بسمه تعالی

سازمان فناوری اطلاعات ایران

معاونت امنیت فضای تولید و تبادل

اطلاعات

مرکز ماهر

مقاوم‌سازی امنیتی MySQL

فهرست مطالب

پیشگفتار

در این گزارش، مقاوم‌سازی[1] امنیتی MySQL نسخه 5.7 بر روی سیستم‌عامل Ubuntu 17.04 64-bit مورد بحث و بررسی قرار می‌گیرد. در این راستا، برای هر یک از پارامترهای امنیتی تاثیرگذار، شرح مختصری ارایه می‌گردد و سپس تهدید/توجیه امنیتی آن مورد بررسی قرار می‌گیرد. در نهایت نیز نحوه اطلاع از وضعیت فعلی پارامتر و چگونگی مقاوم‌سازی آن تشریح می‌گردد.

بررسی پارامترهای مربوط به مقاوم‌سازی PostgreSQL در پنج بخش مختلف صورت می‌گیرد. در بخش اول، تنظیمات مربوط به امن‌سازی محیط اجرای سیستم مدیریت پایگاه داده (سمپاد) مورد بحث و بررسی قرار می‌گیرد. در بخش دوم، پارامترهای مربوط به نصب و پیکربندی امن پایگاه داده تشریح می‌گردد. در بخش سوم، تنظیمات مربوط به امن‌سازی اتصال به پایگاه‌داده بررسی می‌گردد. در بخش چهارم، تنظیمات مربوط به رویدادنگاری تشریح می‌شود. در بخش پنجم نیز برخی از پیکربندی‌های با سطح حساسیت پایین‌تر آورده شده است. در پایان گزارش نیز، نحوه اجرای اسکریپت‌ها و اعمال تنظیمات مورد نیاز برای مقاوم سازی پایگاه داده بیان می‌گردد. لازم به ذکر است که برخی از توضیحات مهم امنیتی در رابطه با احراز اصالت کاربران، کنترل دسترسی و اقدامات قابل انجام پس از نصب سمپاد، در قالب رهنمون‌های امنیتی در پیوست‌های این گزارش آورده شده است.

[1] Hardening

۱ امن سازی محیط اجرا

در هر سیستم عامل، ابزارها و روش‌های مختلفی برای امن‌سازی محیط اجرا وجود دارد. پیکربندی نرم‌افزارهایی چون سیستم مدیریت پایگاه داده (سمپاد) نیز می‌بایست بسیار دقیق و با رویکرد امنیتی انجام شود. با این حال، برای ایجاد یک مدل جامع برای محافظت از اطلاعات و سرویس‌های اطلاعاتی، باید در سطح سیستم‌عامل نیز تمهیدات لازم اندیشیده شود. چرا که اگر تنظیمات سِرور پایگاه داده به خوبی صورت گرفته باشد اما تمهیدات امنیتی مورد نیاز بر روی فایل‌های آن در سطح سیستم‌عامل اعمال نشده باشد، یک مهاجم می‌تواند به سادگی از داده‌ها نسخه‌برداری کرده و از این طریق اطلاعات زیادی فاش شود.

در این بخش بر روی کنترل دسترسی فایل‌ها و پردازه‌ها در سیستم‌های مبتنی بر یونیکس که شاخص‌ترین آن‌ها لینوکس است، متمرکز می‌شویم. در این سیستم‌ها مجوزها بر روی سه دسته از کاربران مشخص می‌شود. اولین دسته مربوط به مالک است (u)، دسته دوم گروهی از کاربران که در گروه مالک قرار می‌گیرند (g) و دسته سوم دیگر کاربران (o) را مشخص می‌کند. مجوزهای اصلی نیز شامل خواندن (r)، نوشتن (w) و اجرا (x) می‌شود. لذا مجوزهای مربوط به هر فایل یا دایرکتوری با یک نه‌تایی نشان داده می‌شود. به عنوان مثال اگر مجوزهای فایلی به صورت rw-r--x باشد، بدین معنی است که مالک این فایل، حق خواندن و نوشتن، گروه کاربران متعلق به گروه مالک، حق خواندن و دیگر کاربران، حق اجرای این فایل را دارند. تنظیم دقیق این پارامترها نقش بسزایی در تأمین امنیت پایگاه داده دارد.

۱ – ۱ پیکربندی فایل تنظیمات

تنظیمات اصلی MySQL در فایل my.cnf قرار دارد، از این رو حفاظت از این فایل مهم است.

تهدید/توجیه امنیتی:

از آنجایی که در این فایل تنظیمات اصلی MySQL وجود دارد، تنها مالک این فایل یعنی root که مدیر سیستم است، حق تغییر این فایل را دارد. پس باید دقت کرد که اولاً مالک این فایل root باشد، ثانیا مجوز نوشتن تنها به مالک داده‌شده باشد.

اطلاع از وضعیت فعلی:

با استفاده از دستور زیر می‌توان حقوق دسترسی مربوط به فایل تنظیمات اصلی MySQL را مشاهده نمود:

```
ls    -lh    /etc/mysql/my.cnf
```

مقاوم‌سازی:

دستورات زیر به ترتیب مالکیت، گروه کاربری مالکیت و حقوق دسترسی به فایل را بـه صـورت امـن تنظـیم می‌نمایند:

```
chown   -R      root    /etc/mysql/my.cnf

chgrp   -R      root    /etc/mysql/my.cnf

chmod   644             /etc/mysql/my.cnf
```

۲ – ۱ پیکربندی دایرکتوری ذخیره داده

در MySQL، داده‌های اصلی (مانند جداول و غیره) به‌صورت پیش‌فرض در دایرکتوری زیر قرار دارند.

```
/var/lib/mysql
```

تهدید/توجیه امنیتی:

برخلاف فایل‌های دیگر، مدیر پایگاه داده نباید مالک این فایل باشد. مالکیت فایل باید متعلق به کاربری بدون هرگونه حقوق ممتاز (مثلا کاربری با نام mysql) باشد؛ زیرا این کاربر اجازه انجام هیچ عملیاتی داخل سیستم عامل را ندارد. همچنین هیچ‌کس دیگری به جز مدیر سمپاد نیز حق خواندن، تغییر یـا اجـرای ایـن فایـل را نباید داشته باشد. درنتیجه تمامی حقوق روی این فایل را از همه گرفته و مجوزهای مربوطه را تنها به mysql می‌دهیم.

اطلاع از وضعیت فعلی:

مجوزهای مربوط به فایل‌های موجود در این دایرکتوری باید به صورت rwxrwx--- mysql mysql باشد.

```
ls -lh /var/lib/mysql
```

مقاوم‌سازی:

دو دستور اول مالکیت و گروه مالکیت فایل را به mysql تغییر می‌دهد و دستور آخر مجوزها روی فایل را امن می‌کند.

```
chown   -R      mysql   /var/lib/mysql/*
chgrp   -R      mysql   /var/lib/mysql/*
chmod   770             /var/lib/mysql/*
```

۳ – ۱ امن سازی کاربر mysql

یکی از کاربردهای کاربر mysql در قسمت قبل ملاحظه شد. در این قسمت به امن سازی این کاربر پرداخته می‌شود.

تهدید/توجیه امنیتی:

پردازه MySQL نباید حق اجرای دستورات shell را داشته باشد. در این صورت حتی اگر مهاجم بتواند به این به پردازه نفوذ کند، باز هم قادر نخواهد بود به سطر فرمان سِرور دسترسی پیدا کند. برای این کار باید در فایل /etc/passwd/ آخرین ستون مربوط به کاربر mysql را از /bin/bash به /bin/false تغییر داد.

اطلاع از وضعیت فعلی:

با اجرای دستور زیر اطلاعات مربوط به این کاربر نمایش داده می‌شود.

```
grep   mysql        /etc/passwd
```

مقاوم‌سازی:

اجرای دستور زیر حق اجرای shell از mysql را سلب می‌کند.

```
sed   -i   's/mysql\(.*\):\/bin.*/mysql\1:\/bin\/false/'   /etc/passwd
```

۴ – ۱ پیکربندی فایل‌های رویدادنگاری

در MySQL تمام رویدادهای مرتبط با پایگاه داده در فایل‌های از قبل تعیین شده‌ای ثبت می‌شوند.

تهدید/توجیه امنیتی:

هیچ کاربری به جز کاربران ویژه‌ای همچون adm و mysql نباید حق خواندن یا نوشتن روی فایل‌های رویدادنگاری را داشته باشد. رعایت این مورد امنیتی از نشت اطلاعات به بیرون این فایل‌ها جلوگیری می‌کند. به عنوان مثال ممکن است یک دستورGRANT شامل اطلاعات مهمی باشد که به صورت رمز نشده در فایل رویدادنگاری ذخیره‌شده است. مالک این فایل mysql است.

