دانشگاه صنعتی شریف

ITRC

آزمایشگاه و مرکز تخصصی آپا

در حوزه پایگاه داده‌ها

چگونه می‌توان MySQL را امن ساخت

ساره سادات امامی

emami@ee.kntu.ac.ir

دی‌ماه ۱۳۸۷

مقاله سفید - گزارش فنی

مقدمه

MySQL یکی از محبوب‌ترین پایگاه داده‌ها به خصوص در محیط‌های اینترنتی است. از مزایای MySQL سادگی استفاده از آن و کارایی بالای آن است. در راستای استفاده مؤثرتر از این پایگاه داده‌ها باید آن را تا حد ممکن امن ساخت. امن سازی یک سیستم مدیریت پایگاه داده‌ها از جنبه‌های مختلفی قابل توجه است، برای مثال نصب آن بر روی کارگزار[۱] باید به گونه‌ای امن باشد که آن را از دید دیگران محفوظ نگاه دارد، به همین ترتیب برای محافظت پایگاه داده‌ها در شبکه باید تدابیر مختلفی در نظر گرفته شود. در این مقاله سعی شده است، روش‌های امن سازی پایگاه داده‌های MySQL به طور کامل شرح داده شود.

مراحل امن‌سازی MySQL

در این مقاله امن سازی کارگزار MySQL و برقراری امنیت در ارتباطات آن در ۹ مرحله توضیح داده می-شود، که هر مرحله خود شامل چندین بخش است. این مراحل از قرار زیر هستند:

۱. نصب امن کارگزار MySQL
   أ. ایجاد کاربر و گروه کاربری اختصاصی در سیستم عامل
   ب. اعطای حقوق مورد نیاز به کاربر ایجاد شده
   ج. اعمال chrooting در سیستم عامل FreeBSD
۲. پیکربندی کارگزار MySQL به شکلی امن
   أ. غیرفعال‌سازی دسترسی‌های ناامن از راه دور به کارگزار MySQL
   ب. فعال‌سازی دسترسی‌های امن از راه دور به کارگزار MySQL با استفاده از پروتکل‌های SSL و یا SSH
   ج. بهینه‌سازی تنظیمات پیکربندی کارگزار MySQL با افزودن تنظیماتی خاص به فایل پیکربندی

[۱] Server

۳. حذف اشیاء غیرضروری در پایگاه داده‌های MySQL

أ. حذف پایگاه داده‌های test

ب. حذف حساب‌های کاربری بی‌نام

۴. ایمن‌سازی حساب‌های کاربری

أ. اطمینان حاصل کردن از وجود گذرواژه برای کلیه حساب‌های کاربری، با بررسی اطلاعات کلیه حساب‌های کاربری و انتساب گذرواژه به حساب‌هایی که گذرواژه ندارند

ب. تا جای ممکن عدم استفاده از کاراکترهای عام در نام میزبان حساب‌های کاربری

ج. ایمن‌سازی حساب کاربری مدیر با تغییر نام آن از root به نامی دیگر، اعطای مجوزی قوی به آن، غیرفعال‌سازی دسترسی از راه دور به آن و یا مشخص کردن نام میزبان راه دور به طور کامل

۵. استفاده از احراز هویت ایمن برای تصدیق هویت کاربران

أ. انتخاب گذرواژه‌های قوی و ایمن برای کاربران

ب. رمزگذاری گذرواژگان با استفاده از تابع PASSWORD() در MySQL از نسخه‌های بعد از 4.1

ج. ارائه ایمن گذرواژه به کارگزار با قرار دادن آن در فایلی که از دید دیگر کاربران به دور است و یا عدم ارائه آن به شکلی که در دید دیگران باشد

۶. کنترل دسترسی کاربران

أ. اعطای تنها حقوق مورد نیاز به هر کاربر و ابطال مجوزهای غیر ضروری آنها

ب. عدم اعطای حقوق مدیریتی به کاربران معمولی و غیر مدیر

ج. محدود کردن تعداد اتصالات همزمان هر کاربر به کارگزار MySQL

د. استفاده از دیدها برای محدود کردن دسترسی‌های کاربران

۷. ذخیره‌سازی امن اطلاعات

أ. رمز کردن اطلاعات ستون‌های حساس جداول با استفاده از توابع رمزگذاری مانند AES

۸. پشتیبان‌گیری منظم از اطلاعات
   أ. پشتیبان‌گیری در زمان‌های مشخص با استفاده از برنامه‌های جانبی mysqldump و mysqlhotcopy (در شرایط خاص می‌توان در سطح دستورات SQL نیز از اطلاعات پشتیبان گرفت)
   ب. فعال‌سازی ثبت دودویی وقایع برای استفاده از آن در به‌روز رسانی اطلاعات بعد از بازیابی پشتیبان‌ها
۹. فعال‌سازی گزارش‌های ثبت وقایع در کارگزار برای انجام امور نظارتی در سیستم
   أ. فعال‌سازی گزارش ثبت خطا برای پیگیری خطاهای رخ داده در سیستم
   ب. فعال‌سازی گزارش ثبت پرس و جوی جامع برای نظارت بر اتصالات و پرس و جوهای هر کاربر
   ج. فعال‌سازی گزارش ثبت دودویی وقایع به منظور پیگیری تغییرات درخواستی هر کاربر و همچنین استفاده از آن در به‌روز رسانی بازیابی پایگاه‌های داده‌ها بعد از بازیابی فایل پشتیبان
   د. فعال‌سازی ثبت پرس و جوی کند برای شناسایی پرس و جوهای زمان‌بر و تشخیص مکان‌هایی که برای افزایش سرعت نیاز به شاخص دارند

۱. نصب امن کارگزار MySQL

اولین قدم برای داشتن یک پایگاه داده‌های امن، نصب ایمن کارگزار آن است. قبل از نصب MySQL بهتر است یک کاربر و گروه کاربری بر روی سیستم عامل ایجاد شود که مختص MySQL باشد. برای مثال اگر از سیستم عامل لینوکس استفاده می‌کنید، با اجرای دستورات زیر در پوسته[۲] لینوکس می‌توانید گروه و کاربر اختصاصی MySQL را بسازید.

[۲] Shell

```
addgroup MySQL
adduser --group MySQL mysql
```

در سیستم عامل ویندوز باید به مسیر Control Panel -> Administrative Tools -> Computer Management -> Local Users and Groups رفته، سپس گروه کاربری و کاربر جدید از این گروه را ایجاد کنید. حال با کاربر جدید وارد سیستم عامل شده و MySQL را نصب کنید. نصب MySQL در سیستم عامل لینوکس می‌تواند با استفاده از فایل‌های rpm و یا کامپایل کد منبع[۳] آن انجام شود. در سیستم عامل ویندوز نیز می‌توان علاوه بر کامپایل و ایجاد فایل اجرایی، از فایل‌های نصب خودکار مخصوص ویندوز (windows installer) استفاده کرد. بعد از نصب MySQL باید مالکیت مسیری که پایگاه داده‌های MySQL در آنجا قرار دارد، به علاوه اجازه خواندن این فایل‌ها را فقط به گروه و کاربر MySQL داد. برای مثال در سیستم عامل لینوکس باید دستورات زیر را اجرا کرد:

```
chown -R root /usr/local/mysql
chown -R mysql /usr/local/mysql/var
chgrp -R MySQL /usr/local/mysql
```

همچنین باید مجوزهای زیر را بر روی فایل پیکربندی MySQL (در لینوکس my.cnf و در ویندوز my.ini) تعریف کرد:

```
chown root:sys /etc/my.cnf
chmod 644 /etc/my.cnf
```

به این ترتیب مالک فایل پیکربندی MySQL را کاربران root و sys تعریف کرده و اجازه خواندن و نوشتن بر روی فایل را به این مالکان می‌دهیم، همچنین به اعضای گروه و دیگر کاربران فقط اجازه خواندن فایل پیکربندی را می‌دهیم. در صورتی که از سیستم عامل ویندوز استفاده می‌کنید برای اعطای هریک از این مجوزها باید بر روی فایل و یا پوشه مورد نظر راست-کلیک کرده و در بخش security مجوزهای مورد نظر را تعریف کنید.

