Sécuriser un serveur MySQL sur Windows

Traduction de l'article de Myke Miller du 1er février 2005
Disponible à cette adresse sur le site de MySQL :
http://dev.mysql.com/techresources/articles/securing_mysql_windows.html

Traduction réalisée par Xavier Chatard.
Toutes les erreurs de traduction peuvent m'être signalées par e-mail à l'adresse suivante :

- xavier<à supprimer>@thierrynardoux.com

Introduction :

Fin janvier 2005, un nouveau vers nommé Forbot était diffusé sur l'Internet visant les installations de MySQL mal sécurisées et les exploitaient pour accéder aux machines Windows les hébergeant. Forbot n'était pas un vers dans le sens où il devait être signalé pour continuer à infecter d'autres machines. Une fois que les lignes de communication entre Forbot et ses contrôleurs coupées, la diffusion du BOT a été stoppée. Davantage d'information sur Forbot peut être trouvée dans un article à http://dev.mysql.com/tech-resources/articles/security_alert.html .

Il est important de comprendre que Forbot n'exploitait aucune faiblesse ou vulnérabilité de MySQL. Il n'existe aucun correctif pour empêcher de futurs exploits. Forbot agissait en exploitant des installations de MySQL mal configurées qui ont été installées sans mot de passe administrateur ou avec un mot de passe facile à trouver. Comme exemple de mot de passe que Forbot utilisait pour accéder au compte administrateur, vous aviez **abcd123** ou **654321**.

MySQL AB assure une part importante dans le développement de nouveaux procédés pour s'assurer que l'installation par défaut de MySQL est la plus sécurisée possible et développe de nouvelles technologies qui aident les clients à rester à jour et d'être prévenu des nouvelles mises à jour et alertes techniques. Il existe cependant des choses à faire dès maintenant pour sécuriser votre serveur MySQL.

Le but de cet article est de fournir la liste des tâches qu'un administrateur doit effectuer pour sécuriser correctement une installation de MySQL sous Windows. Bien que les procédures listées soient écrites pour les utilisateurs de Windows, le contenu ci-dessous peut être aussi bénéfique aux utilisateurs Linux et Unix. Bien que Forbot visait les machines Windows, les utilisateurs de Linux et Unix pourraient être la cible de futures variantes de cette méthode d'attaque.

Etape 1 : Installer MySQL sur une version récente de Windows basée sur la technologie NT.

Les versions récentes de Windows basées sur la technologie NT incluant Windows 2000, Windows XP et Windows Server 2003 sont bien plus sures et stables que les précédentes versions de Windows comme Windows 95, Windows 98 et Windows Me.

Assurez vous que votre système d'exploitation hôte est bien à jour avec les derniers services packs et correctifs.

Etape 2 : Installer MySQL sur un système de fichier NTFS.

NTFS est un système de fichiers bien plus sur que son prédécesseur FAT32. NTFS supporte le contrôle d'accès , les fichiers importants, le cryptage de données. Pour plus d'information sur l'avantage de NTFS par rapport à FAT32 voire http://www.ntfs.com/ntfs_vs_fat.htm.

Etape 3 : Installer MySQL sur une machine autonome.

En production, MySQL devrait être installé sur un serveur dédié à l'hébergement de MySQL Server. Tous les services qui ne sont pas requis doivent être désactivés et il ne doit y avoir aucune application superflue en cours d'exécution. Ce n'est pas uniquement pour améliorer la stabilité du serveur, c'est aussi pour libérer plus de ressources pour MySQL et empêcher des applications tierces d'être des menaces potentielles de sécurité. Seul l'administrateur devrait pouvoir se connecter à la machine.

Etape 4 : Installer la dernière version de production de MySQL.

