Mysql 8, SSL con Let’s Encrypt

Lo mejor es trabajar en un entorno seguro.

La aparición de nuevas versiones de muchos softwares, que fuerzan al uso de una seguridad mayor, sobre todo en las comunicaciones entre nodos, habida cuenta del auge de los microservicios, y la amplitud de posibilidades para este tema, nos lleva a un camino ya conocido: la desactivación de la seguridad en cuanto surge algo que nos molesta, en lugar de aprender el camino del cómo hacerlo con el método seguro.

En el anterior artículo QGIS, Navicat, y otros problemas de conexión con MySQL 8 hablamos de una salida rápida, pero la realidad es que es mejor dedicar un tiempo y hacer la correcta.

Aquí te explico cómo securizar con SSL de Let’s Encrypt un servidor MySQL 8.X, aunque estoy seguro de que te vale con algún matiz, para MariaDb o Percona.

Instalando Let’s Encrypt en MySQL

No es el alcance explicar cómo se obtiene un certificado Let’s Encrypt en un servidor pero te dejo algunos tips, al final del documento.

Entendemos para este artículo que tenemos un servidor o un sistema de certificados para el hostname basado en Let’s Encrypt, ya está ya instalado y que se renueva regularmente.

Ubicación de los certificados Let’s Encrypt

Esto es sólo un ejemplo, para entenderlo. En realidad debes tener claro dónde está en tu servidor.

```
# ls  -lisa /etc/letsencrypt/live/<mydomain.tld>/
total 12
428663 4 drwxr-xr-x 2 root root 4096 sep 15 14:32 .
```

```
428659 4 drwx------ 3 root root 4096 mar  3  2022 ..
399891 0 lrwxrwxrwx 1 root root   54 sep 15 14:32 cert.pem ->
../../archive/mydomain.tld/cert5.pem
399898 0 lrwxrwxrwx 1 root root   55 sep 15 14:32 chain.pem ->
../../archive/mydomain.tld/chain5.pem
399930 0 lrwxrwxrwx 1 root root   59 sep 15 14:32 fullchain.pem ->
../../archive/mydomain.tld/fullchain5.pem
399892 0 lrwxrwxrwx 1 root root   57 sep 15 14:32 privkey.pem ->
../../archive/mydomain.tld/privkey5.pem
```

El problema es que estando en esa ubicación, mediante enlaces a los ficheros reales, correspondientes a cada una de las versiones (renovaciones) no conseguí que funcionaran.

Así que vamos a solucionarlo

Despliegue de los certificados Let’s Encrypt para MySQL

Configuración mysql

Configurar el fichero `/etc/my.cnf` o el usado por su distribución. En mi caso para una Ubuntu 20.04 `/etc/mysql/mysql.conf.d/mysqld.cnf`

```
[mysqld]
ssl_ca=/var/lib/mysql/chain.pem
ssl_cert=/var/lib/mysql/cert.pem
ssl_key=/var/lib/mysql/privkey.pem
```

Crear los ficheros de los certificados

Notas y descargo de responsabilidad

- El dominio se refiere al hostname de la máquina
- Las otras variables debe ser revisadas adaptándose a el servidor que estamos configurando
- No es copiar y pegar, sino una guia que debemos comprender.ç
- No soy responsable de nada de lo que te pase.
- Por su puesto debes tener backup

```
# domain=mydomain.tld
# cert_dir=/var/lib/mysql
# user=mysql.mysql
# cp /etc/letsencrypt/live/$domain/cert.pem $cert_dir
# cp /etc/letsencrypt/live/$domain/privkey.pem $cert_dir
```

```
# openssl x509 -in /etc/letsencrypt/live/$domain/chain.pem > $cert_dir/chain.pem
# chown $user $cert_dir/*.pem
# chmod 600 $cert_dir/*.pem
# mysql --login-path=root@localhost --execute="ALTER INSTANCE RELOAD TLS"
```

Si todo ha ido bien, no saldrá ningún mensaje de error, y MySQL 8.0 ya está preparado para usar SSL en su trabajo.

> Entendemos que tenemos y usamos un fichero ~/.my.cnf para acceder sin password como root a mysql