یکی دیگر از تدابیر امنیتی پیشنهادی در زمان نصب و راه اندازی اولیه کارگزار MySQL، که تنها قابل اجرا در سیستم عامل لینوکس FreeBSD است، chrooting است. chrooting به عملیات تغییر محل فایل‌های

[۳] Source code

اجرایی سرویس پایگاه داده‌های MySQL (یا همان daemonها) از مسیر پیش فرض و انتقال آنها به مسیر دیگر، گفته می‌شود. با استفاده از chrooting برنامه MySQL فقط تحت مجوزهای تعریف شده اجرا شده و دسترسی به آن به طور کامل تحت کنترل قرار می‌گیرد. برای این کار توصیه می‌شود ابتدا یک بار کارگزار را راه اندازی کرده (با اجرای دستور start service mysql) و بعد از اطمینان از اتصال به کارگزار MySQL، کارگزار را غیر فعال سازید (با اجرای دستور stop service mysql). برای اعمال chrooting ابتدا مسیر جدیدی مانند /chroot/mysql ساخته و کلیه پوشه‌های لازم را در آن ایجاد می‌کنیم:

```
mkdir -p /chroot/mysql/dev
mkdir -p /chroot/mysql/etc
mkdir -p /chroot/mysql/tmp
mkdir -p /chroot/mysql/var/tmp
mkdir -p /chroot/mysql/usr/local/mysql/libexec
mkdir -p /chroot/mysql/usr/local/mysql/share/mysql/English
```

دسترسی به پوشه‌های ساخته شده باید محدود و به صورت زیر باشد:

```
chown -R root:sys /chroot/mysql
chmod -R 755 /chroot/mysql
chmod 1777 /chroot/mysql/tmp
```

سپس فایل‌های زیر باید در مسیرهای جدید کپی شوند:

```
cp /usr/local/mysql/libexec/mysqld /chroot/mysql/usr/local/mysql/libexec/
cp /usr/local/mysql/share/mysql/english/errmsg.sys
/chroot/mysql/usr/local/mysql/share/mysql/english/
cp /etc/hosts /chroot/mysql/etc/
cp /etc/host.conf /chroot/mysql/etc/
cp /etc/resolv.conf /chroot/mysql/etc/
cp /etc/group /chroot/mysql/etc/
cp /etc/master.passwd /chroot/mysql/etc/passwords
cp /etc/my.cnf /chroot/mysql/etc/
```

حال باید گذرواژه[۴] کلیه کاربران و گروه‌های غیر از mysql از فایل‌های /chroot/mysql/etc/passwords و /chroot/mysql/etc/group حذف شود. سپس پایگاه داده‌های گذرواژه باید به شکل زیر ساخته شود:

```
cd /chroot/mysql/etc
pwd_mkdb -d /chroot/mysql/etc passwords
rm -rf /chroot/mysql/etc/master.passwd
```

[۴] Password

یکی دیگر از فایل‌هایی که باید در مسیر جدید ساخته شود، وسیله تهی به نام null است که در مسیر /chroot/mysql/dev/ قرار دارد:

```
mknod /chroot/mysql/dev/null c 2 2
chown root:sys /chroot/mysql/dev/null
chmod 666 /chroot/mysql/dev/null
```

در نهایت باید فایل‌های پایگاه‌های داده‌های MySQL و جداول مجوزها[۵] در مسیر جدید کپی شوند:

```
cp -R /usr/local/mysql/var/ /chroot/mysql/usr/local/mysql/var
chown -R mysql:mysql /chroot/mysql/usr/local/mysql/var
```

در شرایطی که از زبان دیگری به غیر از انگلیسی برای نگهداری داده‌ها استفاده می‌شود باید فایل‌های مربوطه از مسیر /usr/local/mysql/share/mysql/charsets در مسیر جدید معادل آن کپی شوند.

بعد از انتقال اطلاعات لازم به مسیر مورد نظر، باید اجرای صحیح MySQL را در مسیر جدید آزمایش کرد:

```
chrootuid /chroot/mysql mysql /usr/local/mysql/libexec/mysqld &
```

۲. پیکربندی کارگزار MySQL

در این مرحله باید کارگزار MySQL را با توجه به نیازهای امنیتی خود پیکربندی کرد. فایل اصلی پیکربندی MySQL در سیستم عامل لینوکس به طور پیش فرض در مسیر /etc/my.cnf، و در سیستم عامل ویندوز در مسیر نصب آن قرار دارد (برای مثال C:\Program Files\MySQL\MySQL Server 5.0\my.ini).

تغییرات مورد نیاز برای پیکربندی ایمن MySQL از قرار زیر است:

غیر فعال سازی دسترسی‌های ناامن از راه دور به کارگزار

کارگزار MySQL به طور پیش فرض قابل دسترسی از راه دور بوده و به درگاه ۳۳۰۶ TCP/IP گوش می-دهد، چنانچه نیازی به دسترسی از راه دور به پایگاه داده‌ها نیست، برای مثال به طور مستقیم از پایگاه داده-ها استفاده نمی‌شود و از نرم افزار واسطی مانند PHP استفاده می‌شود، بهتر است دسترسی از راه دور به کارگزار MySQL غیر فعال شود. برای این کار باید تنظیم زیر را به محتویات فایل پیکربندی MySQL اضافه کرد:

[۵] Grant tables

```
skip-networking
```

به هر حال توصیه می‌شود اجازه دسترسی از راه دور به فراکاربران مدیر (برای مثال root) داده نشود، زیرا منشأ بسیاری از آسیب‌پذیری‌ها دسترسی راه دور به کاربر root است.

فعال‌سازی دسترسی‌های امن از راه دور به کارگزار

توصیه می‌شود در صورت نیاز به اتصال از راه دور به پایگاه داده‌های MySQL است، برقراری ارتباط و تبادل اطلاعات در بستری امن و رمزشده صورت پذیرد. برای برقراری ارتباط امن و رمز شده می‌توان از پروتکل‌های ارتباطی SSL[۶] و SSH[۷] بهره برد. کارگزار MySQL در نسخه‌های بعد از 4.0.0، از SSL در ارتباطات پشتیبانی می‌کند، به علاوه درگاه SSH نیز می‌تواند در هنگام ارسال داده‌ها به عنوان مسیری برای رمزگذاری و فشرده سازی مورد استفاده قرار گیرد. پروتکل SSL مستقیم با کارگزار MySQL در ارتباط بوده و رمزگذاری و رمزگشایی اطلاعات در سمت کارگزار از طریق خود MySQL انجام می‌شود. اما در SSH این ارتباط از طریق سیستم عامل کنترل و رمز می‌شود، و اطلاعات در سمت کارگزار بعد از رمزگشایی در اختیار MySQL قرار می‌گیرند، به عبارتی رمزگشایی و رمزگذاری اطلاعات برعهده کارگزار SSH است.

پروتکل SSL اساساً برای امن سازی ارتباط در لایه انتقال اطلاعات[۸] شبکه استفاده می‌شود که عموماً بر رمزگذاری و انتقال اطلاعات با استفاده از سیستم رمزنگاری کلید عمومی[۹] تکیه دارد. در این پروتکل هدف حفظ حریم خصوصی، صحت و درستی اطلاعات است. در مقاله سفید "برقراری ارتباط امن از طریق SSL در MySQL" چگونگی اتصال به کارگزار MySQL با استفاده از پروتکل SSL توضیح داده شده است.