اطلاع از وضعیت فعلی:

مجوزهای مربوط به این فایل باید به صورت rw-r-----mysql adm باشد.

```
ls     -lh  /var/log/mysql*
```

مقاوم‌سازی:

اجرای دستور فوق باعث امن شدن دسترسی‌ها به این دایرکتوری می‌شود.

```
chown   -R                  mysql    /var/log/mysql.*
```

```
chgrp   -R          adm          /var/log/mysql.*
chmod   640                      /var/log/mysql.*
chown               mysql        /var/log/mysql/mysql/*
chgrp               mysql        /var/log/mysql/mysql/*
chmod   640                      /var/log/mysql/mysql/*
```

۵-۱ جمع‌بندی

در این فصل به تشریح برخی از مهم‌ترین پارامترهای امنیتی محیط اجرای سمپاد که به طور مستقیم بر عملکرد آن تاثیرگذار است، پرداختیم. در این راستا، تنظیمات مربوط به حقوق دسترسی فایل تنظیمات، دایرکتوری ذخیره‌داده، امن‌سازی کاربر MySQL، راه‌اندازی سرویس MySQL توسط یک کاربر عادی و پیکربندی فایل رویدادنگاری مورد بحث و بررسی قرار گرفت. مدیر سامانه به منظور بررسی پارامترهای مقاوم‌سازی و تهیه گزارش در این زمینه می‌تواند از چک لیست زیر استفاده نماید.

	عنوان	تنظیم صحیح	
		بله	خیر
۲	**امن سازی محیط اجرا**		
۲-۱	پیکره‌بندی فایل تنظیمات	☐	☐
۲-۲	پیکربندی دایرکتوری ذخیره داده	☐	☐
۲-۳	امن‌سازی کاربر MySQL	☐	☐
۲-۴	راه اندازی سرویس MySQL توسط یک کاربر عادی	☐	☐
۲-۵	پیکربندی فایل رویدادنگاری	☐	☐

۲ پیکربندی امن پایگاه‌داده

انجام برخی اعمال پس از نصب MySQL ضروری است. به عنوان مثال دایرکتوری داده باید مقداردهی اولیه شود و جداول اعطا[۲]ایجاد شوند. در تمامی سیستم‌های عامل، یکی از ملاحظات اصلی امنیتی این است که حساب‌های اولیه در جداول اعطا حاوی هیچ گذرواژه‌ای نباشند. همچنین در MySQL برخی امکانات به طور پیش‌فرض برای تمامی کاربران وجود دارد. این مجوزها در برخی مواقع باعث مشکلات امنیتی می‌شود. همچنین برخی پایگاه‌های داده ایجاد می‌شوند که فقط به منظور آموزش استفاده می‌شوند. در ادامه به چگونگی تنظیم پارامترهای مربوط به پیکربندی امن پایگاه‌داده می‌پردازیم. لازم به ذکر است که علاقمندان می‌توانند برای کسب اطلاعات بیشتر در زمینه رهنمون‌های پس از نصب سمپاد، بخش مربوطه در پیوست را مشاهده نمایند.

۲ – ۱ پارامتر allow-suspicious-udfs

این پارامتر قابلیت بارگذاری توابع تعریف شده توسط کاربر را تعیین می‌نماید. به صورت پیش فرض این گزینه غیرفعال است و تنها توابعی که حداقل یک نشانه اضافی دارند می‌توانند بارگذاری شوند. این امر از بارگذاری توابع از فایل‌های شیئ اشتراکی جلوگیری می‌نماید.

تهدید/توجیه امنیتی:

پرهیز از داشتن کتابخانه‌های اشتراکی که شامل بارگذاری توابع تعریف شده توسط کاربر نیست، سطح حمله به سامانه را کاهش می دهد.

اطلاع از وضعیت فعلی:

با اجرای دستور زیر می‌توان از وضعیت فعلی این پارامتر در سامانه اطلاع پیدا کرد.

```
mysqld --verbose --help | grep allow-suspicious-udfs | awk '/allow-suspicious-udfs/{i++}i==2{print $2; exit}'
```

مقاوم‌سازی:

برای جلوگیری از داشتن کتابخانه‌های اشتراکی که شامل بارگذاری توابع تعریف شده توسط کاربر نباشد، از دستور زیر استفاده می‌کنیم:

---

[2] Grant tables

```
sed -i 's/\[mysqld\]/\[mysqld\]\nallow-suspicious-udfs/’     /etc/mysql/my.cnf
```

۲ – ۲ پارامتر create_user_priv

امتیاز CREATE USER، حقوق داده شده به یک کاربر به منظور اضافه کردن، حذف کردن، تغییر نام و لغو امتیازات کاربران را مدیریت می‌نماید.

تهدید/توجیه امنیتی:

کاهش تعداد کاربران دارای امتیاز CREATE USER، تعداد کاربران با قابلیت اضافه/حذف کاربران، تغییر نام کاربران و تغییر در امتیازات کاربران را کاهش می‌دهد. این امر سبب می‌شود تا رویدادهای ناخواسته حاصل از اعمال مذکور در سامانه کم شده و در نتیجه سطح امنیت افزایش یابد.

اطلاع از وضعیت فعلی:

با اجرای دستور زیر می‌توان از وضعیت فعلی این پارامتر در سامانه اطلاع پیدا کرد.

```
MYSQL_PWD=$password mysql --user=$user -e "SELECT user FROM mysql.user WHERE
    Create_user_priv = 'Y';" | awk '{if ($1 != "user") print $1}'
```

مقاوم‌سازی:

برای گرفتن این امتیاز از کاربران مورد نظر، از دستور زیر استفاده می‌گردد.

```
mysql --user=$user --password=$password -e "REVOKE CREATE USER ON *.* FROM ‘<User name>’;”
```

۲ – ۳ پارامتر secure-file-priv

این پارامتر مسیرهای استفاده شده بوسیله LOAD DATA INFILE یا SELECT local_file را محدود می‌کند. اغلب توصیه می‌شود که این پارامتر به یک مکانی از فایل سیستم که شامل تنها فایل‌های مورد نیاز برای بارگذاری سامانه MySQL است، آدرس دهی شود.

تهدید/توجیه امنیتی:

تنظیم صحیح secure_file_priv باعث کاهش توانایی مهاجم برای خواندن فایل‌های حساس از سرور آسیب دیده از طریق آسیب پذیری تزریق SQL می‌شود.

اطلاع از وضعیت فعلی:

با اجرای دستور زیر می‌توان از وضعیت فعلی این پارامتر در سامانه اطلاع پیدا کرد.

```
grep secure-file-priv /etc/mysql/my.cnf
```

مقاوم‌سازی:

به منظور تنظیم امن این پارامتر از دستور زیر بهره می‌گیریم.

```
sed -i 's/\[mysqld\]/\[mysqld\]\nsecure-file-priv = \/var\/lib\/mysql-files/' /etc/mysql/my.cnf
```

۲ – ۴ پارامتر skip-grant-tables

این پارامتر منجر به اجرای سِرور بدون استفاده از سیستم مجوزدهی می‌شود. این امر برای تمامی کاربرانی که به سِرور دسترسی دارند امکان دسترسی نامحدود به تمامی پایگاه‌های داده را فراهم می‌آورد. یک سِرور در حال اجرا با اجرای دستورات mysqladmin flush-privileges و mysqladmin reload، یا اجرای FLUSH PRIVILEGES مجدداً از سیستم مجوزدهی استفاده می‌نماید. این گزینه از بارگذاری plugin هایی که با استفاده از دستور INSTALL PLUGIN، یا توابع تعریف شده توسط کاربر و یا رخدادهای زمان‌بندی شده نصب شده‌اند، جلوگیری می‌نماید. به منظور بارگذاری pluginها از گزینه --plugin-load استفاده کنید. در صورتی که MySQL با استفاده از گزینه --disable-grant-options پیکربندی شده باشد، استفاده از --skip-grant-tables امکان‌پذیر نیست.

تهدید/توجیه امنیتی:

در صورت استفاده از این پارامتر، تمام مشتریان سامانه آسیب دیده، دسترسی نامحدود به تمام پایگاه‌های داده را خواهند داشت.

اطلاع از وضعیت فعلی:

با اجرای دستور زیر می‌توان از وضعیت فعلی این پارامتر در سامانه آگاه شد.

```
grep skip-grant-tables /etc/mysql/my.cnf
```

مقاوم‌سازی:

به منظور تنظیم امن این پارامتر از دستور زیر بهره می‌گیریم.

```
sed -i 's/\[mysqld\]/\[mysqld\]\nskip-grant-tables = FALSE/' /etc/mysql/my.cnf
```

۲ – ۵ پارامتر local_infile

در MySQL، دستوری به نام LOAD DATA LOCAL INFILE وجود دارد. با استفاده از دستور LOAD DATA می‌توان یک فایل را بر روی سِرور بارگذاری کرد. در صورت استفاده از گزینه LOCAL می‌توان فایلهای موجود در ماشین کاربر را بارگذاری نمود.

تهدید/توجیه امنیتی:

دو مشکل امنیتی در استفاده از گزینه LOCAL وجود دارد:

۱. انتقال فایل از ماشین کاربر به سِرور با دسترسی مستقیم سِرور MySQL به فایل مورد نظر صورت می‌پذیرد. این فایل می‌تواند با فایل مشخص شده در دستور LOAD DATA متفاوت بوده و امکان دسترسی سِرور به تمامی فایلهای مجاز کاربر را فراهم کند.
۲. در محیط وب که ارتباط یک کاربر از طریق سِرور وب انجام می‌شود، یک کاربر می‌تواند با استفاده از LOAD DATA LOCAL تمامی فایلهایی را که سِرور وب به آنها دسترسی خواندن دارد را بخواند. در چنین محیطی ماشین کاربر نسبت به سِرور MySQL در حقیقت نقش سِرور وب را دارا است.