A l'heure où j'écris MySQL 4.1.9 est la dernière version de production de MySQL. Un nouvel installeur Windows a été introduit avec MySQL 4.1.5 simplifiant la procédure d'installation.de MySQL. Et il est recommandé à tous les utilisateurs de mettre à jour la dernière version 4.1 de MySQL. Bien que les problèmes de sécurités soient généralement résolus sur les versions précédentes de MySQL, utiliser la dernière version de production vous assurera une installation aussi stable et sécurisée que possible. Utiliser une version de pré-production comme MySQL 5.0 n'est pas conseillée pour les serveurs de production car tous les bugs n'ont pas nécessairement été identifiés et corrigés diminuant ainsi la stabilité et éventuellement la sécurité.

Etape 5 : Sécuriser les comptes utilisateurs de MySQL.

Durant la procédure d'installation fournissez un mot de passe administrateur quand on vous le demande. Assurez vous que le mot de passe administrateur est un mot de passe complexe contenant des lettres des chiffres et des symboles. Le mot de passe doit avoir au moins 6 caractères, ne doit pas contenir des mots trouvés dans un dictionnaire et les lettres doivent être à casse variable.

Un procédé que j'utilise consiste à trouver une phrase facile à retenir pour moi, puis prendre les premières lettres de chaque mot ainsi que la ponctuation et de les combiner pour en faire un mot de passe. Par exemple, prenons la phrase "To be, or not to be : that is the question!"; on peut convertir cette phrase en ce mot de passe "2b,On2b:Titq!". Ce mot de passe est assez puissant (ou l'était jusqu'à ce que je l'utilise dans cet article) et il est facile à retenir. Essayez d'utiliser des phrases un peu moins communes dans le cas ou de futurs vers utiliseraient cette technique pour générer une liste de mots de passes utilisés au moment d'attaquer le compte administrateur.

En plus, cocher la case marquée "**Root May Only Connect from Localhost**" et laisser la case "**Create An Anonymous Account**" décochée. Cela augmentera grandement la sécurité de votre installation MySQL.

Pour les utilisateurs voulant sécuriser une installation déjà existante, il est possible de supprimer le compte utilisateur anonyme en suivant ces lignes de commandes :

```
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 4.1.9-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use mysql;
Database changed
mysql> DELETE FROM user WHERE user = '';
Query OK, 2 rows affected (0.03 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.05 sec)

mysql>
```

En outre, le compte administrateur peut être limité aux sessions sur **localhost** avec les commandes suivantes :

```
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 4.1.9-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use mysql;
Database changed
mysql> DELETE FROM user WHERE user = 'root' AND host = '%';
Query OK, 2 rows affected (0.03 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.05 sec)

mysql>
```

Ceci laissera un seul compte administrateur ne pouvant se connecter que depuis **localhost**. Ajoutez l'entrée suivante à votre fichier **hosts** (typiquement situé *à C:\WINDOWS\system32\drivers\etc\hosts*) :

```
127.0.0.1 localhost
```

Ceci empêchera des erreurs d'ouverture où MySQL ne peut résoudre la valeur pour **localhost** dans la table *utilisateurs*.

Etape 6 Désactiver l'accès TCP/IP

Par défaut, le serveur MySQL permet les connections via TCP/IP de n'importe quel machine (mais peut rejeter une connexion basée sur l'adresse de l'utilisateur distant). Dans beaucoup de cas les connexions TCP/IP ne sont pas nécessaires et peuvent être neutralisées pour empêcher l'accès à distance au serveur MySQL. Si vous employez MySQL localement pour le développement ou pour l'utilisation d'un serveur web, vous devriez neutraliser la gestion du réseau TCP/IP.

Pour désactiver la gestion du réseau TCP/IP, choisir l'option Detailed Configuration durant l'installation et décocher l'option Enable TCP/IP Networking (vous pouvez reconfigurer votre serveur MySQL en démarrant l'assistant de configuratoin MySQL se trouvant à **démarrer > Programmes > MySQL > MySQL Server 4.1 > MySQL Server Instance Config Wizard**).