پروتکل SSH نیز برای برقراری ارتباط امن و رمز شده ایجاد شده است. این پروتکل خود حاوی برخی استانداردهای احراز هویت بوده و برای رمزنگاری داده‌ها از روش‌های مختلف از جمله رمزنگاری کلید عمومی استفاده می‌کند. هدف اصلی این پروتکل برقراری ارتباط به شکلی امن است، با این وجود بستری امن برای

---

۶ Secure Socket Layer
۷ Secure SHell
۸ Transport layer
۹ Public key cryptography

انتقال اطلاعات را فراهم می‌آورد که به طور پیش فرض از رمزنگاری متقارن استفاده می‌کند. در سیستم عامل لینوکس این پروتکل به طور خودکار پیاده سازی شده است، اگر کارخواه[۱۰] از سیستم عامل لینوکس استفاده کند، با استفاده از پیاده سازی SSH در سیستم عامل خود می‌تواند به کارگزار MySQL متصل گردد. اما چنانچه کارخواه از سیستم عامل ویندوز استفاده کند، می‌تواند از نرم افزارهایی که این پروتکل را پیاده‌سازی کرده‌اند، مانند PUTTY، استفاده کند. نحوه برقراری ارتباط از طریق PUTTY در مقاله سفید "برقراری ارتباط امن از طریق Putty بین کارگزار و کارخواه MySQL در Windows XP" توضیح داده شده است. در نهایت چنانچه کارگزار بر روی سیستم عامل ویندوز نصب شده باشد، می‌توان از نرم افزار OpenSSH برای برقراری ارتباط امن استفاده کرد که به طور مفصل در مقاله سفید "برقراری ارتباط امن در MySQL با استفاده از SSH در ویندوز" توضیح داده شده است.

بهینه‌سازی تنظیمات پیکربندی کارگزار

برای بهبود ایمنی عملکرد پایگاه داده‌های MySQL، بهتر است تغییرات زیر را در تنظیمات فایل پیکربندی کارگزار MySQL انجام داد:

- برای بهبود امنیت محلی، مناسب است استفاده از دستور LOAD DATA LOCAL INFILE را غیرفعال ساخت. بدین وسیله از خواندن غیرمجاز فایل‌های محلی با استفاده از این دستور جلوگیری می‌شود. از این دستور در تزریق دستورات SQL[۱۱] از طریق سایت‌های وب تحت PHP بسیار استفاده می‌شود. برای غیرفعال‌سازی استفاده از این دستور باید پارامتر زیر را به فایل پیکربندی MySQL اضافه کرد.

```
local-infile=0
```

- اگر بخواهیم مطمئن باشیم که دستور SHOW DATABASES فقط پایگاه داده‌هایی را نشان می‌دهد که کاربر حق دسترسی به آنها را دارد، باید دستور زیر را به تنظیمات پیکربندی اضافه کرد:

[۱۰] Client
[۱۱] SQL injection

```
safe-show-database
```

- برای اطمینان از ایجاد کاربر جدید با استفاده از دستور GRANT تنها توسط کاربرانی که دارای مجوز INSERT بر روی جدول mysql.user هستند، باید دستور زیر به تنظیمات پیکربندی افزوده شود:

```
safe-user-create
```

- افزودن دستور زیر به فایل پیکربندی برای جلوگیری از احراز هویت کاربرانی که گذرواژه آنها با نسخه‌های قبل از MySQL 4.1 ایجاد شده است، پیشنهاد می‌شود (گذرواژه در این نسخه‌ها با الگوریتم‌های رمزنگاری ضعیفی رمز می‌شود که به راحتی قابل شکستن هستند):

```
secure-auth
```

- هرگز از دستور skip-grant-tables به غیر از شرایط ضروروی و خاص استفاده نکنید، زیرا بدون در نظر گرفتن حقوق کاربران به آنها اجازه دسترسی می‌دهد. استفاده از این انتخاب به غیر از شرایط بحرانی و برای رفع مشکلات بسیار خطرناک است.
- اجازه استفاده از لینک‌های نمادین[۱۲] یا همان symlink برای ارتباط با جداول را لغو کنید. زیرا مخصوصاً در زمانی که MySQL با کاربر root فراخوانی می‌شود، هر کاربری که حق نوشتن بر روی مسیرهای داده کارگزار را دارد، می‌تواند فایل‌های پایگاه داده‌ها را از روی سیستم حذف کند. برای این منظور باید دستور زیر را به فایل پیکربندی افزود:

```
skip-symbolic-links
```

[۱۲] Symbolic links

۳. حذف اشیاء غیرضروری

در این مرحله باید اشیاء غیرضروری موجود در پایگاه داده‌های MySQL را حذف کنیم، زیرا ممکن است خود این اشیاء باعث بروز آسیب پذیری در MySQL گردند. از جمله اشیاء غیرضروری در MySQL پایگاه داده‌های اضافی test و حساب‌های کاربری بی‌نام[۱۳] هستند.

برای حذف پایگاه داده‌های test باید با کاربر مدیر که حق Drop بر روی این پایگاه داده‌ها را دارد به کارگزار MySQL متصل شده و دستور زیر را اجرا کرد:

```
mysql> DROP DATABASE test;
```

برخی از نسخه‌های نصب MySQL، حساب‌های کاربری بی‌نامی ایجاد می‌کنند، که پایگاه داده‌ها را بسیار آسیب پذیر می‌سازد. برای از بین بردن این حساب‌های کاربری باید با کاربر مدیر به کارگزار متصل شده و دستورات زیر را اجرا کرد.

```
mysql> USE mysql;
mysql> DELETE FROM user WHERE User='';
mysql> DELETE FROM db WHERE User='';
mysql> FLUSH PRIVILEGES;
```

با اجرای دستور FLUSH PRIVILEGES بعد از حذف کاربران بی‌نام، به کارگزار اطلاع داده می‌شود که جداول حقوق خود را دوباره بازیابی کند و احراز هویت و مجازشماری کابران را بر اساس اطلاعات جدید موجود در جداول مجوزها انجام دهد.

۴. ایمن‌سازی حساب‌های کاربری

در هنگام نصب کارگزار MySQL، پایگاه داده‌های مدیریتی به نام mysql ساخته می‌شود. جداولی که اطلاعات مدیریتی را در خود نگهداری می‌کنند، در این پایگاه داده‌ها قرار دارند. جدول user یکی از جداول مدیریتی است که اطلاعات کاربران معتبر و برخی حقوق کلی کاربر را در خود دارد. برای ایمن‌سازی حساب‌های کاربری MySQL باید به نکات زیر توجه داشت:

[۱۳] Anonymous accounts

اطمینان از وجود گذرواژه برای حساب‌های کاربری

برای حفظ امنیت MySQL، کلیه حساب‌های کاربری باید دارای گذرواژه باشند. به این ترتیب، حساب‌های کاربری که دارای گذرواژه نیستند، باید حذف شوند و یا گذرواژه به آنها اختصاص یابد. برای یافتن چنین حساب‌ها باید دستور زیر را بر روی جدول user از پایگاه داده mysql اجرا کرد.

```
mysql> SELECT * FROM mysql.user WHERE Password = '';
```

با اجرای این پرس و جو[۱۴] کلیه سطرهایی که غیر امن هستند و باید به آنها گذرواژه نسبت داده شود، به دست می‌آیند. از آنجاییکه هر حساب کاربری با نام کاربری و نام میزبان آن مشخص می‌شود، برای انتساب گذرواژه به یک حساب کاربری باید نام کاربری و نام میزبان نسبت داده شده به آن را مشخص کرد. فرض کنید مقادیر ستون‌های User و Host منسوب به یک حساب کاربری بدون گذرواژه به ترتیب user-name و host-name هستند. دستور انتساب گذرواژه new-pass به این حساب کاربری از قرار زیر است:

```
mysql> UPDATE user SET Password = PASSWORD('new-pass')
-> WHERE User='user-name' AND Host='host-name';
```

همچنین می‌توان از دستور SET PASSWORD نیز برای این منظور به شکل زیر استفاده کرد:

```
mysql> SET PASSWORD FOR 'user-name'@'host-name'=PASSWORD('new-pass')
```

اگر برای نسبت دادن گذرواژه از دستور UPDATE استفاده می‌کنید، توصیه می‌شود بعد از آن دستور FLUSH PRIVILEGES را اجرا کنید، تا تغییرات اعمال شوند.