غیرفعال کردن استفاده از این دستور می‌تواند از خواندن غیرمجاز داده‌های داخلی جلوگیری کند. این مسئله زمانی مهم می‌شود که یک برنامه مبتنی بر PHP نیز روی سِرور در حال اجرا باشد، چرا که در این صورت حملات تزریق SQL زیادی روی آن قابل اجرا خواهد بود. علاوه بر این، در برخی موارد با استفاده از این دستور می‌توان به فایل‌هایی مانند فایل گذرواژه‌ها دسترسی پیدا کرد. دستورات زیر نمونه‌ای از این حمله هستند:

```
mysql> LOAD DATA LOCAL INFILE '/etc/passwd' INTO TABLE table1
mysql> SELECT load_file("/etc/passwd")
```

هرچند که استفاده از دستور اول در نسخه‌های بالاتر MySQL غیرمجاز است اما دستور دوم قابل اجرا است.

اطلاع از وضعیت فعلی:

```
mysql> SHOW global VARIABLES like 'local_infile';
```

با اجرای این دستور اطلاعات مربوط به این پارامتر نمایش داده می‌شود. در صورتی که MySQL با استفاده از فایل منبع ایجاد شود اما configure با گزینه --enable-local-infile فراخوانی نشود، دستور LOAD DATA LOCAL برای یک کاربر قابل استفاده نخواهد بود مگر اینکه دستور زیر را فراخوانی نماید:

```
mysql_options(… MYSQL_OPT_LOCAL_INFILE, 0)
```

در صورتی که LOAD DATA LOCAL در سِرور و یا در ماشین کاربر غیرفعال باشد، کاربری که قصد استفاده از این دستور را داشته باشد، با پیغام خطای زیر مواجه می‌شود:

```
ERROR 1148: The used command is not allowed with this MySQL version
```

مقاوم‌سازی:

در صورتی که مقدار این متغیر در mysql تغییر یابد، مقدار جدید تنها در یک نشست اعمال می‌شود. با استفاده از پارامتر --local-infile=0 می‌توان تمامی دستورات LOAD DATA LOCAL را از سمت سِرور غیرفعال نمود:

```
sed -i 's/\[mysqld\]/\[mysqld\]\nlocal-infile = 0 /' \ /etc/mysql/my.cnf
```

به‌طور کلی گزینه local-infile در فایل my.cnf وجود ندارد. با اجرای این دستور این متغیر به فایل اضافه می‌شود و درنتیجه می‌توان از این دسته حملات جلوگیری کرد.

در ماشین کاربر و در خط فرمان mysql، امکان فعالسازی LOAD DATA LOCAL با استفاده از گزینه --local-infile[=1] وجود دارد. علاوه بر این، غیر فعالسازی این دستور با استفاده از --local-infile=0 امکانپذیر است. به صورت پیش‌فرض در mysqlimport، بارگذاری محلی فایل غیرفعال است؛ فعالسازی آن با استفاده از گزینه --local یا L– امکان‌پذیر است. با این وجود، استفاده از عملگر بارگذاری محلی تنها در صورتی که سِرور با آن موافقت نماید، امکان‌پذیر است.

۲-۶ پارامتر skip_show_database

با استفاده از این پارامتر، دستور SHOW DATABASES تنها برای کاربرانی که دارای مجوز SHOW DATABASES هستند مجاز است. این دستور نام تمامی پایگاه‌های داده را نمایش می‌دهد. بدون استفاده از این گزینه، دستور فوق در دسترس تمامی کاربران قرار دارد، اما تنها در صورتی نام یک پایگاه داده را نمایش می‌دهد که کاربر، مجوز ذکر شده یا مجوزی دیگر را روی این پایگاه‌داده داشته باشد. توجه داشته باشید که تمامی مجوزهای سراسری به عنوان مجوزی برای یک پایگاه داده‌ی خاص نیز محسوب می‌شوند.

تهدید/توجیه امنیتی:

به طور پیش‌فرض، حق دیدن لیست پایگاه داده‌ها به تمامی کاربران داده‌شده است. این اطلاعات می‌تواند برای یک مهاجم بسیار مفید باشد، پس لازم است این حق محدود شود و تنها به کاربران مورد اعتماد داده شود.

اطلاع از وضعیت فعلی:

برای مطلع شدن از مقدار این پارامتر از دستور زیر استفاده می‌شود.

```
mysql> SHOW   global      VARIABLES      like      '%skip_show_database%';
```

مقاوم‌سازی:

برای محدود کردن درخواست show databases، باید skip_show_database به فایل my.cnf اضافه شود.

```
sed   -i   's/\[mysqld\]/\[mysqld\]\nskip_show_database   /' \                /etc/mysql/my.cnf
```

۲-۷ پارامتر skip-symbolic-links

این پارامتر جلوی استفاده از پیوندها به جداول را می‌گیرد. این امر خصوصاً در مواقعی حائز اهمیت است که mysqld با استفاده از root اجرا شده باشد. در این صورت، تمامی کاربرانی که به دایرکتوری داده سِرور دسترسی نوشتن دارند می‌توانند تمام فایل‌های سیستم را تغییر دهند.

تهدید/توجیه امنیتی:

این پارامتر می‌تواند اقدامات سامانه را به سوی فایل‌ها و دایرکتوری‌های خاصی هدایت و در عملکرد عادی سامانه اختلال ایجاد کند.

اطلاع از وضعیت فعلی:

با اجرای دستور زیر می‌توان از وضعیت فعلی این پارامتر در سامانه اطلاع پیدا کرد.

```
MYSQL_PWD= <Password> mysql --user= <User name> -e "SHOW variables LIKE 'have_symlink';"| grep
    have_symlink | awk '{print $2}'
```

مقاوم‌سازی:

این پارامتر به صورت پیش‌فرض غیرفعال است و در صورت فعال بودن آن می‌توان از دستور زیر برای مقداردهی امن این پارامتر بهره گرفت.

```
sed -i 's/\(.*\)skip-symbolic-links\(.*\)/#skip-symbolic-links/'  /etc/mysql/my.cnf
```

۸ – ۲ حذف پایگاه داده تست

به طور پیش‌فرض در سمپاد MySQL پایگاه داده‌ای به نام test وجود دارد. این پایگاه داده می‌تواند توسط کاربران قابل‌دسترسی باشد. لازم به ذکر است که وجود این پایگاه می‌تواند مخاطرات امنیتی را به سامانه تحمیل نماید.

تهدید/توجیه امنیتی:

در صورتی که پایگاه داده‌ای وجود داشته باشد که بتوان به‌صورت بی‌نام با آن کار کرد، احتمال اینکه مهاجم بتواند به اطلاعات مهمی دست‌یابد یا خرابی در آن ایجاد کند، در حالی که قابل‌شناسایی نیز نباشد، بسیار بالا است.

اطلاع از وضعیت فعلی:

برای اطمینان از وجود یا عدم وجود این پایگاه داده می‌توان از دستور زیر استفاده کرد:

```
mysql> show        databases   like   'test';
```

در صورتی که خروجی این دستور یک پایگاه داده بود، باید آن را حذف کرد.

مقاوم‌سازی:

برای مقاوم‌سازی باید از دستور حذف پایگاه داده استفاده کرد:

```
mysql> drop database test;
```

۲-۹ تنظیمات فایل تاریخچه

در طول نصب MySQL، اطلاعات زیادی در مورد سیستم و فرآیند نصب در فایل تاریخچه ذخیره می‌شود. این فایل می‌تواند در حین عملیات نصب و یافتن ایرادات نصب بسیار مفید باشد. با استفاده از این فایل مدیر سیستم می‌تواند اشکالات را یافته و برطرف نماید؛ اما این اطلاعات پس از نصب دیگر استفاده‌ای ندارند. علاوه بر این اطلاعات، هرگاه دستوری در MySQL اجرا شود، در این فایل نگهداری می‌شود.

تهدید/توجیه امنیتی:

اطلاعات موجود در این فایل می‌تواند مورد تهاجم حمله‌کننده قرار گیرد. از طرفی ممکن است اطلاعات ذخیره‌شده حاوی گذرواژه نیز باشد که به صورت آشکار در این فایل ذخیره‌شده است. از این رو باید این فایل را حذف کرد. به عنوان مثال اگر در mysql دستور ایجاد یک کاربر با گذرواژه وارد شود، دستور در فایل mysql-history ذخیره می‌شود و مهاجم با باز کردن این فایل می‌تواند به سادگی به گذرواژه‌ها دسترسی پیدا کند.

اطلاع از وضعیت فعلی:

برای یافتن این فایل می‌توان از دستور زیر استفاده کرد:

```
find / -name *mysql_history*
```

مقاوم‌سازی:

```
rm /root/.mysql_history
ln -s /dev/null /root/.mysql_history
```

با استفاده از این دستور این فایل پاک می‌شود و از این پس چیزی درون فایلی با این نام نگهداری نمی‌شود.