Comprobación MySQL

```
mysql > SHOW VARIABLES LIKE '%ssl%';
+---------------------------------+----------------------------+
| Variable_name                   | Value                      |
+---------------------------------+----------------------------+
| admin_ssl_ca                    |                            |
| admin_ssl_capath                |                            |
| admin_ssl_cert                  |                            |
| admin_ssl_cipher                |                            |
| admin_ssl_crl                   |                            |
| admin_ssl_crlpath               |                            |
| admin_ssl_key                   |                            |
| have_openssl                    | YES                        |
| have_ssl                        | YES                        |
| mysqlx_ssl_ca                   |                            |
| mysqlx_ssl_capath               |                            |
| mysqlx_ssl_cert                 |                            |
| mysqlx_ssl_cipher               |                            |
| mysqlx_ssl_crl                  |                            |
| mysqlx_ssl_crlpath              |                            |
| mysqlx_ssl_key                  |                            |
| performance_schema_show_processlist | OFF                        |
| ssl_ca                          | /var/lib/mysql/chain.pem   |
| ssl_capath                      |                            |
| ssl_cert                        | /var/lib/mysql/cert.pem    |
| ssl_cipher                      |                            |
| ssl_crl                         |                            |
```

```
| ssl_crlpath                       |                            |
| ssl_fips_mode                     | OFF                        |
| ssl_key                           | /var/lib/mysql/privkey.pem |
| ssl_session_cache_mode            | ON                         |
| ssl_session_cache_timeout         | 300                        |
+-----------------------------------+----------------------------+
27 rows in set (0,00 sec)
```

Modificación del script de renovación de Let’s Encrypt

Obtenemos el path de de los hooks de Let’s Encrypt.

```
## Versión de certbot recomendada con snap, pero puede variar si usamos package de la distro o
binario. Por eso buscamos
# systemctl | grep cer
  snap-certbot-
2344.mount                                                                    loaded
active mounted   Mount unit for certbot, revision 2344
  snap-certbot-
2414.mount                                                                    loaded
active mounted   Mount unit for certbot, revision 2414

snap.certbot.renew.timer
   loaded active waiting   Timer renew for snap application certbot.renew
#  systemctl status snap.certbot.renew.timer
● snap.certbot.renew.timer - Timer renew for snap application certbot.renew
     Loaded: loaded (/etc/systemd/system/snap.certbot.renew.timer; enabled; vendor preset:
enabled)
     Active: active (waiting) since Tue 2022-10-04 20:35:18 UTC; 1 weeks 2 days ago
    Trigger: Fri 2022-10-14 11:31:00 UTC; 1h 53min left
   Triggers: ● snap.certbot.renew.service

Warning: journal has been rotated since unit was started, output may be incomplete.
```

Y después creamos el fichero `/etc/letsencrypt/renewal-hooks/deploy/mysqld-deploy.sh` que se encargará de ajustar tras la renovación los ficheros para MySQL.

```
#!/bin/sh
domain=mydomain.tld
cert_dir=/var/lib/mysql
user=mysql.mysql
```

```
cp /etc/letsencrypt/live/$domain/cert.pem $cert_dir
cp /etc/letsencrypt/live/$domain/privkey.pem $cert_dir

# Only keep 1st certificate (C=US/O=Let's Encrypt/CN=R3), that is, get rid
# of 2nd certificate "ISRG Root X1" issued by "DST Root CA X3" which is expired.
# https://letsencrypt.org/2020/12/21/extending-android-compatibility.html

openssl x509 -in /etc/letsencrypt/live/$domain/chain.pem > $cert_dir/chain.pem
chown $user $cert_dir/*.pem
chmod 600 $cert_dir/*.pem
mysql --login-path=root@localhost --execute="ALTER INSTANCE RELOAD TLS"
```

Y le damos permisos de ejecución

```
chmod 755 /etc/letsencrypt/renewal-hooks/deploy/mysqld-deploy.sh
```