عدم استفاده از کاراکترهای عام در شرایط غیر ضروری

اگر مقدار Host در جدول user صرفاً نام خاصی را بیان کند، کارخواه فقط از آن میزبان خاص اجازه اتصال به کارگزار را دارد. اما اگر این مقدار شامل کاراکترهای عام[۱۵] باشد، کاربر قادر به اتصال به کارگزار از میزبان-های مختلفی خواهد بود. برای مثال اگر مقدار میزبان عبارتی مانند "%.xyz.com" باشد، کاربر می‌تواند از هر

[۱۴] Query
[۱۵] Wildcards

میزبانی در دامنه[۱۶] xyz.com به کارگزار متصل شود. مقدار میزبان حتی می‌تواند برابر با عبارت "%" باشد، که در این صورت امکان اتصال کاربر به کارگزار از هر میزبانی ممکن است. حساب‌های کاربری که در نام میزبان آنها از کاراکتر عام استفاده شده است نسبت به آنهایی که مشخصاً نام میزبان آنها بیان شده است، بیشتر در معرض حمله قرار دارند. در شرایطی که نام میزبان در حساب کاربری به طور خاص بیان شده باشد، مانند localhost، مهاجم به غیر از حدس زدن گذرواژه، فقط باید از میزبانی که کارگزار در آن نصب شده است برای اتصال به آن استفاده کند، که بسیار ایمن‌تر از زمانی است که مهاجم بتواند حمله را از هر میزبانی انجام دهد.

برای یافتن حساب‌های کاربری که از کاراکتر عام '%' در نام میزبان آنها استفاده شده است، می‌توان از پرس و جوی زیر استفاده کرد.

```
mysql> SELECT * FROM mysql.user WHERE Host LIKE '%\%%';
```

بعد از یافتن این حساب‌های کاربری، برای تغییر نام میزبان آنها به نامی محدودتر می‌توان از دستور UPDATE استفاده کرد. بعد از انجام تغییرات برای اعمال آنها بهتر است دستور FLUSH PRIVILEGES اجرا شود.

زمانی که حساب کاربری جدیدی می‌سازید (برای مثال با استفاده از دستور GRANT)، توجه داشته باشید که تا حد مقدور در نام میزبان کاربر از کاراکتر عام استفاده نشده باشد.

ایمن‌سازی حساب کاربری مدیر

در هنگام نصب MySQL کاربر مدیر آن به طور پیش‌فرض با نام کاربری root ایجاد می‌شود. root یک فراکاربر بوده و همه مجوزهای سیستم را داراست. اولین اقدام برای ایمن سازی کاربر مدیر، تغییر نام آن از root به یک نام کاربری ناآشنا است، زیرا در حملات لغت نامه‌ای جامع[۱۷] دانستن نام کاربری مدیر به مهاجم کمک بزرگی می‌کند. برای تغییر نام کاربری مدیر می‌توان دستور زیر را اجرا کرد:

---

[۱۶] Domain

[۱۷] Brute-force attacks

```
mysql> USE mysql;
mysql> UPDATE user SET User='new_username' WHERE User='root';
mysql> FLUSH PRIVILEGES;
```

توجه داشته باشید که کاربران مدیر حتماً باید دارای گذرواژه باشند و گذرواژه آنها اگر در فایلی نگهداری می‌شود آن فایل باید از دید دیگر کاربران سیستم عامل مخفی باشد. همچنین تا حد ممکن دسترسی از راه دور به کاربران مدیر را باید غیرفعال ساخت، و در شرایطی که نیاز به دسترسی از راه دور به کاربر مدیر است، سعی شود که میزبان آن به طور خاص مشخص شود.

۵. احراز هویت ایمن کاربران

احراز هویت در MySQL با استفاده از نام کاربری و گذرواژه انجام می‌شود. به این ترتیب که برای هر حساب کاربری یک نام کاربری و یک گذرواژه در نظر گرفته می‌شود. البته در ایجاد حساب کاربری، میزبانی که کاربر از آن به کارگزار متصل می‌شود نیز از اهمیت برخوردار است. برای مثال در دو حساب کاربری متفاوت می‌توان دو نام کاربری یکسان را در شرایطی که میزبان‌های متفاوتی دارند، تعریف کرد.

انتخاب گذرواژه مناسب

برای انتخاب گذرواژه بهتر است از ترکیب حروف و ارقام استفاده گردد و گذرواژه تا حد ممکن طولانی باشد. در حال حاضر رمزگشاهای متنوعی برای رمزگشایی گذرواژه‌ها وجود دارند. در نتیجه باید در ایجاد گذرواژه دقت کرد. سعی کنید از کلمات معنی‌دار در گذرواژه استفاده نکنید، زیرا این گذرواژه‌ها در حملات لغت‌نامه-ای جامع به راحتی به دست می‌آیند. یکی از راه‌های ایجاد گذرواژه که به یادآوردنی و در عین حال پیچیده باشد این است که یک جمله‌ای مانند "Mary had a little lamb" در نظر گرفته شود و اولین حروف از کلمات این جمله انتخاب گردد (برای مثال "mhall"). به دنبال این کلمات می‌توان اعداد به یادآوردنی نیز اضافه کرد، در این صورت یک گذرواژه پیچیده و به یادآوردنی ساخته می‌شود.

رمزگذاری گذرواژه

نگهداری گذرواژه‌ها در پایگاه داده‌ها نباید به صورت متن ساده و رمز نشده باشد. زیرا هنگامی که سیستم در خطر شنود قرار گیرد، مهاجم به راحتی به لیست تمامی گذرواژه‌ها دسترسی پیدا کرده و می‌تواند از آنها سوء استفاده کند. بنابراین باید از توابع رمزگذاری درهم‌ساز[۱۸] (یک‌طرفه) مانند md5 و SHA1 استفاده کرد. MySQL خود به طور پیش فرض گذرواژه را با تابع درهم‌ساز خود رمز می‌کند. در نسخه‌های قدیم MySQL (نسخه‌های قبل از MySQL 4.1) برای رمزگذاری گذرواژه از توابع درهم‌ساز قدیمی استفاده می‌شد، به این ترتیب مقدار درهم‌شده گذرواژه، ۱۶ بایت بوده و اکنون به راحتی قابل شکستن است. در نسخه‌های بعد از MySQL 4.1 از توابعی که مقدار درهم‌شده ۴۱ بایتی از گذرواژه ایجاد می‌کنند استفاده شده است، که بسیار امن‌تر از مقدار درهم‌شده نسخه‌های قدیمی است.