۲-۱۰ جمع‌بندی

در این فصل به تشریح برخی از مهم‌ترین پارامترهای امنیتی مربوط به پیکربندی سمپاد قبل از بکارگیری عملیاتی آن پرداختیم. در این راستا، تنظیمات مربوط به عدم بارگذاری توابع تعریف شده توسط کاربران در کتابخانه‌های اشتراکی، اعطای مجوز حذف/ ایجاد و تغییر نام کاربران، محدود کردن دسترسی به فایل‌های حساس، غیر فعال کردن مجوزدهی در سامانه، بارگذاری امن فایل‌ها بر روی سامانه، عدم نمایش اسامی تمامی پایگاه‌داده‌های موجود در سامانه برای تمامی کاربران، جلوگیری از ایجاد پیوند برای جداول، حذف پایگاه‌داده تست و حذف فایل‌های لاگ مربوط به روال نصب مورد بحث و بررسی قرار گرفت. مدیر سامانه به منظور بررسی پارامترهای مقاوم‌سازی و تهیه گزارش در این زمینه می‌تواند از چک لیست زیر استفاده نماید.

عنوان		تنظیم صحیح	
		بله	خیر
۳	**پیکره‌بندی امن پایگاه‌داده**		
۳-۱	پارامتر allow-suspicious-udfs	☐	☐
۳-۲	پارامتر create_user_priv	☐	☐
۳-۳	پارامتر secure-file-priv	☐	☐
۳-۴	پارامتر skip-grant-tables	☐	☐
۳-۵	پارامتر local_infile	☐	☐
۳-۶	پارامتر skip_show_database	☐	☐
۳-۷	پارامتر skip-symbolic-links	☐	☐
۳-۸	حذف پایگاه داده تست	☐	☐
۳-۹	حذف فایل تاریخچه mysql_history	☐	☐

۳ امن سازی اتصال به پایگاه‌داده

تنظیمات پیش‌فرض سمپاد، همواره یکی از مسائل بغرنجی است که بستر مناسبی را برای تهدیدهای امنیتی در سامانه پدید می‌آورد. از این رو تغییر آن‌ها به مقادیر امن یکی از نکات ضروری در ایجاد امنیت است. در یک دسته‌بندی کلی، تنظیمات مربوط به امن‌سازی اتصال به پایگاه‌داده را می‌توان در سه رده: امن‌سازی کانال ارتباطی، امن‌سازی احراز اصالت و امن سازی کنترل دسترسی تقسیم‌بندی نمود.

از آنجاییکه تنظیمات مربوط به احراز اصالت و کنترل دسترسی از جایگاه ویژه‌ای در امن‌سازی اتصال به پایگاه‌داده برخوردار هستند و تنظیمات آنها به خط‌مشی‌های داخلی سازمان استفاده کننده از سمپاد وابسته است، لذا تشریح آنها در قالب رهنمون‌هایی به منظور تنظیمات امن در پیوست آورده شده است. در ادامه برخی از مهمترین پارامترهای تاثیرگذار در امن‌سازی اتصال به پایگاه‌داده که بیشتر بر روی امن‌سازی کانال ارتباطی متمرکز هستند، آورده شده است.

۳-۱ پارامتر bind-address

امن‌سازی سمپاد، وابستگی زیادی به سطح انتظار ما از سمپاد دارد. یکی از انتظارات مهم در این زمینه، نیاز به دسترسی از راه دور به سمپاد است.

تهدید/توجیه امنیتی:

چنانچه نیاز باشد که دسترسی به سمپاد تنها به صورت محلی صورت پذیرد، می‌توان با غیرفعال کردن تماس‌هایTCP/IP، سطح امنیت را به طور قابل ملاحظه‌ای افزایش داد. برای این کار کافی است bind-address را به شبکه داخلی محدود کنیم.

اطلاع از وضعیت فعلی:

با اجرای دستور زیر، اطلاعات مربوط به این پارامتر نمایش داده می‌شود.

```
grep   bind-address          /etc/mysql/my.cnf
```

در صورتی که مقدار آن برابر با 127.0.0.1 باشد، بدین معنی است که تنها ارتباطات داخلی امکان‌پذیر هستند. مقدار پیش‌فرض این پارامتر نیز برابر با همین مقدار است.

مقاوم‌سازی:

اجرای دستور زیر سبب محدود شدن ارتباطات سمپاد می‌شود.

```
sed   -I   's/\(.*\)bind-address\(.*\)/bind-address = 127.0.0.1/' \  /etc/mysql/my.cnf
```

۳-۲ پارامتر max_connections

این پارامتر حداکثر اتصالات همزمان به سِرور را مشخص می‌کند. یکی از این اتصالات همیشه برای کاربر با مجوز SUPER رزرو شده است. مقدار پیش‌فرض این پارامتر ۱۰۰ است.

تهدید/توجیه امنیتی:

اگر مقدار این پارامتر کوچک باشد، به راحتی می‌توان حمله منع سرویس روی آن اعمال کرد. به عنوان مثال اگر مقدار این پارامتر ۱۰ باشد، یک مهاجم می‌تواند با برقراری ۱۰ ارتباط با سمپاد، سمپاد را مشغول کند و نگذارد با کاربران دیگر ارتباط برقرار کند و در نتیجه از سرویس‌دهی به کاربران مجاز جلوگیری می‌شود.

اطلاع از وضعیت فعلی:

با اجرای دستور زیر می‌توان از وضعیت فعلی این پارامتر در سامانه اطلاع پیدا کرد.

```
grep   max_connections      /etc/mysql/my.cnf
```

مقاوم‌سازی:

با اجرای دستور زیر، مقدار پارامتر به مقدار موردنظر تغییر می‌یابد. لازم است بجای <Appropriate value> مقدار مناسبی بسته به میزان بار معمول روی سمپاد قرار گیرد.

```
sed    -i  's/\(.*\)max_connections = 100\(.*\)/max_connections   \       = <Appropriate value>\2/'  my.cnf
```

۳-۳ محدود کردن منابع یک حساب کاربری

امکان محدود کردن منابع سمپاد MySQL به تفکیک حساب کاربری وجود دارد. این منابع عبارتند از:

- MAX_QUERIES_PER_HOUR: حداکثر تعداد پرس و جوها در هر ساعت
- MAX_UPDATES_PER_HOUR: حداکثر تعداد دستورات به روز رسانی در هر ساعت
- MAX_CONNECTIONS_PER_HOUR: حداکثر تعداد اتصالات در هر ساعت
- MAX_USER_CONNECTIONS: حداکثر تعداد اتصالات همزمان از طریق حساب خاص

تهدید/توجیه امنیتی:

تنظیم پارامتر max_connections فقط بر روی اتصالات همزمان، آن هم صرف نظر از کاربری خاص محدودیت می‌گذارد. در مواردی محدودیت‌های بیشتری در خصوص یک حساب کاربری خاص مورد نظر است.

اطلاع از وضعیت فعلی:

دستور اول، محدودیت‌های کاربر جاری را به همراه سایر مجوزهایش نمایش می‌دهد. دستور دوم همان کار را برای حساب کاربری مشخص انجام می‌دهد.

```
mysql> SHOW      GRANTS;
mysql> SHOW    GRANTS          FOR       '<user name>'@'<host>';
```

مقاوم‌سازی:

دستور زیر موجب تغییر محدودیت‌های کاربر user1@localhost می‌شود.

```
GRANT USAGE ON *.* TO 'user1'@'localhost'
  WITH MAX_QUERIES_PER_HOUR <value>
       MAX_UPDATES_PER_HOUR <value>
       MAX_CONNECTIONS_PER_HOUR <value>
       MAX_USER_CONNECTIONS <value>;
```

۳-۴ پارامتر max_connect_errors

این پارامتر حداکثر تلاش ناموفق برای اتصال به سِرور را مشخص می‌کند.

تهدید/توجیه امنیتی:

این پارامتر می‌تواند از حمله آزمون جامع جلوگیری کند. این حمله به این صورت است که مهاجم برای یک کاربر مشخص، تعداد زیادی گذرواژه را امتحان می‌کند تا بتواند آن حساب کاربری را هک کند و از طریق آن وارد سیستم شود. در صورت محدود شدن این پارامتر، تعداد تلاش‌های مهاجم محدود و شانس موفقیت مهاجم کاهش می‌یابد.

اطلاع از وضعیت فعلی:

با اجرای دستور زیر می‌توان از وضعیت فعلی این پارامتر در سامانه آگاه شد.

```
mysql> SHOW global  VARIABLES like '%max_connect_error%';
```

مقاوم‌سازی:

با اجرای این دستور، مقدار پارامتر به مقدار موردنظر تغییر می‌یابد. لازم است بجای <Appropriate value> مقدار مناسبی بسته به حساسیت سمپاد قرار گیرد.

```
sed    -i  "s/\[mysqld\]/\[mysqld\]\nmax_connect_errors = <Appropriate value>/" /etc/mysql/my.cnf
```

۵-۳ پارامتر port

این پارامتر نشان دهنده شماره درگاهی است که سِرور بر روی آن پاسخ می‌دهد. مقدار پیش فرض آن 3306 است. این درگاه نباید از میزبان‌های غیر قابل اعتماد قابل دسترس باشد.