Les utilisateurs d'anciennes versions de MySQL peuvent rajouter les lignes suivantes dans la section **[mysqld]** de leur fichier de configuration du serveur MySQL

```
skip-networking

enable-named-pipes
```

Ceci désactivera les connexions TCP/IP et activera les pipes nommés. Pour que cela fonctionne il faut installer MySQL sur un système d'exploitation basé sur Windows NT, et utiliser le serveur mysqld-nt.exe. L'emplacement de votre fichier de configuration changera selon la version de MySQL que vous avez installée, recherchez un des dossiers suivants :

- C:\my.cnf
- C:\Windows\my.ini

Etape 6.1 : Utiliser les pipes nommés

Une fois le serveur démarré avec la gestion des pipes nommés, vous pouvez connecter avec la commande suivante.

```
C:\>mysql -h . -u root -p
```

Cela vous connectera au serveur en utilisant les pipes nommés. Avec les versions récentes du serveur (4.1 et superieur) vous pouvez aussi utiliser l'option **--protocol=pipe** au lieu de spécifier **–h .** sur la ligne de commande :

```
C:\>mysql --protocol=pipe -u root -p
```

Les différents outils client et APIs peuvent employer une syntaxe différente pour se connecter par l'intermédiaire des pipes nommés, consulter la documentation des outils clients ou APIs pour plus d'informations sur les connexions par l'intermédiaire de pipes nommés.

Etape 6.2

En plus des pipes nommés, MySQL 4.1 gère l'utilisation de la mémoire partagée pour les connexions à MySQL. Pour activer la mémoire partagée, ajouter la ligne suivante à la section [mysqld] de votre fichier de configuration du serveur MySQL.

```
shared-memory
```

L'assistant de configuration de MySQL ne propose pas l'option pour configurer la mémoire partagée, de ce fait, cette option doit être ajoutée dans le fichier de configuration manuellement. Les utilisateurs de l'assistant de configuration de MySQL pourront trouver le fichier **my.ini** dans **C:\Program Files\MySQL\MySQL Server 4.1\my.ini**.

Pour se connecter au serveur via la mémoire partagée, utilisez la syntaxe suivante.

```
C:\>mysql --protocol=memory -u root -p
```

Etape 7 : Lier l'adresse TCP/IP

Dans certaines situations, il n'est pas possible de désactiver la gestion du réseau TCP/IP même quand le serveur ne recevra que des requêtes à partir de **localhost**. Par exemple, quand vous utilisez des outils qui ne supporte pas les connexions par pipes nommés ou mémoire partagée. Dans de telles situations vous pouvez ajouter à la section **[mysqld]** du fichier de configuration de votre serveur la ligne suivante :

```
bind-address=127.0.0.1
```

Cela engendrera que le serveur MySQL ne répondra uniquement qu'aux requêtes provenant de localhost, et ignorera les autres requêtes provenant des autres machines.

Etape 8 : Protéger le serveur avec un pare-feu

Tous les serveurs devraient être protégés par un pare-feu en tant que première ligne de défense contre les utilisateurs malveillants. En aucunes circonstances le serveur MySQL ne doit être accessible depuis l'Internet. Quand un serveur MySQL est utilisé par des machines clientes à travers un LAN, il peut être nécessaire de permettre l'accès externe à MySQL à d'autres machines sur le réseau local, mais le LAN devrait être séparé de l'Internet par un pare-feu qui bloque le trafic sur le port 3306. Pour le moins un pare-feu logiciel devrait être installé sur le serveur MySQL permettant seulement les connexions à partir du réseau local et autres adresses IP de confiances. Dans le meilleur des cas vous devriez placer un pare-feu matériel entre le serveur MySQL et l'Internet. Ceci ne signifie pas que les utilisateurs ne peuvent pas accéder à MySQL à distance, il est possible d'employer un tunnel SSH pour gérer le trafic MySQL à travers un pare feu. Voir mon article sur les tunnels SSH à http://www.vbmysql.com/articles/security/gui-tunnel.html pour plus d'information.

