%';
+---------------+----------+
| Variable_name | Value    |
+---------------+----------+
| have_openssl  | DISABLED |
| have_ssl      | DISABLED |
| ssl_ca        |          |
| ssl_capath    |          |
| ssl_cert      |          |
| ssl_cipher    |          |
| ssl_key       |          |
+---------------+----------+
7 rows in set (0.00 sec)
```

Enable SSL

yaSSL w/ official MySQL binary

To compile your own or w/OpenSSL:

```
./configure --with-ssl
```

```
./configure --with-ssl=/path/to/ssl
```

Enable SSL In >5.1.10

```
./configure --with-yassl
            --with-openssl

configure --with-*ssl=/path/to/ssl
```

SSL Enabled

```
mysql> SHOW VARIABLES LIKE '%ssl%';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| have_openssl  | YES   |
| have_ssl      | YES   |
| ssl_ca        |       |
| ssl_capath    |       |
| ssl_cert      |       |
| ssl_cipher    |       |
| ssl_key       |       |
+---------------+-------+
7 rows in set (0.00 sec)
```

Can mysql.user handle it?

```
SELECT COLUMN_NAME
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME='user'
AND TABLE_SCHEMA='mysql'
AND COLUMN_NAME IN
    ('ssl_cipher','x509_issuer','x509_subject');

+---------------+
| COLUMN_NAME   |
+---------------+
| ssl_cipher    |
| x509_issuer   |
| x509_subject  |
+---------------+
3 rows in set (0.02 sec)
```

Create Certificate Authority

Used to sign certificates

```
openssl req -new -x509 -keyout cakey.pem \
-out cacert.pem -days 3600 -config openssl.cnf

Sample output:
Using configuration from openssl.cnf
Generating a 1024 bit RSA private key
..................++++++
.........++++++
writing new private key to 'cakey.pem'
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
```

Info for CA

```
You are about to be asked to enter information that
 will be incorporated into your certificate request.
What you are about to enter is what is called a
 Distinguished Name or a DN.
There are quite a few fields but you can leave some
 blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

Info for CA

```
-----
Country Name (2 letter code) [GB]:US
State or Province Name (full name) [Some-
 State]:Massachusetts
Locality Name (eg, city) []:Cambridge
Organization Name (eg, company) [Internet Widgits Pty
 Ltd]:The Pythian Group
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:certs.pythian.com
Email Address []:cabral@pythian.com
```

Create Server Key/Request

```
openssl req -new -keyout server-key.pem -out \
   server-req.pem -days 3600 -config openssl.cnf

Sample output:
Using configuration from openssl.cnf
Generating a 1024 bit RSA private key
..++++++
..........++++++
writing new private key to 'server-key.pem'
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
```

Info for Server Request

```
-----
Country Name (2 letter code) [GB]:US
State or Province Name (full name) [Some-
 State]:Massachusetts
Locality Name (eg, city) []:Cambridge
Organization Name (eg, company) [Internet Widgits Pty
 Ltd]:The Pythian Group
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:db.pythian.com
Email Address []:cabral@pythian.com
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Remove Key Passphrase

```
openssl rsa -in server-key.pem -out server-key.pem
```

Create Certificate

Sign the server certificate request
using Certificate Authority generated

```
openssl ca -policy foo -out server-cert.pem \
    -config openssl.cnf -infiles server-req.pem

Using configuration from openssl.cnf
Enter PEM pass phrase:
Check that the request matches the signature
Signature ok
```

Sign the Request

```
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'US'
organizationName      :PRINTABLE:'The Pythian Group'
commonName            :PRINTABLE:'db.pythian.com'
Certificate is to be certified until Apr 01
 14:22:46 2010 GMT
(365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit?
 [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Create Client Request

```
openssl req -new -keyout client-key.pem -out \
   client-req.pem -days 3600 -config openssl.cnf
```

Same as Server certificate

Including removing the passphrase

Sign Client Certificate

```
openssl ca -policy foo -out client-cert.pem \
   -config openssl.cnf -infiles client-req.pem
```

Edit Configs

```
[client]
ssl-ca=/path/to/cacert.pem
ssl-cert=/path/to/client-cert.pem
ssl-key=/path/to/client-key.pem
```

```
[mysqld]
ssl-ca=/path/to/cacert.pem
ssl-cert=/path/to/server-cert.pem
ssl-key=/path/to/server-key.pem
```

User GRANT

```
GRANT ... ON ... TO ... REQUIRE

SSL   (--ssl-ca required)
X509  (--ssl ca, --ssl-key, --ssl-cert, not checked)
CIPHER 'cipher'  (strength)
ISSUER 'issuer'  (issuer)
```

Connect

If you REQUIRE anything, SSL **must** be used.

Secure Replication

Use a secure user....

No really, that's it!!!

cabral@pythian.com