تابع رمزگذاری گذرواژه در MySQL با استفاده از تابع PASSWORD() قابل فراخوانی است، این تابع با تابع رمزگذاری گذرواژه در سیستم عامل متفاوت است و عملیات رمزگذاری در سیستم مدیریت پایگاه داده‌ها انجام می‌شود. برای اطمینان از رمزشدن گذرواژه توجه به نکات زیر ضروری است:

- اگر برای ایجاد حساب کاربری از دستور GRANT استفاده می‌کنید، برای رمزگذاری گذرواژه نیازی به استفاده از تابع PASSWORD() نیست. زیرا این دستور به طور پیش فرض گذرواژه را رمز می‌کند.

```
mysql> GRANT SELECT ON *.* TO 'ali'@'x.com' IDENTIFIED BY 'my-pass';
```

- در شرایطی که مستقیماً از دستور INSERT بر روی جدول user برای ایجاد حساب کاربری استفاده می‌کنید، استفاده از تابع PASSWORD() ضروری است، در غیر این صورت گذرواژه به شکل متن ساده و رمز نشده ذخیره می‌گردد.

```
mysql> INSERT INTO mysql.user (Host, User, Password)
-> VALUES('x.com', 'ali', PASSWORD('my-pass'));
mysql> FLUSH PRIVILEGES;
```

[۱۸] Hash function

- برای تغییر گذرواژه نیز اگر مستقیماً از دستور UPDATE بر روی جدول user استفاده می‌کنید، حتماً از تابع PASSWORD() برای رمزگذاری گذرواژه استفاده کنید.

```
mysql> UPDATE mysql.user SET Password = PASSWORD('new-pass')
-> WHERE Host = 'x.com' AND User = 'ali';
mysql> FLUSH PRIVILEGES;
```

- توجه داشته باشید که اگر از دستور SET PASSWORD برای تغییر گذرواژه استفاده می‌کنید، باید برای رمزکردن گذرواژه از تابع PASSWORD() استفاده کنید.

```
mysql> SET PASSWORD FOR 'ali'@'.com' = PASSWORD('new-pass');
```

استفاده ایمن از گذرواژه

با استفاده از روش‌های مختلفی می‌توان گذرواژه را برای اتصال به کارگزار ارائه کرد. برای مثال می‌توان گذرواژه را در فایل پیکربندی کارگزار نگاه داشت تا در هنگام اتصال به کارگزار ارسال شود، و یا می‌توان در زمان اتصال به کارگزار در خط فرمان[۱۹] گذرواژه را وارد کرد. در هریک از این روش‌ها توصیه‌هایی برای برقراری امنیت بیشتر وجود دارد.

- کاربران در پاره‌ای از موارد برای اتصال سریع به کارگزار پایگاه داده‌های MySQL گذرواژه حساب کاربری خود را در فایل پیکربندی نگهداری می‌کنند. در این شرایط گذرواژه به صورت متن ساده و رمز نشده می‌باشد، به این ترتیب در دسترس کسانی خواهد بود که به این فایل دسترسی دارند. برای جلوگیری از این مشکل حتماً مجوزهای دسترسی به فایل پیکربندی را چک کنید. هرگونه دسترسی به این فایل فقط باید محدود به کاربر مالک فایل باشد که حق اتصال به کارگزار پایگاه داده‌ها را دارد (برای مثال در لینوکس حقوق دسترسی به این فایل باید به خواندن و نوشتن توسط مالک فایل یا 600 تغییر یابد).
- گذرواژه نباید در دستور درخواست برقراری ارتباط با کارگزار MySQL در خط فرمان با استفاده از گزینه‌های -ppassval و --password= passval آورده شود، زیرا به این طریق به طور واضح قابل

[۱۹] Command line

مشاهده از طرف دیگران است. در عوض می‌توان از p- و password-- بدون بیان مقدار گذرواژه در ادامه گزینه استفاده کرد. در این صورت در ابتدا و قبل از برقرارای ارتباط، گذرواژه از کاربر پرسیده می‌شود، که با کاراکتر * نمایش داده می‌شود.

- MySQL اجازه مشخص کردن گذرواژه را در متغیر محیطی MYSQL_PWD می‌دهد. اما نباید از این امکان استفاده کرد؛ زیرا دیگر کاربران سیستم عامل با پردازش وضعیت سیستم می‌توانند اطلاعات محیطی کاربر را مشاهده کنند و در نتیجه به گذرواژه وی دست یابند.

۶. کنترل دسترسی‌های کاربران

در MySQL از روش ماتریسی ACL[۲۰] برای کنترل دسترسی کاربران استفاده شده است. حقوق کاربر می‌توانند در سطوح مختلفی تعریف شوند:

- حقوق می‌توانند در سطح کاربر باشند، که در جدول mysql.user مشخص می‌شوند، این سطح، حقوق کلی کاربر که بر همه پایگاه‌های داده‌ها قابل اعمال است، و نیز حقوق مدیریتی کاربر را مشخص می‌کند، مانند حق Drop_priv، Insert_priv، Shutdown_priv، File_priv و ... .
- برخی حقوق در سطح میزبان هستند، که در جدول mysql.host آورده می‌شوند. حقوق کاربرانی که از میزبانی خاص به کارگزار متصل می‌شوند بر هر پایگاه داده‌ها در این سطح قرار دارد، برای مثال حقوق Insert_priv، Grant_Prive، Update_priv و ... .
- حقوق سطح پایگاه داده‌ها در جدول mysql.db نگهداری می‌شود. این سطح حقوق هر کاربر را از هر میزبان بر هر پایگاه داده‌ها بیان می‌کند، مانند حقوق Grant_priv، Create_priv، Execute_priv و ... .
- حقوق ممکن است در سطح جداول پایگاه‌های داده‌ها باشند، که در جدول mysql.tables_priv نگهداری می‌شوند. یا در سطح ستون‌های جداول باشند، که در جدول mysql.columns_priv قرار

[۲۰] Access Control List

دارند. همچنین می‌توانند در سطح روال‌های ذخیره شده در پایگاه‌های داده‌ها باشند، که در جدول procs_priv مشخص می‌شوند. هریک از این حقوق می‌تواند شامل مجوزهای Insert، Delete، Select و ... باشند.

به این ترتیب، برای برقرای امنیت در کنترل دسترسی کاربران توجه به نکات زیر پیشنهاد می‌شود:

- حقوق هر کاربر باید با توجه به فعالیت‌های وی مشخص و به او اختصاص یابند. بنابراین باید از اعطای حقوق اضافی به کاربران پرهیز شود. اعطا یا ابطال مجوزهای کاربران می‌تواند با استفاده از دستورات GRANT و REVOKE و یا دسترسی مستقیم به جداول مجوزهای مربوطه صورت گیرد. در صورت تغییر مستقیم مجوزها با دسترسی به جداول مجوزها، باید از دستور FLUSH PRIVILEGES برای اعمال تغییرات استفاده کرد.
- حقوق مدیریتی مانند Process، File، Super و Shutdown را به کاربران عادی و غیر مدیر اعطا نکنید. زیرا برای مثال کاربری که دارای حق Process است، می‌تواند با استفاده از دستور SHOW PROCESSLIST متن همه دستوراتی که در حال اجرا هستند را ببیند و به اطلاعات زیادی دست یابد، مثلاً دستور تغییر گذرواژه کاربر دیگری را می‌تواند به طور کامل مشاهده کرده و به گذرواژه وی پی ببرد. کاربری که حق Super را دارد، می‌تواند اتصال کارخواهان دیگر را قطع کند، و یا با تغییر متغیرهای سیستمی نحوه عملکرد کارگزار را تغییر دهد. همچنین اگر کاربری مجوز File را داشته باشد، می‌تواند فایل‌های سیستم را تغییر دهد و یا با استفاده از دستور LOAD DATA INFILE محتویات فایل‌های خارجی (مثلاً /etc/passwd) را به جدولی ریخته و با دستور SELECT محتویات آن را مشاهده کند. یک کاربر با داشتن مجوز Shutdown می‌تواند کارگزار را از کار انداخته و غیر فعال سازد.
- در صورت امکان، تعداد اتصالات ممکن از هر حساب کاربری را در هر لحظه محدود کنید. برای این منظور می‌توانید از دستور GRANT به شکل زیر استفاده کنید. با قرار دادن مقدار گزینه

MAX_USER_CONNECTIONS به عدد مورد نظر، امکان سوء استفاده و برقراری اتصال از طریق یک حساب کاربری معتبر را توسط مهاجمین از بین می‌برید.

```
mysql> GRANT SELECT ON *.* TO 'Umanager'@'x.com'
    -> IDENTIFIED BY 'my-pass' WITH MAX_USER_CONNECTIONS = 1;
```

- تا حد ممکن از دیدها[۲۱] و اعطای مجوز بر روی دیدها برای محدود کردن دسترسی‌های کاربران استفاده کنید. با استفاده از دیدها کاربران به بخشی از داده‌های جداول دسترسی داشته و اجازه مشاهده و تغییر همه داده‌های جدول را ندارند.