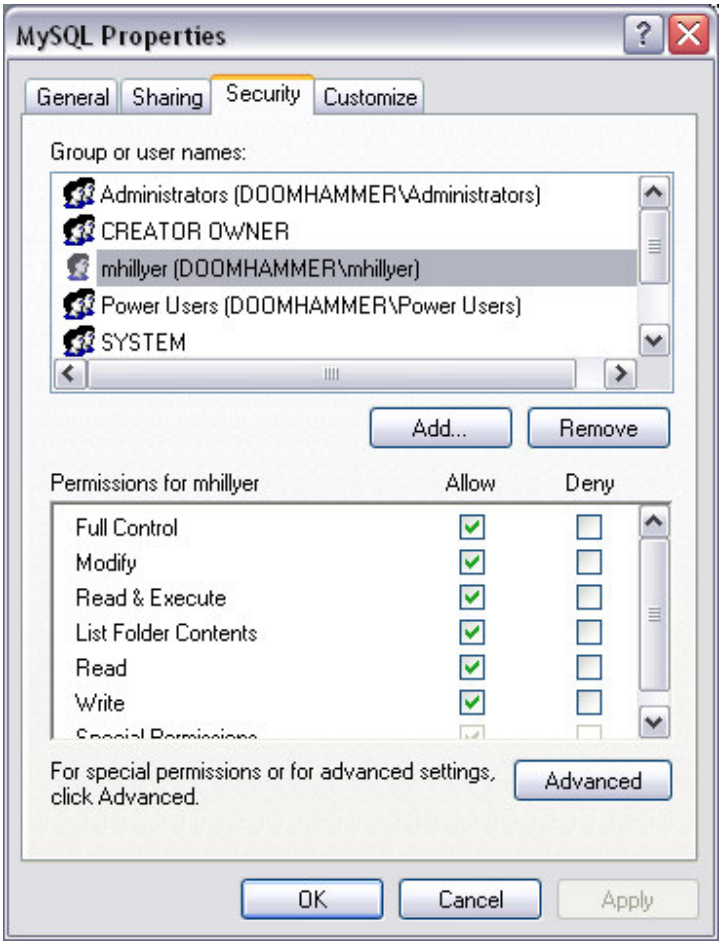
Etape 9 Lancer le service MySQL en tant qu'utilisateur limité

Par défaut, le service Mysl Serveur s'exécute en tant qu'utilisateur local privilégié du système. MySQL peut être exécuté en tant qu'utilisateur limité pour restreindre les possibilités et limiter ce dont un serveur MySQL compromis est capable.

Premièrement, créer un compte utilisateur Windows nommé *mysql* avec un mot de passe fort. Arrêtez le service en utilisant la fenêtre **Services** dans la section **outils d'administrations** du panneau de configuration de Windows. Click droit sur MySQL dans la liste des services et choisir l'option **Arrêter** du menu déroulant.

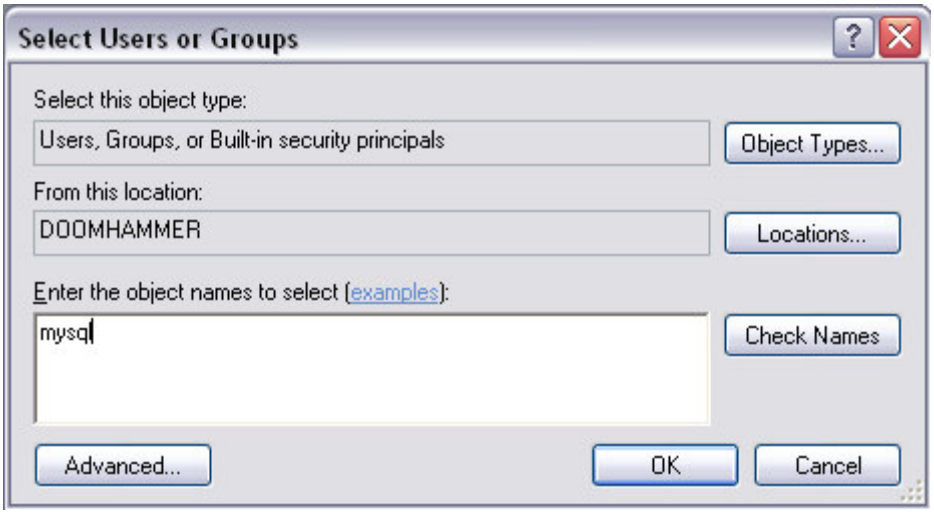
Réduisez la fenêtre des Service et explorez le répertoire MySQL, habituellement situé à **C:\Program Files\MySQL\MySQL Server 4.1**.