Comprobación

```
openssl s_client -starttls mysql -showcerts -connect mydomain.tld:3306
CONNECTED(00000003)
depth=2 C = US, O = Internet Security Research Group, CN = ISRG Root X1
verify return:1
depth=1 C = US, O = Let's Encrypt, CN = R3
verify return:1
depth=0 CN = api.mydomain.tld
verify return:1
---
Certificate chain
 0 s:CN = api.mydomain.tld
   i:C = US, O = Let's Encrypt, CN = R3
-----BEGIN CERTIFICATE-----
MIIFZjCCBE6gAwIBAgISBJPIsuc3JOh84ALWyVAXpFjxMA0GCSqGSIb3DQEBCwUA
…
…
BpPXg5qzChYB5e2/wRhXRZb3IejNUDg8tQzU3hL6sJPcNtwnYAFyxDcg
-----END CERTIFICATE-----
 1 s:C = US, O = Let's Encrypt, CN = R3
   i:C = US, O = Internet Security Research Group, CN = ISRG Root X1
-----BEGIN CERTIFICATE-----
MIIFFjCCAv6gAwIBAgIRAJErCErPDBinU/bWLiWnX1owDQYJKoZIhvcNAQELBQAw
```

```
…
…
MldlTTKB3zhThV1+XWYp6rjd5JW1zbVWEkLNxE7GJThEUG3szgBVGP7pSWTUTsqX
nLRbwHOoq7hHwg==
-----END CERTIFICATE-----
---
Server certificate
subject=CN = api.mydomain.tld

issuer=C = US, O = Let's Encrypt, CN = R3

---
No client certificate CA names sent
Requested Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:Ed25519:Ed448:RSA-
PSS+SHA256:RSA-PSS+SHA384:RSA-PSS+SHA512:RSA-PSS+SHA256:RSA-PSS+SHA384:RSA-
PSS+SHA512:RSA+SHA256:RSA+SHA384:RSA+SHA512:ECDSA+SHA224:RSA+SHA224
Shared Requested Signature Algorithms:
ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:Ed25519:Ed448:RSA-PSS+SHA256:RSA-PSS+SHA384:RSA-
PSS+SHA512:RSA-PSS+SHA256:RSA-PSS+SHA384:RSA-PSS+SHA512:RSA+SHA256:RSA+SHA384:RSA+SHA512
Peer signing digest: SHA256
Peer signature type: RSA-PSS
Server Temp Key: X25519, 253 bits
---
SSL handshake has read 3415 bytes and written 468 bytes
Verification: OK
---
New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Server public key is 2048 bit
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 0 (ok)
---
---
Post-Handshake New Session Ticket arrived:
SSL-Session:
    Protocol  : TLSv1.3
    Cipher    : TLS_AES_256_GCM_SHA384
```

```
    Session-ID: AD8278545269458B04E87DF04C50140678DEE6C2E2A5BC9017329D1D170A80B1
    Session-ID-ctx:
    Resumption PSK:
DFB2E7716E7B0579E911AB32999D976CBA419B43C0F2A69FAB68D7C13DF7E52FF58A43E3709F62B7E80E6269707AD0
02
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 300 (seconds)
    TLS session ticket:
    0000 - 9b 98 52 5a 9f bc be 7c-b9 37 3a 3a e7 bf 49 7c   ..RZ...|.7::..I|
    0010 - c9 de 90 6b 1d 08 c8 9e-f7 dc c4 04 c6 e4 48 41   ...k..........HA
    …
    …
    00b0 - 09 68 bb 27 18 d3 3e f6-2e e9 f4 6e ee 3f 49 26   .h.'..>....n.?I&
    00c0 - 7c 55 be 6b 46 1b 3c b3-0d 72 d4 93 da 7e 6f c2   |U.kF.<..r...~o.

    Start Time: 1665740692
    Timeout   : 7200 (sec)
    Verify return code: 0 (ok)
    Extended master secret: no
    Max Early Data: 0
---
read R BLOCK
---
Post-Handshake New Session Ticket arrived:
SSL-Session:
    Protocol  : TLSv1.3
    Cipher    : TLS_AES_256_GCM_SHA384
    Session-ID: 2C2F40E7309289A3368A250BC710F04021E66B9D821DFA74694AACFE05E9E742
    Session-ID-ctx:
    Resumption PSK:
DE4C98072B6A8DC2A3636ADFBDF3612090C3C1A5BCB1CD267C1B688015F12FAE5946F2541FB0A428BEDE5483E1DA7A
46
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    TLS session ticket lifetime hint: 300 (seconds)
    TLS session ticket:
    0000 - 9b 98 52 5a 9f bc be 7c-b9 37 3a 3a e7 bf 49 7c   ..RZ...|.7::..I|
```

```
    0010 - 40 00 eb f6 2b 3e 84 fc-4a 7d 6e 00 e7 96 af ce   @...+>..J}n.....
    …
    …
    00b0 - 14 4a 48 3e 33 4b 19 b4-df 14 24 bb 28 bc 55 29   .JH>3K....$.(.U)
    00c0 - 73 23 37 e2 3c e7 0b ea-ed 25 5e 3d 28 cc b5 0d   s#7.<....%^=(...

    Start Time: 1665740692
    Timeout   : 7200 (sec)
    Verify return code: 0 (ok)
    Extended master secret: no
    Max Early Data: 0
---
read R BLOCK
```

Enlaces externos

- Mysql :: Using Encrypted connections
- Myswl 8 :: Alter User
- Certificado Let’s Encrypt para servidor sin servidor web (ElasticSearch)
- MySQL And Lets Encrypt
- Cron job for let's encrypt renewal

Aviso

Esta documentación y su contenido, no implica que funcione en tu caso o determinados casos. También implica que tienes conocimientos sobre lo que trata, y que en cualquier caso tienes copias de seguridad. El contenido el contenido se entrega, tal y como está, sin que ello implique ningún obligación ni responsabilidad por parte de Castris

Si necesitas soporte profesional puedes contratar con Castris soporte profesional.

Revision #2
Created 2022-10-14 11:49:53 UTC by Abkrim
Updated 2022-10-14 11:57:43 UTC by Abkrim