۷. ذخیره‌سازی امن اطلاعات

داده‌های حساس را نباید به شکل ساده و رمز نشده در جداول نگهداری کرد. زیرا چنانچه فایل حاوی اطلاعات در اختیار کاربران سود جو قرار گیرد، اگر داده‌ها رمز نشده باشند می‌توانند به اطلاعات ستون‌های جدول دست یابند. در MySQL دستورات متفاوتی برای رمزکردن داده‌ها بر اساس الگوریتم‌های مختلف وجود دارند. با استفاده از این دستورات می‌توان داده موجود در یک ستون خاص از پایگاه داده‌ها را رمز کرد. جدول زیر توابع رمزگذاری، رمزگشایی و درهم‌ساز MySQL را نشان می‌دهد.

دستور	توضیحات
AES_ENCRYPT(str, key_str)	بر اساس الگوریتم AES داده را رمز می‌کند (کلید ۱۲۸ بیتی است).
AES_DECRYPT(crypted, key_str)	بر اساس الگوریتم AES داده را رمز گشایی می‌کند.
ENCODE(str, pass_str)	داده رشته‌ای را با استفاده از یک کلمه رمز ورودی، رمز می‌کند.
DECODE(crypted, pass_str)	داده رمز شده را به وسیله کلمه رمز ورودی، رمزگشایی می‌کند.
DES_ENCRYPT (str[,(key_num|key_str)])	بر اساس الگوریتم DES داده را رمز می‌کند.
DES_DECRYPT(crypted[, key_str])	بر اساس الگوریتم DES داده رمزشده را رمز گشایی می‌کند.
MD5(str)	ایجاد مجموع مقابله‌ای[۲۲] ۱۲۸ بیتی ورودی با الگوریتم MD5
SHA1(), SHA()	ایجاد مجموع مقابله‌ای ۱۶۰ بیتی ورودی با الگوریتم SHA1

[۲۱] View
[۲۲] Checksum

در هنگام پرس و جو و بازیابی اطلاعات، باید داده رمزشده را رمزگشایی کرد. توابع رمزگذاری ودرهم‌سازی، در پاسخ، مقادیر دودویی بر می‌گردانند. برای داشتن ستونی در جدول که داده آن از نوع رمز شده است، باید نوع داده آن را BLOB گذاشت و از انتخاب انواع داده CHAR و VARCHAR اجتناب کرد. زیرا در این نوع داده‌ها، تدابیری برای بهینه‌سازی صورت می‌گیرد (برای مثال حذف فضای خالی[۲۳])، که ممکن است موجب خراب شدن و تغییر داده رمز شده گردد. برای مثال فرض کنید ستون حقوق در جدول کارمندان یک شرکت داده حساس محسوب شده و باید رمز شده نگهداری شود، به این ترتیب برای اضافه کردن داده به این ستون و یا مشاهده مقادیر آن باید به شکل زیر عمل کرد.

```
mysql> CREATE TABLE Employee (name VARCHAR(30), salary BLOB);
mysql> INSERT INTO Employee VALUES
    -> ('Ali', AES_ENCRYPT('320000','my_pass_string')),
    -> ('Ahmad', AES_ENCRYPT('400000','my_pass_string')),
    -> ('Babak', AES_ENCRYPT('380000','my_pass_string'));
mysql> SELECT name, AES_DECRYPT(salary,'my_pass_string')
    -> FROM Employee;
```

برای ایمنی بیشتر بهتر است کلید رمزنگاری را همراه با هر پرس و جو به شکل خام به کارگزار ارسال نکنید. در این صورت می‌توانید کلید رمزنگاری را در متغیری نگاه داشت و در هنگام اجرای پرس و جو از آن متغیر استفاده کرد. برای مثال می‌توان به شکل زیر عمل کرد:

```
mysql> SELECT @key_val:='my_pass_string';
mysql> INSERT INTO Employee VALUES
    -> ('Ali', AES_ENCRYPT('320000',@key_val)),
    -> ('Ahmad', AES_ENCRYPT('400000', @key_val)),
    -> ('Babak', AES_ENCRYPT('380000', @key_val));
mysql> SELECT name, AES_DECRYPT(salary, @key_val) FROM Employee;
```

۸. پشتیبان‌گیری و بازیابی اطلاعات

یکی از مهمترین وظایف مدیران پایگاه‌های داده‌ها، پشتیبان‌گیری مداوم از داده‌ها و جداول موجود و بازیابی اطلاعات در زمان بروز مشکل و خرابی در داده‌های اصلی است. MySQL روش‌های مختلفی را برای

[۲۳] Space

پشتیبان‌گیری و بازیابی پایگاه‌های داده‌ها فراهم می‌آورد. استفاده مستقیم از دستورات SQL و یا برنامه‌های جانبی چون mysqldump و mysqlhotcopy از این قبیل هستند.

پشتیبان‌گیری و بازیابی اطلاعات در سطح دستورات SQL

در شرایطی که می‌خواهید در سطح SQL از داده‌های جداول خود پشتیبان بگیرید، می‌توانید از دستور SELECT…INTO OUTFILE استفاده کنید و داده‌های خود را در فایل رشته‌ای پشتیبان قرار دهید.

```
mysql> SELECT * FROM db-name.tbl-name INTO OUTFILE table_name.txt;
```

طبیعتاً برای بازیابی اطلاعات از این فایل‌های پشتیبان باید از دستور LOAD DATA INFILE و یا برنامه جانبی mysqlimport استفاده کرد.

```
mysql> LOAD DATA INFILE table_name.txt INTO TABLE db_name.tbl-name;
Shell> mysqlimport [options] db_name text_file1 [text_file2 …];
```

دستور mysqlimport با توجه به نام فایل رشته‌ای، داده را در جداولی در پایگاه داده‌ها کپی می‌کند. برای مثال چنانچه نام فایل رشته‌ای یکی از سه رشته patient.text، patient.txt و یا patient باشد، اطلاعات را در جدول patient در پایگاه داده‌های مشخص شده قرار می‌دهد.

استفاده از دستور BACKUP TABLE روش دیگری برای تهیه فایل پشتیبان در MySQL است. با استفاده از این دستور داده‌های جدول مورد نظر را می‌توان در فایل پشتیبان قرار داد. البته این دستور فقط قابل استفاده برای جداولی است که از موتور ذخیره‌سازی MyISAM استفاده می‌کنند. برای اجرای این دستور باید جداول خود را در برابر خواندن قفل نمایید. به هر حال، استفاده از این دستور پیشنهاد نمی‌گردد. استفاده از دستورات SQL برای پشتیبان‌گیری عموماً دارای نواقص امنیتی هستند که استفاده از آنها را مناسب نمی‌سازد. بهتر است از برنامه‌ای جانبی mysqldump و mtysqlhotcopy برای پشتیبان‌گیری استفاده کنید. با کمک این برنامه‌ها نه تنها قادر خواهید بود از کل پایگاه داده‌های خود پشتیبان بگیرید، بلکه به سرعت بیشتری نیز دست خواهید یافت.

پشتیبان‌گیری با استفاده از برنامه جانبی mysqldump

کارخواه mysqldump برای تهیه رونوشت[۲۴] از یک یا چند پایگاه داده‌ها استفاده می‌شود. از این رونوشت می‌توان به عنوان پشتیبان و یا برای انتقال پایگاه‌های داده‌ها به یک کارگزار SQL (صرفاً منظور کارگزار MySQL نیست) استفاده کرد. رونوشت حاصل از این دستور حاوی دستورات SQL مورد نیاز برای تهیه پایگاه‌های داده‌ها (ایجاد و مقداردهی جداول) است. اگر قصد دارید پشتیبان گیری را بر روی کارگزار انجام دهید و همچنین جداول شما از نوع MyISAM هستند، بهتر است از دستور mysqlhotcopy استفاده کنید. زیرا این دستور فرایندهای پشتیبان‌گیری و بازیابی اطلاعات را با سرعت بیشتری انجام می‌دهد.