Changer les permissions de ce dossier et de son contenu en permettant l'accès à l'utilisateur MySQL et en bloquant tous les autres utilisateurs. Bouton droit sur le dossier et choisir l'option **Propriétés** du menu déroulant ; Sélectionner l'onglet **Sécurité** (s'il n'est pas présent, c'est que votre installation de Windows utilise *le partage de fichier simple*. Choisissez l'option **Options des dossiers …** à partir du menu **Outils** et cliquez sur l'onglet **Affichage**. Descendez à la fin des **Options avancées** et décochez **Utiliser le partage de fichiers simple**) :

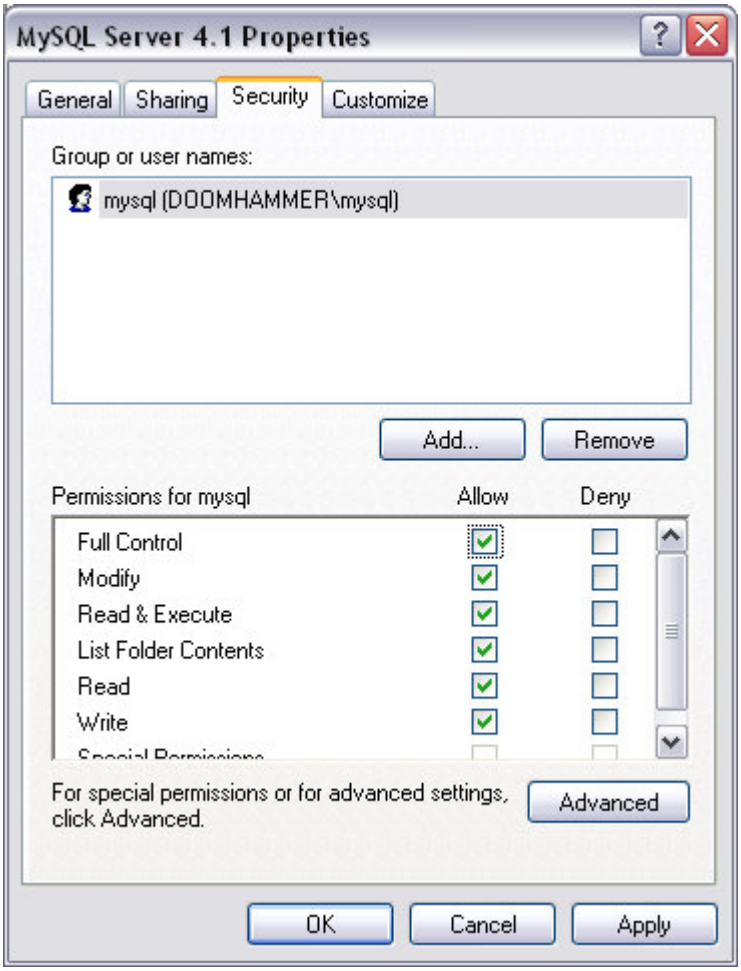


En premier lieu il y a un certain nombre de permissions fournies aux utilisateurs existants qui sont hérités à partir des répertoires parents. Celles-ci peuvent être enlevées en cliquant sur le bouton **Paramètres Avancés** et décoché **Hérite de l'objet parent**.... Une fois terminé, cliquez le bouton **Supprimer** pour enlever les permissions existantes.