تهدید/توجیه امنیتی:

لازم است شماره درگاه دیگری استفاده شود تا توانایی مهاجمان در شناسایی مشخصات سمپاد و حمله به آن کاهش یابد.

اطلاع از وضعیت فعلی:

با بررسی خروجی دستور زیر می‌توان تعیین کرد که سِرور روی چه درگاهی اتصالات را می‌پذیرد.

```
grep  port          /etc/mysql/my.cnf
```

یک راه ساده به منظور بررسی باز بودن درگاه MySQL، استفاده از دستور زیر از یک ماشین راه دور است. توجه داشته باشید که server_host نام میزبان و یا IP میزبانی است که سمپاد MySQL روی آن اجرا می‌شود و port_num شماره درگاه مورد بررسی است:

```
telnet     server_host port_num
```

در صورتی که telnet به حالت تعلیق در آمده و یا ارتباط پذیرفته نشود، درگاه بسته است.

مقاوم‌سازی:

رویکرد اول در مقاوم‌سازی، تغییر شماره درگاه به مقداری غیر از مقدار پیش فرض است. با اجرای دستور زیر، شماره درگاه به مقدار مورد نظر تغییر می‌یابد. لازم است بجای <Appropriate value> مقدار مناسبی قرار گیرد.

```
sed    -i   's/\(.*\)port.*=\(.*\)/port = <Appropriate value>/'  \ /etc/mysql/my.cnf
```

رویکرد دوم، محدود کردن دسترسی به درگاه از طریق دیوار آتش و یا مسیریاب است. مگر اینکه دلیلی خاص برای دسترس‌پذیر بودن آن وجود داشته باشد.

OpenSSL ۳-۶

از SSL برای امن کردن یک ارتباط و جلوگیری از افشاء اطلاعات استفاده می‌شود. از آنجا که در صورت فعال بودن این پارامتر، ارتباط رمز شده است، حتی اگر مهاجمی بتواند داده‌های ارتباط را شنود کند قادر به رمزگشایی اطلاعات نخواهد بود.

اطلاع از وضعیت فعلی:

با بررسی خروجی دستور زیر می‌توان از مقدار این پارامتر مطلع شد. در صورتی که مقدار این پارامتر برابر با enabled باشد، OpenSSL فعال است.

```
mysql > SHOW VARIABLES LIKE '%openssl%';
```

تهدید، توجیه امنیتی:

حفاظت ارتباطات شبکه با سمپاد از اهمیت زیادی برخوردار است. در صورتی که ترافیک به صورت آشکار باشد، امکان شنود گذرواژه‌ها وجود دارد. می‌توان از ابزارهایی مانند tcpdump و strings برای بررسی رمزگذاری یا عدم رمزگذاری ترافیک شبکه استفاده کرد. در اغلب موارد، دستوری مشابه دستور زیر قابل استفاده است. البته در صورت مشاهده داده‌ها به صورت رمزی نمی‌توان استنباط کرد که اطلاعات حتما رمزگذاری می‌شود.

```
shell> tcpdump -l -i eth0 -w - src or dst port 3306 | strings
```

مقاوم‌سازی:

برای مقاوم‌سازی ابتدا می‌بایست گواهی‌های زیر ایجاد شوند:

- کلید و گواهی مرجع صدور گواهی
- کلید رمزنگاری و درخواست گواهی سِرور
- کلید رمزنگاری و درخواست گواهی کاربر

پس از آنکه فایل‌های زیر ایجاد شدند، باید در فایل my.cnf تغییرات مربوطه اعمال شوند. برای این کار کافی است ابتدا خطوط زیر را از حالت غیرفعال خارج کرده و سپس مقادیر مربوط به آدرس گواهی و کلیدها جایگذاری شوند.

```
ssl-ca=$DIR/cacert.pem
ssl-cert=$DIR/client-cert.pem
```

```
ssl-key=$DIR/client-key.pem
```

OpenSSH ۳-۷

یک رویکرد برای امن کردن یک ارتباط و جلوگیری از افشای اطلاعات است. در این روش با استفاده از تونل، ترافیک شبکه رمز می‌شود. از جمله مزایای این روش می‌توان به موارد زیر اشاره کرد:

- عدم نیاز به تغییر تنظیمات MySQL.
- عدم وجود سربار مدیریتی برای ایجاد و نگهداری کلیدها و گواهی‌ها.
- استفاده از تونل با ساختار MySQL سازگار است و نیازی نیست MySQL در برقراری تونل، عملیاتی انجام دهد.
- سادگی در ایجاد تونل.

علیرغم مزایای گفته‌شده، این روش معایبی نیز دارد، ازجمله:

- مکانیزم ایجاد تونل باید توسط کاربر انجام پذیرد که باعث ایجاد مدیریت غیرمتمرکز می‌شود.
- ماشین سِرور باید OpenSSH را همیشه در حال اجرا داشته باشد، این کار با آنکه کار ساده‌ای است اما اجرای همیشگی آن ضروری نیست. از طرف دیگر، کاربر نیز می‌بایست SSH را بر روی سیستم خود نصب‌کرده باشد.

اطلاع از وضعیت فعلی:

برای اطلاع از وجود openssh-client بر روی سیستم، می‌توان از دستور زیر استفاده کرد:

```
apt -q list openssh-clients
```

تهدید/ توجیه امنیتی:

توجیه امنیتی این پارامتر نیز شبیه به پارامتر openssl است با این تفاوت که ایجاد و نگهداری این ارتباط بسیار ساده‌تر از رویکرد استفاده از openssl است.

مقاوم‌سازی:

ابتدا باید از سمت کاربر دستور زیر اجرا شود:

```
ssh -f user@ssh.example.com -L 3307:mysql1.example.com:3306 -N
```

در این دستور، user نام کاربری SSH بر روی میزبان ssh.example.com است که سِرور SSH، بر روی آن نصب شده است. همچنین mysql1.example.com و 3306 به ترتیب نام میزبان و درگاهی است که MySQL بر روی آن نصب شده است. با اجرای این دستور، درگاه 3307 بر روی ماشین کاربر، آماده دریافت

ارتباط می‌شود و کلیه اتصالات TCP به این درگاه از طریق سِرور SSH (بـر روی ssh.example.com) منجـر به اتصال از آن سِرور به درگاه 3306 بر روی mysql1.example.com می‌شود. لذا نگرانی از امنیت ارتباط بین میزبان SSH و MySQL وجود نداشته است؛ یا در حالتی که دو میزبان یکسان مشخص شده باشـند، هـر دو خدمت بر روی یک میزبان نصب شده‌اند. ولی تمام ارتباطات بین کاربر و میزبان SSH رمـز می‌شـود. پـس از اجرای موفق این دستور، کاربر می‌تواند به صورت localhost بـا MySQL ارتبـاط برقـرار کنـد و ترافیـک تـا میزبان ssh.example.com رمز می‌شود.

```
mysql -h 127.0.0.1 -P 3307
```

۳-۸ تغییر نام کاربری و گذرواژه root

نام کاربری پیش‌فرض برای مدیر پایگاه داده که در هنگام نصب مشخص می‌شود، root است.

تهدید/توجیه امنیتی:

مهاجمان عموماً سعی بر دسترسی به root دارند. برای سخت شدن عملیات دسترسی مهاجمان به حساب این کاربر، بهتر است نام آن را از حالت پیش‌فرض تغییر داد. علاوه بر این توصیه می‌شود گذرواژه آن نیز بـه یـک گذرواژه طولانی و پیچیده تغییر یابد تا احتمال موفقیت مهاجم تا حد امکان کاهش یابد.

مقاوم‌سازی:

برای تغییر نام کاربری و تغییر گذر واژه به ترتیب از دو دستور زیر استفاده می‌شود.

```
mysql> RENAME USER root@localhost TO <new-name>@localhost;
mysql> SET PASSWORD FOR '<new-name>'@'localhost' = PASSWORD('<new-password>');
```

۳-۹ حذف حساب‌های کاربری بی‌نام

پایگاه داده MySQL به صورت پیش‌فرض دارای تعدادی کاربر بی‌نام است. در صورتی که وجود این حساب‌ها در سیستم لازم باشد و مدیر سیستم نخواهد آن‌ها را حذف کند، باید برای آن‌ها گذرواژه تعیین کند.

تهدید/توجیه امنیتی:

در صورت وجود چنین حساب‌هایی در پایگاه داده، هرکس می‌تواند بدون استفاده از گذر واژه، به پایگـاه داده متصل شود و اعمالی را که به تنهایی مجوز انجام آن‌ها را نداشته است، انجام دهد.

اطلاع از وضعیت فعلی:

برای اطمینان از وجود یا عدم وجود این گونه حساب‌های کاربری در پایگاه داده می‌توان از دستور زیر استفاده کرد.

```
mysql> SELECT host,user,password FROM mysql.user where user="";
```

در یک پایگاه داده امن، دستور فوق نباید خروجی همراه داشته باشد.

مقاوم‌سازی:

برای مقاوم‌سازی می‌توان حساب‌های کاربری بی‌نام را با استفاده از دستور زیر حذف کرد.