برای استفاده از mysqldump می‌توان از سه روش اصلی زیر استفاده کرد:

```
1. >mysqldump [options] db_name [tables] [> 'BackupFile.sql']
2. >mysqldump [options] --databases DB1[ DB2 ...][>'BackupFile.sql']
3. >mysqldump [options] --all-databases [> 'BackupFileDir']
```

در روش اول می‌توان پشتیبان را از جداول خاصی از یک پایگاه داده‌ها تهیه کرد. در روش دوم پشتیبان گیری در سطح چند پایگاه داده‌ها که نام آنها مشخص شده است انجام می‌شود. در نهایت با استفاده از روش سوم می‌توان از کل پایگاه‌های داده‌ها پشتیبان‌گیری کرد.

با مشخص کردن مسیر فایل پشتیبان، خروجی دستور mysqldump در آن فایل قرار می‌گیرد. برای اجرای پشتیبان‌گیری، این برنامه باید به کارگزار MySQL متصل شود، بنابراین باید در کارگزار MySQL یک حساب کاربری که دارای مجوزهای مورد نیاز (حداقل دارای SELECT بر روی پایگاه‌های داده‌های مورد نظر برای پشتیبان‌گیری) باشد را ایجاد کرده، و سپس در زمان اجرای mysqldump در بخش [options] نام کاربری و گذرواژه را به کارگزار ارسال کرد. برای مثال دستور زیر از محتویات یک جدول از پایگاه داده‌های dbtest رونوشت تهیه می‌کند.

```
C:\> mysqldump --user=uback -p testdb table1 > c:\testdb_t1.sql
```

[۲۴] Dump

توجه داشته باشید که اگر از گزینه‌های اختیاری quick-- و opt-- در دستور mysqldump استفاده نکنید، قبل از تهیه رونوشت از نتایج دستور، کلیه مجموعه‌های نتایج در حافظه بارگزاری می‌شوند. در این صورت چنانچه از پایگاه داده‌های بزرگی پشتیبان می‌گیرید، سرعت بسیار کاهش می‌یابد. البته گزینه opt-- به طور پیش‌فرض فعال می‌باشد و با استفاده از گزینه skip-opt-- غیر فعال می‌گردد.

برای بازیابی اطلاعات از روی رونوشت تهیه شده توسط mysqldup، می‌توان از دستور mysql در خط فرمان استفاده کرد. برای این کار باید با حساب کاربری مناسب به کارگزار MySQL متصل شده و محتویات پایگاه داده‌های مورد نظر را به هنگام ساخت. برای مثال با استفاده از دستور زیر محتویات فایل پشتیبان به پایگاه داده‌های خاصی از MySQL منتقل می‌گردد.

```
C:\> mysql -u uback -p testdb < c:\testdb_back.sql
```

اگر پشتیبان‌گیری از کل یک یا چند پایگاه داده‌ها انجام گرفته باشد، باید در دستور فوق نام پایگاه داده‌ها را دیگر وارد نکرد. پایگاه‌های داده‌ها با نام‌های قبلی خود بازیابی می‌گردند.

یکی از مصارف mysqldump کپی کردن یک پایگاه داده‌ها از روی یک کارگزار MySQL بر روی کارگزار دیگر است:

```
shell> mysqldump --opt db_name | mysql --host=remote_host -C db_name
```

اگر جداول با موتور ذخیره‌سازی InnoDB ذخیره شده باشند و بخواهیم پشتیبان‌گیری را به صورت بر-خط[۲۵] انجام دهیم، می‌توانیم از mysqldump با گزینه single-transaction-- استفاده کنیم. به این ترتیب در ابتدای اجرای دستور، جداول در برابر خواندن با دستور FLUSH TABLES WITH READ LOCK قفل می‌شوند:

```
shell> mysqldump --all-databases --single-transaction > all databases.sql
```

برای بازیابی اطلاعات تا لحظه‌ای معین، در شرایطی که همه اطلاعات در فایل پشتیبان تهیه شده با mysqldump وجود نداشته باشند، باید از اجرای دستورات ثبت شده در گزارش ثبت دودویی وقایع[۲۶]

[۲۵] Online
[۲۶] Binary log

استفاده کرد (البته گزارش ثبت دودویی وقایع باید با انتخاب گزینه log-bin در فایل پیکربندی فعال شده باشد). برای این کار می‌توان بعد از بازیابی اطلاعات پشتیبان موجود، تغییرات بعدی را با اجرای فایل‌های ثبت دودویی وقایع مورد نظر اعمال کرد:

```
shell> mysql -u uback -p < all_databases.sql
shell> mysqlbinlog hostname-bin.[0-9]* | mysql
```

اگر در جداول MyISAM مشکلی پیش آمده و می‌خواهید آنها را بازیابی کنید، ابتدا بهتر است از دستور myisamchk -r و یا REPAIR TABLE استفاده کنید، در ۹۹.۹٪ موارد ایرادات به وجود آمده با اجرای این دستورات برطرف می‌گردند. اگر از این طریق مشکل به وجود آمده برطرف نگردید، همان روش قبلی را اجرا کنید (یعنی بعد از بازیابی فایل پشتیبان، دستورات ذخیره شده در ثبت دودویی وقایع را باز اجرا کنید).

پشتیبان‌گیری با استفاده از برنامه جانبی mysqlhotcopy

دستور mysqlhotcopy در اصل یک اسکریپت Perl است که از دستورات LOCK TABLES، FLUSH TABLES و cp یا scp برای ایجاد سریع پشتیبان از یک پایگاه داده‌ها استفاده می‌کند. استفاده از این دستور سریع‌ترین روش برای گرفتن پشتیبان از یک پایگاه داده‌ها و یا جداول است. اما این دستور فقط بر روی همان ماشینی که فایل‌های پایگاه‌های داده‌ها در آن قرار دارند، قابل اجراست. این دستور فقط برای پشتیبان‌گیری از جداول MyISAM و ISAM بر روی سیستم عامل Unix قابل استفاده است.

```
shell> mysqlhotcopy db_name [/path/to/new_directory]
shell> mysqlhotcopy db_name_1 ... db_name_n /path/to/new_directory
```

برای پشتیبان گیری از جداول، می‌توان نام جداول را با استفاده از عبارات منظم[۲۷] مشخص کرد، از علامت ~ نیز برای قبول نقض عبارت استفاده می‌شود.

```
shell> mysqlhotcopy db_name./regex/
shell> mysqlhotcopy db_name./~regex/
```

[۲۷] Regular expression

برای اجرای این دستور باید اجازه دسترسی به فایل‌های پایگاه داده‌هایی که از آنها پشتیبان‌گیری به عمل می‌آید داشت. همچنین کاربری که این دستور را اجرا می‌کند باید دارای حق SELECT بر روی جداول هدف و حق RELOAD (برای اجرای دستور FLUSH TABLES) در MySQL باشد.