Cliquez sur le bouton **Ajouter...** sous la liste utilisateurs pour ajouter l'utilisateur **mysql** à la liste des utilisateurs avec la permission d'accéder au répertoire d'installation de MySQL (qui devrait maintenant être vide puisque nous avons enlevé toutes les permissions existantes) :

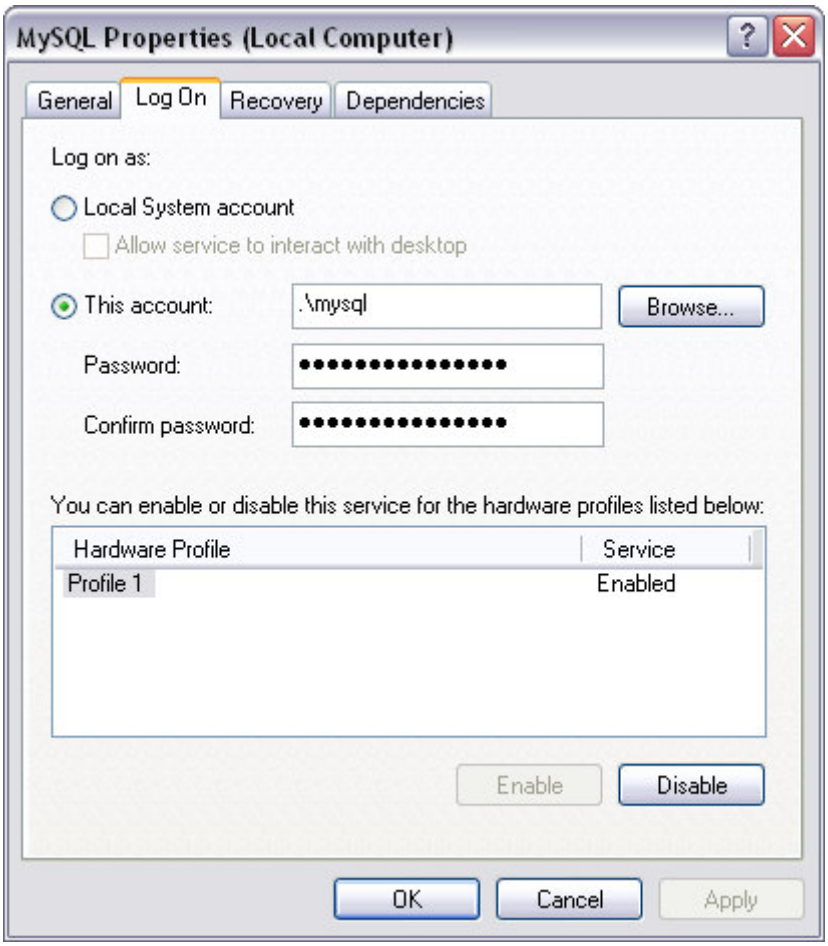


Entrer le nom utilisateur mysql et cliquez sur le bouton **Vérifier les noms**. Le chemin et le nom de l'utilisateur appropriés devraient être complétés et vous pouvez cliquer sur le bouton **OK**. Vérifiez l'option **Contrôle total** dans la liste des permissions pour donner un accès complet à ce répertoire.



Cliquez sur **Appliquer** pour accorder les permissions. En plus, accordez les permissions à votre propre compte utilisateur, ainsi vous pourrez modifiez plus tard le fichier de configuration et utiliser les outils tel que **myisamchk** sans changer d'utilisateur.

Une fois les permissions sur le répertoire configurées, retournez sur la liste des services ; click droit sur le service **MySQL** et sélectionnez l'onglet **Connexion** :



Choisir le bouton radio **Ce compte** et remplissez les informations sur le compte **mysql** créé précédemment. Cliquez sur OK pour enregistrer les changements, puis clic droit sur le service MySQL dans la liste des services et choisir l'option **Démarrer**. Votre service MySQL devrait démarrer, exécuté en tant que l'utilisateur limité *mysql*.

