QGIS Application - Feature request #8180
Encryption of Passwords in qgs files

2013-06-27 05:15 AM - Jonathan Moules

Status:	Open		
Priority:	Normal		
Assignee:			
Category:	Project Loading/Saving		
Pull Request or Patch supplied:	No	**Resolution:**	
Easy fix?:	No	**Copied to github as #:**	16999

Description

At least with Oracle (and the others? I have no way to test), when a project is stored as a qgs file, the password is stored in plaintext:

```
&lt;datasource&gt;dbname='co_gislive.world' host=10.76.112.32 port=1521 user='OSMM'
password='password_here_in_plain_text' estimatedmetadata=true srid=27700 type=POLYGON
table="OSMM"."AGRICULTURE_MIDLANDS" (SDO_GEOMETRY) sql=&lt;/datasource&gt;
```

It should be encrypted, probably using AES-256 and a unique salt I guess. I don't know if this carries for other passwords stored by QGIS, either database or WMS/WFS etc - ideally they should all be encrypted.

Using Master 226c524

Related issues:

Related to QGIS Application - Bug report # 9030: WMS passwords stored in plai...	**Closed**	**2013-11-07**
Duplicates QGIS Application - Feature request # 4823: Add warning when saving...	**Closed**	**2012-01-16**
Duplicated by QGIS Application - Feature request # 17009: Encypted password i...	**Closed**	**2017-08-11**

History

#1 - 2013-06-27 05:22 AM - Jürgen Fischer

You don't need to save username/passwords. If you do they'll be unencrypted (and you are warned about that), if you don't QGIS prompts for credentials, when it needs them.

#2 - 2013-06-27 05:24 AM - Jürgen Fischer

- Tracker changed from Bug report to Feature request

#3 - 2013-06-27 06:21 AM - Jonathan Moules

Fair point. But then I'd have to share the database password with the users which is very sub-optimal for a whole raft of reasons. I have no issue with storing the password if required (as it will be for us), but it should definitely be encrypted.

#4 - 2013-06-29 01:31 AM - Jürgen Fischer

- Target version set to Future Release - High Priority

#5 - 2013-08-12 02:03 AM - Matthias Kuhn

The AES encrypted password (In the project file) + the encryption password (which will have to be given to the users, so they can tell it to QGIS which then can decrypt the password and send it to the DB for login) + an optional salt (which QGIS also has to know) will give a user everything he needs to know, in order to decrypt the original password again. With the difference to the current solution, that 98% of the users won't notice and won't worry, that it would be easy for the other 2% (under them possible attackers) to decrypt the PW.
Disclaimer: At least that's the way I understand this (as a non-crypto expert). If there's a misunderstanding from my side, please let me know.

I'd rather implement a proper user role setup in your database (with readonly for some users) with firewall (maybe VPN).

Concerning QGIS: Instead of encrypting keys and store them in the qgs file, a central key storage system may be better suited (some platforms have already built-in things e.g. gnome-keyring)
The user will then have his personal password to get access to the keystore. The first time the user connects to the database, you will have to tell him the database password which will then be encrypted on his machine (with his personal password) and stored there, so he doesn't have to memorize it or note it down.

#6 - 2013-08-12 03:12 AM - Jonathan Moules

Matthias Kuhn wrote:

> *The AES encrypted password (In the project file) + the encryption password (which will have to be given to the users, so they can tell it to QGIS which then can decrypt the password and send it to the DB for login) + an optional salt (which QGIS also has to know) will give a user everything he needs to know, in order to decrypt the original password again.*

Yes, it's basically like DRM - it's impossible to actually secure it because you need the keys to unlock it with the stuff you want to lock. But unlike DRM I think some sort of security is still better than nothing. :-)

> *I'd rather implement a proper user role setup in your database (with readonly for some users) with firewall (maybe VPN).*

We do both and probably more, but it is still like best practice to keep passwords encrypted. At least this way if/when someone posts a project online the passwords aren't open for the world to see. Not all users abide by security best practices; most probably even aren't aware of these issues.

> *Concerning QGIS: Instead of encrypting keys and store them in the qgs file, a central key storage system may be better suited (some platforms have already built-in things e.g. gnome-keyring)*
> *The user will then have his personal password to get access to the keystore. The first time the user connects to the database, you will have to tell him the database password which will then be encrypted on his machine (with his personal password) and stored there, so he doesn't have to memorize it or note it down.*

This is the sort of solution I was thinking of based on your reply. Seems like one promising direction.

#7 - 2015-01-26 09:30 AM - Jürgen Fischer
- Category changed from Data Provider/Oracle to Project Loading/Saving
- Assignee deleted (Jürgen Fischer)
- Target version changed from Future Release - High Priority to Future Release - Lower Priority

#8 - 2017-05-01 12:48 AM - Giovanni Manghi
- Easy fix? set to No

#9 - 2017-08-11 04:02 PM - Jürgen Fischer
- Duplicated by Feature request #17009: Encypted password in registry added