۹. فعال‌سازی ثبت وقایع

در MySQL چندین نوع ثبت وقایع[۲۸] صورت می‌گیرد، که فعالیت‌های انجام شده از قبیل دسترسی‌های موفق و ناموفق به کارگزار، پرس و جوهای انجام شده بر پایگاه‌های داده‌ها و خطاها و هشدارها با زمان وقوع در آنها ثبت می‌گردند. از این وقایع ثبت شده نه تنها برای بازیابی اطلاعات در شرایط بروز خرابی و مشکل می‌توان استفاده کرد، بلکه می‌توان برای شناسایی حملات و کاربران مهاجم که حتی فقط اقدام به تجاوز به کارگزار را کرده‌اند، استفاده کرد. پیشنهاد می‌شود در صورت امکان گزارشات ثبت وقایع را فعال سازید، زیرا به این طریق می‌توان فعالیت‌های رخ داده در کارگزار را پی‌گیری کرده و در صورت بروز خطا از آنها استفاده کرد. البته در پاره‌ای از موارد برای بازیابی اطلاعات نیز می‌توان از این گزارشات استفاده کرد، که توضیح داده خواهد شد. به طور پیش‌فرض، کلیه گزارش‌های ثبت وقایع در مسیر داده (data) MySQL ایجاد شده و نگهداری می‌شوند. چهار نوع گزارش اصلی که توسط MySQL از وقایع ارائه می‌شود، به شرح زیر است:

گزارش ثبت خطا

گزارش ثبت خطا[۲۹]، مشکلاتی که در زمان آغاز به کار، اجرا و یا خاموش شدن کارگزار MySQL رخ می‌دهد، را در بردارد. اگر فایل اجرایی کارگزار (mysqld) به دلایل نامعلومی از کار بیفتد و mysqld_safe مجبور به راه اندازی مجدد آن شود، mysqld_safe پیغامی مبنی بر اجرای مجدد mysqld را در فایل گزراش ثبت خطا می‌نویسد. به علاوه، اگر جدولی نیاز به بررسی و یا تعمیر خودکار داشته باشد، پیغامی در این گزارش نوشته می‌شود. به صورت دلخواه می‌توان از MySQL خواست تا هشدارها را نیز در این گزارش ثبت کند.

[۲۸] Logging
[۲۹] Error log

برای فعال‌سازی این گزارش ثبت وقایع باید mysqld را با گزینه اختیاری [log-error[=file_name-- اجرا کرد و یا در فایل پیکربندی کارگزار، تنظیم log-error=error-log-file را باید به تنظیمات بخش mysqld اضافه کرد. با اجرای دستور FLUSH LOGS فایل ثبت خطاها با پسوند old- ذخیره شده و فایل خالی جدیدی برای ثبت وقایع خطاهای جدید با همان نام قبلی ایجاد می‌شود.

گزارش ثبت پرس و جوی جامع

گزارش ثبت پرس و جوی جامع[۳۰]، از اتصالات برقرار شده توسط کارخواهان و دستوراتی که از جانب آنها دریافت شده است، گزارشی در اختیار مدیر پایگاه داده‌ها قرار می‌دهد. این گزارش زمانی مفید است که مشکوک به رخ دادن خطایی در یک کارخواه باشیم، در این صورت می‌توان دستوراتی که وی به کارگزار ارسال کرده است را بررسی کرد. در این گزارش دستورات کاربر به همان ترتیبی که دریافت شده‌اند ثبت می‌شوند (ممکن است با ترتیب اجرای آنها متفاوت باشد). برای فعال‌سازی این گزارش باید mysqld را با گزینه [log[=log-file-name-- اجرا کرد و یا در فایل پیکربندی کارگزار MySQL تنظیم زیر را اضافه کرد:

```
log=query-log-file
```

چنانچه برای فایل گزارش نامی انتخاب نگردد، به عنوان پیش‌فرض نام host_name.log برای ثبت گزارش پرس و جوها انتخاب می‌گردد.

گزارش ثبت دودویی وقایع

گزارش ثبت دودویی وقایع، گزارشی حاوی کلیه دستوراتی است که داده‌ای را تغییر می‌دهند و یا برای پاسخ‌دهی استفاده می‌شوند. دستوراتی که قابلیت انجام تغییری را داشته باشند، اما موفق به ایجاد تغییر نشوند (مانند دستور DELETE که شرط آن برقرار نباشد و سطری را حذف نکند)، نیز در این گزارش ثبت می‌شوند. همچنین، در این گزارش مدت زمانی که هر کدام از دستورات برای اعمال تغییرات صرف کرده‌اند، بیان می‌شود. در نسخه‌های قبل از MySQL 5.0 به جای این گزارش، گزارش ثبت تغییرات[۳۱] وجود داشت.

---

[۳۰] General query log
[۳۱] Update log

گزارش دودویی نه تنها کلیه اطلاعات آورده شده در گزارش ثبت تغییرات را در بر دارد، بلکه به شکلی کاراتر و ایمن‌تر به ممیزی تراکنش‌ها می‌پردازد. هدف اصلی از ایجاد این گزارش، به‌هنگام سازی تا حد ممکن پایگاه داده‌ها با اجرای دستورات ذخیره شده است. زیرا این گزارش کلیه دستوراتی که تغییری در داده‌ها ایجاد کرده‌اند را بعد از فرایند پشتیبان‌گیری در بر دارد. بنابراین با بازیابی مجدد پایگاه داده‌ها از روی آخرین فایل پشتیبان و اجرای مجدد دستورات آورده شده در گزارش ثبت دودویی، می‌توان کل پایگاه داده‌ها را بازیابی کرد. برای فعال‌سازی این گزارش باید mysqld را با گزینه [log-bin[=file-name.extention-- اجرا کرد و یا تنظیم زیر را به فایل پیکربندی کارگزار MySQL افزود. با مشخص کردن پسوند عددی فایل ثبت دودویی وقایع، با هر بار راه اندازی کارگزار MySQL، عدد یک به مقدار پسوند فایل افزوده می‌شود. برای مثال اگر پسوند آخرین فایل 00020 باشد، در اجرای بعدی کارگزار فایلی با پسوند 00021 برای ثبت وقایع ایجاد می‌گردد.

```
Log-bin=bin-log-file.[0-9]*
```

همانطور که پیشتر گفته شد، با استفاده از دستور mysqlbinlog کلیه دستورات آورده شده در فایل گزارش ثبت دودویی وقایع مشخص شده، مجدد اجرا می‌گردند.

```
shell> mysqlbinlog [options] log-file1 [log-file2 ...]
```

ثبت پرس و جوی کند

ثبت پرس و جوی کند[۳۲]، پرس و جوهایی که بیش از زمان مورد انتظار طول بکشند و یا از شاخص[۳۳] استفاده نکنند را در بر دارد. زمان مورد انتظار برای اجرای هر پرس و جو در متغیر long_query_time نگهداری می‌شود، که حداقل آن ۱ و پیش‌فرض آن ۱۰ است. زمان صرف‌شده برای قفل کردن اولیه جدول در زمان اجرای پرس و جو حساب نمی‌شود. زمان اجرای هر پرس و جو بعد از اجرای کامل آن و آزاد شدن همه جداول قفل شده، در این گزارش نوشته می‌شود. بنابراین ترتیب پرس و جوهای آورده شده در گزارش

[۳۲] Slow query log
[۳۳] Index

ممکن است با ترتیب اجرای آنها متفاوت باشد. هدف از به دست آوردن این گزارش، یافتن پرس و جوهایی که زمان زیادی را برای اجرا صرف می‌کنند، و استفاده از این اطلاعات در راستای بهینه‌سازی آنها است.

برای فعال ساختن این گزارش باید از گزینه [--log-slow-queries[=file-name در زمان اجرای mysqld استفاده کرد. همچنین برای ثبت اجرای وقایع در هر بار اجرای کارگزار می‌توان عبارت زیر را به فایل پیکربندی افزود:

```
Log-slow-queries=log-file-name
```

منابع

[1] http://dev.mysql.com/doc/#manual
[2] http://www.securityfocus.com/infocus/1726
[3] http://krillz.com/secure-mysql/
[4] http://www.securityfocus.com/infocus/1667
[5] http://www.kitebird.com/articles/ins-sec.html
[6] http://downloads.onestopwebhosting.com/mysql.pdf
[7] http://www.mysql.com/products/backup/
[8] Ron Ben Natan, "Implementing Database Security and Auditing", Digital Press, 2005.
[9] http://www.snailbook.com/faq/ssl.auto.html
[10] http://www.builderau.com.au/program/mysql/soa/Six-steps-to-secure-sensitive-data-in-MySQL/0,339028784,339266102,00.htm