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

۳-۱۰ کاربران بدون گذرواژه

هنگام نصب MySQL، ممکن است کاربری بدون گذرواژه در پایگاه داده ثبت شود. علاوه بر این، گاهی ممکن است هنگام اضافه کردن یک کاربر به پایگاه داده تعیین گذرواژه مربوط به آن فراموش شود.

تهدید/توجیه امنیتی:

در صورتی که در MySQL حساب کاربری وجود داشته باشد که برای آن گذر واژه‌ای تعیین نشده باشد، مهاجم می‌تواند با استفاده از آن حساب به سِرور متصل شود، درحالی‌که در حالت عادی مجوز اتصال را نداشته است.

اطلاع از وضعیت فعلی:

برای مطلع شدن از وجود این نوع کاربران می‌توان از دستور زیر استفاده کرد.

```
mysql> SELECT host,user,password FROM mysql.user where password="";
```

مقاوم‌سازی:

برای مقاوم‌سازی لازم است برای کاربران بدون‌گذر واژه، گذر واژه مناسبی را تعیین نمود.

```
SET PASSWORD FOR '<user>'@'localhost' = PASSWORD('<cleartext password>');
```

۳-۱۱ جمع‌بندی

در این فصل به تشریح برخی از مهم‌ترین پارامترهای امنیتی مربوط به اتصال امن به سمپاد پرداختیم. در این راستا، تنظیمات مربوط به دسترسی از راه دور سمپاد، حداکثر اتصالات همزمان به سمپاد، محدود کردن منابع یک حساب کاربری، حداکثر تلاش ناموفق برای اتصال به سمپاد، شماره درگاه مربوط به شنود یا پاسخگویی سمپاد، امن کردن ارتباط با استفاده از Openssl، امن کردن ارتباط با استفاده از Openssh، تغییر نام کاربری و گذر واژه root، حذف حساب‌های کاربری بی‌نام و تعیین گذر واژه برای کاربران بدون گذر واژه مورد بحث و بررسی قرار گرفت. مدیر سامانه به منظور بررسی پارامترهای مقاوم‌سازی و تهیه گزارش در این زمینه می‌تواند از چک لیست زیر استفاده نماید.

عنوان		تنظیم صحیح	
		بله	خیر
۴	**امن‌سازی اتصال به پایگاه‌داده**		
۱-۴	پارامتر bind-address	☐	☐
۲-۴	پارامتر max_connections	☐	☐
۳-۴	محدود کردن منابع یک حساب کاربری	☐	☐
۴-۴	پارامتر max_connect_errors	☐	☐
۵-۴	پارامتر port	☐	☐
۶-۴	OpenSSL	☐	☐
۷-۴	OpenSSH	☐	☐
۸-۴	تغییر نام‌کاربری و گذرواژه root	☐	☐
۹-۴	حذف حساب‌های کاربری بی‌نام	☐	☐
۱۰-۴	کاربران بدون گذر واژه	☐	☐

۴ تنظیمات رویدادنگاری

رویدادنگاری سیستم و بازبینی آن‌ها در صورت رخداد مشکلات فنی یا امنیتی یکی از نیازمندی‌های اصلی در پایگاه داده‌ها است. با توجه به اینکه انواع وقایعی که در سیستم رخ می‌دهند از درجات اهمیت متفاوتی برخوردارند و ثبت کلیه رویدادها (بدون توجه به ارزش هر یک) می‌تواند منجر به کاهش کارایی و یا هدر رفتن فضای دیسک گردد، لازم است یک زیر مجموعه از وقایع مهم شناسایی گردد و رویداد نگاری در مورد آنها همواره انجام شود.

در مورد رویداد نگاری سایر وقایع، بسته به توانمندی‌های پردازشی و ذخیره‌سازی سمپاد باید تصمیم گرفته شود. در صورتی که امکان رویدادنگاری در سیستم خاموش باشد، فعالیت‌های بدخواهانه قابل شناسایی نخواهند بود. به صورت پیش‌فرض هیچ‌یک از امکانات رویدادنگاری در MySQL فعال نیست.

۴-۱ پارامتر general-log

برای رویدادنگاری از تمامی درخواست‌های انجام‌شده روی پایگاه داده در زمان اجرا، می‌توان پارامتر general-log را فعال کرد. این پارامتر می‌تواند مقدار صفر یا یک داشته باشد. برای تغییر مکانی که رویدادها در آنجا ثبت می‌شود، باید پارامتر دیگری به نام general-log-file را تغییر داد و در آن آدرس فایل مورد نظر را تعیین کرد.

تهدید/توجیه امنیتی:

در صورت غیرفعال بودن این گزینه، رویدادها و درخواست‌ها ثبت نمی‌شوند، درنتیجه کشف رفتارهای بدخواهانه میسر نخواهد بود.

اطلاع از وضعیت فعلی:

برای اطلاع از وضعیت فعلی این پارامتر از دستور زیر بهره می‌گیریم.

```
mysql> SHOW global  VARIABLES like 'general_log';
```

مقاوم‌سازی:

اجرای دستور فوق باعث فعال شدن امکان رویدادنگاری عمومی شده و رویدادها در فایل تعیین شده ذخیره خواهند شد.

```
sed -i 's/\(.*\)general_log =\(.*\)/general_log = 1/'  /etc/mysql/my.cnf
sed -i 's/\(.*\)general_log_file\(.*\)/general_log_file = \/var\/log\/mysql\/mysql.log /' /etc/mysql/my.cnf
```

۴-۲ پارامتر slow_query_log

این پارامتر مربوط به درخواست‌هایی می‌شود که مدت پاسخگویی به آن‌ها بیش از زمان long_query_time است. مقدار پیش‌فرض long_query_time در ابتدا برابر با 10 ثانیه است. به هنگام ثبت این‌گونه درخواست‌ها، زمان مربوط به آن‌ها نیز به میلی‌ثانیه ثبت می‌شود. به‌صورت پیش‌فرض، درخواست‌های مربوط به مدیریت در این فایل ذخیره نمی‌شوند و مقدار آن برابر با صفر است.

تهدید/توجیه امنیتی:

یکی از کاربردهای استفاده از این فایل، یافتن دستوراتی است که بیش از حد طول می‌کشند و کارایی سیستم را کاهش می‌دهند. به عنوان مثال زمانی که بار روی سِرور زیاد است، می‌توان بر اساس این فایل تعدادی از درخواست‌ها را ملغی کرد. از طرف دیگر برخی از حملات به پایگاه داده، حملاتی هستند که مدت زمان زیادی برای پاسخ به آن‌ها لازم است. پس از این طریق می‌توان برخی از حملات را نیز شناسایی کرد. حمله‌ای مانند sleep می‌تواند در این گروه قرار گیرد. نمونه‌ای از این حمله در زیر نشان داده‌شده است.

```
mysql> SELECT SLEEP(11);
```

خروجی که در فایل نوشته می‌شود به صورت زیر است.

```
# Time: 141117 16:00:44
# User@Host: root[root] @ localhost []
# Query_time: 11.007855  Lock_time: 0.000000 Rows_sent: 1  Rows_examined: 0
SET timestamp=1416227444;
SELECT SLEEP(11);
```

اطلاع از وضعیت فعلی:

برای اطلاع از وضعیت فعلی این پارامتر از دستور زیر بهره می‌گیریم.

```
mysql> SHOW global  VARIABLES like 'slow_query_log';
```

همچنین برای اطلاع از وضعیت فعلی long_query_time می‌توان از دستور زیر استفاده نمود.

```
mysql> SHOW global  VARIABLES like 'long_query_time';
```

مقاوم‌سازی:

با تغییر slow_query_log، مقدار log_slow_query نیز تغییر می‌کند. درواقع log_slow_query نامی قدیمی برای این پارامتر است که منسوخ شده است و از آن استفاده نمی‌شود و مقدارش نیز با مقدار slow_query_log برابر است.

```
sed -i 's/\(.*\)log_slow_queries\(.*\)/log_slow_queries = \/var\/log\/mysql\/mysql-slow.log/' /etc/mysql/my.cnf
```

به این صورت می‌توان این پارامتر را فعال و آدرس ذخیره رویدادها را به آدرس مورد نظر خود تغییر داد.

۴-۳ پارامتر log_error

رویدادهای ثبت شده در این فایل مربوط به زمانی می‌شوند که mysqld فعال یا غیرفعال شده باشد و یا خطایی حین اجرای سِرور رخ داده باشد. همچنین در صورتی که mysqld تشخیص دهد جدولی نیاز به چک یا تغییر خودکار دارد، گزارش عملیاتش را در این فایل می‌نویسد. در بسیاری از سیستم‌عامل‌ها، در صورتی که سرویس mysql خاتمه داده شود، در این فایل رویدادنگاری می‌شود و بر اساس آن می‌توان فهمید کجا این اتفاق افتاده است. اگر در سیستمی این گزینه فعال نباشد، تمامی پیام‌های مربوط به آن روی کنسول نمایش داده می‌شود.