Etape 10 : Crypter le répertoire de données.

Pour les utilisateurs qui stockent des informations particulièrement sensibles avec MySQL, il est possible de crypter le répertoire de données de votre installation MySQL. Le cryptage doit être effectué quand le serveur n'est pas en cours d'exécution, et quand vous êtes connecté en tant que l'utilisateur mysql. Les utilisateurs doivent être prévenus que si la clef privée utilisée pour crypter le répertoire de données est perdue, toutes les données du répertoire sont perdues. Les performances peuvent être diminuées du fait que tous les fichiers doivent être décryptés avant de pouvoir y accéder. Considérant le risque existant sur les données et la perte de performance, le cryptage du répertoire de données n'est pas recommandé, uniquement si cela est considéré comme absolument nécessaire, et cela ne devrait être utilisé que par des utilisateurs expérimentés.

Les informations sur le cryptage des données, avec une liste des meilleures pratiques, peut être trouvée à http://support.microsoft.com/default.aspx?scid=kb;EN-US;q223316 . Il est important de noter que les fichiers sont cryptés avec l'utilisateur windows mysql, et les applications externes comme **myisamchk** n'auront pas accès au répertoire de données sans être connectées sous l'utilisateur mysql.

Etape 11 : Donner le minimum de privilèges nécessaires.

Lors de la création de nouveaux utilisateurs et de l'attribution des droits, il est souvent facile de leur attribuer tous les privilèges sur une base de données ou tous les droits globalement mais ceci devrait être évité. Lors de l'attribution des privilèges, essayez d'attribuer le minimum nécessaire à un utilisateur pour qu'il exécute les taches qui lui sont assignées. Attribuez les privilèges par base de base de données, en évitant d'employer un nom d'hôte %. Si un utilisateur doit se connecter d'un réseau 192.168.1.0, accordez les privilèges **'username'@'192.168.1.% '**. Essayez d'être le plus restrictif possible, et n'accordez des privilèges additionnels seulement quand cela est nécessaire.

Par exemple, lors de la création d'un nouvel utilisateur pour la base de données 'fictional' devant effectuer des requêtes et manipuler des données, agissez comme suit :

```
mysql> GRANT SELECT, INSERT, UPDATE, DELETE ON fictional.* TO
'bob'@'192.168.1.%';
```

Etape 12 : Changer le nom du compte administrateur

Le compte administrateur n'à pas besoin de s'appeler '**root**'. Beaucoup d'attaques essaierons de compromettre le compte utilisateur '**root** ' et seront stoppées s'il il n'y a pas d'utilisateur '**root**'. Pour changer le nom du compte administrateur, utilisez les commandes suivantes :

```
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 4.1.9-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> USE mysql;
Database changed
mysql> UPDATE user SET user='bob' WHERE user='root';
Query OK, 1 row affected (0.19 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.23 sec)
```

Vous pouvez naturellement employer n'importe quel nom, mais je recommande de ne pas employer votre propre nom car un attaquant pourrait supposer qu'un compte avec votre nom aurait des privilèges du même niveau que l'administrateur.

Conclusion

Avec de simples étapes, MySQL sur Windows peut être sécurisé et protégé des utilisateurs mal intentionnés tentant d'accéder à MySQL et aux données qu'il contient. Les étapes clefs sont de sécuriser le compte utilisateur par défaut, de limiter les accès extérieurs, et d'utiliser un mot de passe complexe. Ceux qui cherchent à augmenter la sécurité de leur serveur peuvent exécuter MySQL en tant qu'utilisateur limité, changer le nom de compte administrateur, et même crypter les données du répertoire de données.

Il est tout à fait probable qu'il y aura de futures variantes du malware Forbot, mais en se préparant et en adhérant à de simples bonnes règles de sécurité, il est possible de prévenir ces attaques sans jamais atteindre ou compromettre votre serveur MySQL