تهدید/توجیه امنیتی:

در صورت غیرفعال بودن این گزینه، رویدادها و درخواست‌ها تنها در کنسول نمایش داده می‌شوند و نمی‌توان از آن‌ها تاریخچه‌ای نگهداری کرد.

اطلاع از وضعیت فعلی:

برای اطلاع از وضعیت فعلی این پارامتر از دستور زیر بهره می‌گیریم.

```
mysql> SHOW global  VARIABLES like 'log_error';
```

مقاوم‌سازی:

برای مقاوم‌سازی باید log-error را به فایل my.cnf و در زیر [mysqld] و [mysqld_safe] اضافه کرد.

```
sed -i 's/\[mysqld\]/\[mysqld\]\nlog-error = /var/log/mysql/error.log /' /etc/mysql/my.cnf

sed -i s/\[mysqld_safe\]/\[mysqld_safe\]\nlog-error = /var/log/mysql/error.log/' /etc/mysql/my.cnf
```

۴-۴ پارامتر log_bin

این فایل حاوی اطلاعاتی راجع به اتفاقاتی است که باعث ایجاد تغییری در جداول یا داده‌های موجود در پایگاه داده می‌شود. این رخدادها حتی اگر باعث تغییر داده‌ای نشوند نیز ثبت می‌شوند؛ به عنوان مثال اگر دستور DELETE روی جدولی ارسال شود اما چنین جدولی وجود خارجی نداشته باشد، باز هم این اتفاق ثبت می‌شود. در این فایل رخدادهای مربوط به SELECT و SHOW ثبت نمی‌شوند، زیرا تغییری در پایگاه داده ایجاد نمی‌کنند. استفاده اصلی این فایل‌ها برای زمانی است که داده‌های پایگاه داده در سِرور دیگری نیز

کپی شده باشند. در این صورت هرزمان تغییری روی داده‌های سِرور اصلی ایجاد شود، تغییرات به سِرور دیگر نیز ارسال می‌شوند. استفاده دیگر این فایل‌ها در عملیات پشتیبان گیری است. برای بازگردانی اطلاعات، لازم است فایل‌های باینری نیز موجود باشند.

تهدید/توجیه امنیتی:

در صورت غیرفعال بودن این گزینه، امکان بازگردانی اطلاعات و به‌روز رسانی نسخه پشتیبان وجود ندارد. همچنین در صورتی که مهاجم باعث خرابی در پایگاه داده شده باشد، نمی‌توان اطلاعات را بدون از بین رفتن قسمتی از آن بازگردانی کرد.

اطلاع از وضعیت فعلی:

برای اطلاع از وضعیت فعلی این پارامتر از دستور زیر بهره می‌گیریم.

```
mysql> SHOW global  VARIABLES like 'log_bin';
```

مقاوم‌سازی:

برای مقاوم‌سازی لازم است این امکان را فعال و آدرس مناسبی برای ذخیره رویدادها مشخص کرد.

```
sed -i 's/\(.*\)log_bin\(.*\)/log_bin = \/var\/log\/mysql\/mysql-bin.log/'  /etc/mysql/my.cnf
```

۵ – ۴ جمع‌بندی

در این فصل به تشریح برخی از مهم‌ترین پارامترهای امنیتی مربوط به تنظیمات رویدادنگاری پرداختیم. در این راستا، تنظیمات مربوط به رویدادنگاری از تمامی درخواست‌های رسیده حین اجرای سمپاد، رویدادنگاری مربوط به درخواست‌هایی که زمان پاسخگویی به آنها بیش از حد تعیین شده است، رویدادنگاری مربوط به فعال یا غیر فعال بودن سمپاد یا ایجاد اختلال حین اجرای آن و رویدادنگاری مربوط به تغییرات در جداول یا داده‌های موجود در پایگاه‌داده مورد بحث و بررسی قرار گرفت. مدیر سامانه به منظور بررسی پارامترهای مقاوم‌سازی و تهیه گزارش در این زمینه می‌تواند از چک لیست زیر استفاده نماید.

	عنوان	تنظیم صحیح	
		بله	خیر
۵	**تنظیمات رویدادنگاری**		
۵-۱	پارامتر general-log	☐	☐
۵-۲	پارامتر slow_query_log	☐	☐
۵-۳	پارامتر log_error	☐	☐

۵-۴	پارامتر log_bin	☐	☐

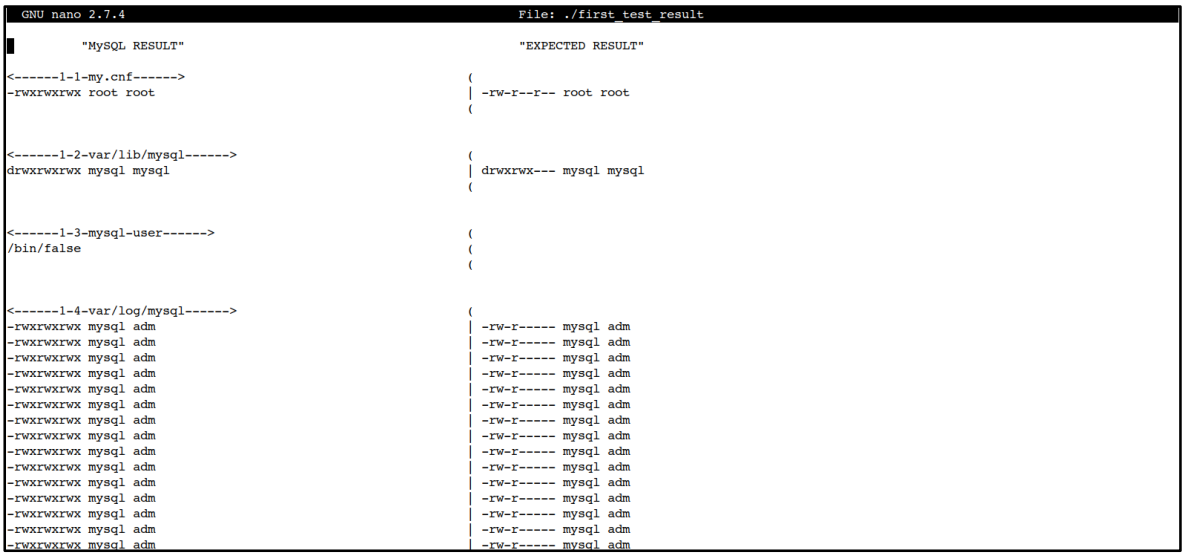
۵ راهنمای ابزار مقاوم‌سازی

این ابزار دارای سه فایل اجرایی است که در ادامه به بررسی هر یک از آنها می‌پردازیم.

۵-۱ فایل start.sh

این فایل اسکریپت، تنها فایلی است که کاربر باید آن را اجرا کند. بقیه اسکریپتها از طریق این فایل اسکریپت فراخوانی و اجرا می‌شوند.

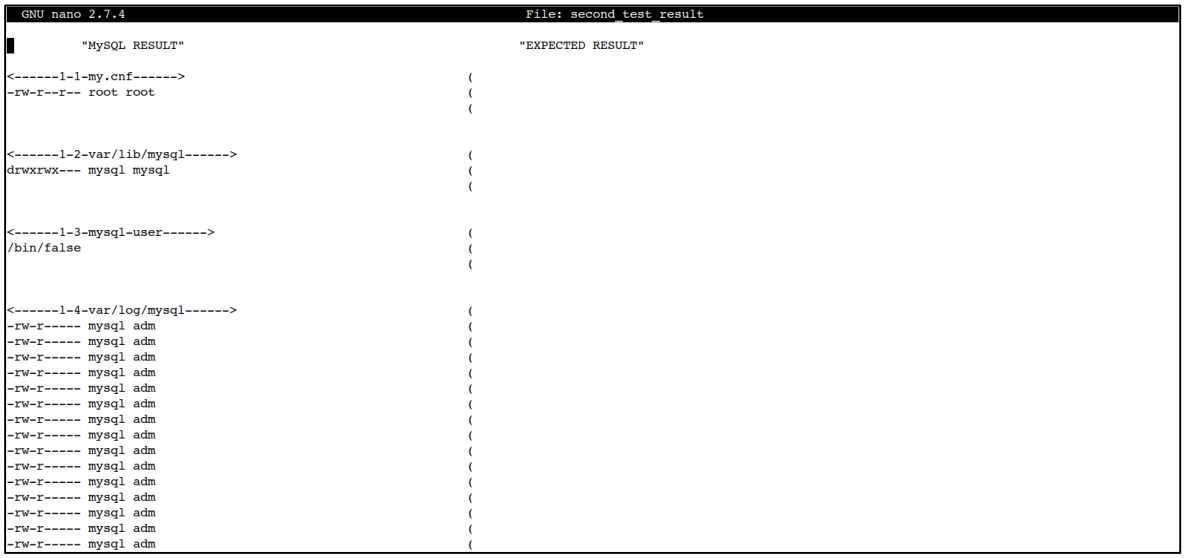
برای آنکه فرآیند تست سمپاد و امن‌سازی آن صورت گیرد، ابتدا لازم است از وجود MySQL روی سیستم اطمینان حاصل کرد. در صورتی که سمپاد MySQL بر روی سیستم نصب باشد و فایل‌های مربوط به تنظیمات موجود باشند، اسکریپت script.sh اجرا می‌شود. در اثر اجرای این فایل، دو پوشه حاوی نتایج آزمایش و نتایج مورد انتظار ایجاد می‌شوند. حال در این اسکریپت قصد داریم این دو پوشه را با هم مقایسه کنیم تا مغایرت‌های سیستم با موارد امنیتی مورد انتظار مشخص شوند. نتایج این آزمایش در فایل first_test_result ثبت می‌شود. نمونه ای از خروجی این برنامه در شکل (۱) نشان داده شده است.



```
 GNU nano 2.7.4                                File: ./first_test_result

          "MySQL RESULT"                              "EXPECTED RESULT"

<------1-1-my.cnf------>                          (
-rwxrwxrwx root root                               | -rw-r--r-- root root
                                                   (


<------1-2-var/lib/mysql------>                    (
drwxrwxrwx mysql mysql                             | drwxrwx--- mysql mysql
                                                   (


<------1-3-mysql-user------>                       (
/bin/false                                         (
                                                   (


<------1-4-var/log/mysql------>                    (
-rwxrwxrwx mysql adm                               | -rw-r----- mysql adm
-rwxrwxrwx mysql adm                               | -rw-r----- mysql adm
-rwxrwxrwx mysql adm                               | -rw-r----- mysql adm
-rwxrwxrwx mysql adm                               | -rw-r----- mysql adm
-rwxrwxrwx mysql adm                               | -rw-r----- mysql adm
-rwxrwxrwx mysql adm                               | -rw-r----- mysql adm
-rwxrwxrwx mysql adm                               | -rw-r----- mysql adm
-rwxrwxrwx mysql adm                               | -rw-r----- mysql adm
-rwxrwxrwx mysql adm                               | -rw-r----- mysql adm
-rwxrwxrwx mysql adm                               | -rw-r----- mysql adm
-rwxrwxrwx mysql adm                               | -rw-r----- mysql adm
-rwxrwxrwx mysql adm                               | -rw-r----- mysql adm
-rwxrwxrwx mysql adm                               | -rw-r----- mysql adm
-rwxrwxrwx mysql adm                               | -rw-r----- mysql adm
```

شکل ۱: محتوای فایل first_test_result

ستون سمت چپ نشان دهنده تنظیمات فعلی است. مواردی که با تنظیمات مورد انتظار مغایرت دارند در مقابل آنها نتیجه مورد انتظار آورده شده است. پس از اجرای این اسکریپت، در صورت تمایل کاربر، اسکریپت repair اجرا می‌شود. سپس هر مورد امنیتی که در آن نتیجه مورد انتظار و نتیجه حاصل از تست مغایر باشند، با موافقت کاربر امن سازی شده و در آخر نیز تست مجددی بر روی سیستم انجام می‌شود. نتیجه تست دوم در فایل second_test_result ذخیره خواهد شد.



شکل ۲: محتوای فایل second_file_result

۵-۲ فایل script.sh

در این فایل دو متغیر با نام‌های result_path و expected_path تعریف شده است. این دو متغیر به پوشه‌هایی اشاره می‌کنند که در آن‌ها به ترتیب نتایج هر آزمایش و نتیجه مورد انتظار آن آزمایش، ساخته می‌شود. در اینجا لازم است بنا به نیازمندی سیستم و نکات گفته شده حین توضیح هر مورد، نتایج مورد انتظار برای هر مورد امنیتی تنظیم شود. این کد توسط برنامه start.sh اجرا می‌شود.

۵-۳ فایل repair.sh

فایل آخر، مربوط به تغییر تنظیمات سیستم می‌شود. در صورتی که تنظیمات به درستی انجام شده باشد انتظار می‌رود که مورد امنیتی در ستون دوم وجود نداشته باشد و تنها چند پیشنهاد برای امنیت بیشتر در آن باقی بماند. در شکل (۲) می‌توان نمونه ای از فایل second_test-result را پس از اعمال تغییرات مشاهده کرد.

۶ جمع‌بندی

در این مستند به بررسی موارد امنیتی مربوط به مقاوم‌سازی سمپاد MySQL پرداخته شد. تنظیمات مربوط به مقاوم‌سازی این سمپاد در چهار بخش مختلف دسته بندی گردید. در بخش اول، امن‌سازی محیط اجرا، بخش دوم نصب و پیکربندی امن پایگاه‌داده، بخش سوم امن‌سازی اتصال به پایگاه‌داده و بخش چهارم تنظیمات رویداد نگاری مورد بحث و بررسی قرار گرفت. در مورد هر پارامتر، کاربرد، ارزش امنیتی و نحوه آگاهی از مقدار کنونی آن پارامتر و چگونگی مقداردهی امن آن توضیحاتی داده شد. در پایان نیز نحوه اجرای ابزار (اسکریپت‌های) مقاوم‌سازی و خروجی‌های حاصل از آنها بیان شدند. خلاصه‌ای از گزارش ارایه شده، به صورت یک چک لیست در ادامه آورده شده است.

	عنوان	تنظیم صحیح	
		بله	خیر
۲	**امن سازی محیط اجرا**		
۲-۱	پیکربندی فایل تنظیمات	☐	☐
۲-۲	پیکربندی دایرکتوری ذخیره داده	☐	☐
۲-۳	امن‌سازی کاربر MySQL	☐	☐
۲-۴	راه اندازی سرویس MySQL توسط یک کاربر عادی	☐	☐
۲-۵	پیکربندی فایل رویدادنگاری	☐	☐
۳	**پیکربندی امن پایگاه‌داده**		
۳-۱	پارامتر allow-suspicious-udfs	☐	☐
۳-۲	پارامتر create_user_priv	☐	☐
۳-۳	پارامتر secure-file-priv	☐	☐
۳-۴	پارامتر skip-grant-tables	☐	☐
۳-۵	پارامتر local_infile	☐	☐
۳-۶	پارامتر skip_show_database	☐	☐
۳-۷	پارامتر skip-symbolic-links	☐	☐

۳-۸	حذف پایگاه داده تست	☐	☐
۳-۹	تنظیمات فایل تاریخچه	☐	☐
۴	**امن‌سازی اتصال به پایگاه‌داده**		
۴-۱	پارامتر bind-address	☐	☐
۴-۲	پارامتر max_connections	☐	☐
۴-۳	محدود کردن منابع یک حساب کاربری	☐	☐
۴-۴	پارامتر max_connect_errors	☐	☐
۴-۵	پارامتر port	☐	☐
۴-۶	OpenSSL	☐	☐
۴-۷	OpenSSH	☐	☐
۴-۸	تغییر نام‌کاربری و گذرواژه root	☐	☐
۴-۹	حذف حساب‌های کاربری بی‌نام	☐	☐
۴-۱۰	کاربران بدون گذر واژه	☐	☐
۵	**تنظیمات رویدادنگاری**		
۵-۱	پارامتر general-log	☐	☐
۵-۲	پارامتر slow_query_log	☐	☐
۵-۳	پارامتر log_error	☐	☐
۵-۴	پارامتر log_bin	☐	☐

۷ مراجع

[1] R. B. Natan, Implementing Database Security and Auditing, Elsevier.

[2] MySQL Reference Manual. URL: http://dev.mysql.com/doc.

[3] Security Configuration Benchmark for MySQL, The Center for Internet Security (CIS), URL: http://cisecurity.org.

[4] OWASP Backend Security Project MySQL Hardening, The Open Web Application Security Project, URL: https://www.owasp.org.

۸ پیوست

۸-۱ رهنمون‌های امنیتی پس از نصب

از آنجا که هنگام نصب MySQL به موارد امنیتی توجه ویژه‌ای نمی‌شود، می‌بایست پس از نصب رهنمون‌های ویژه‌ای را جهت حداقل کردن ریسک‌های امنیتی، مد نظر قرار داد. از این‌رو در این بخش، اجـرای تمهیـدات امنیتی اولیه، امن‌سازی پایگاه‌داده test، تنظیمات جداول اعطا، اجرا/توقف خودکار سمپاد و راه‌اندازی سـمپاد در MySQL توسط یک کاربر عادی مورد بحث و بررسی قرار می‌گیرد.

اجرای تمهیدات امنیتی اولیه:

از آنجا که هنگام نصب MySQL به موارد امنیتی توجه چندانی نمی‌شود، MySQL امکانـات اولیـه‌ای ایجـاد کرده است که با استفاده از آن می‌توان برخی از حفره‌های امنیتی را برطرف نمود. برای این کار کـافی اسـت دستور mysql_secure_installation را در محیط shell اجرا کنیم. پس از اجـرای ایـن دسـتور، مـوارد زیـر اصلاح می‌شوند:

۱) تغییر گذرواژه root
۲) حذف کاربران بی‌نام
۳) غیرفعال کردن امکان اتصال از راه دور root به سیستم
۴) حذف پایگاه داده test و دسترسی به آن
۵) به‌روز رسانی جدول مجوزها

که شرح هر یک از موارد فوق، در بخش‌های مختلف سند پیش‌رو موجود است.

امن‌سازی پایگاه‌داده تست:

به صورت پیش فرض، جدول mysql.db شامل سطرهایی است که امکان دسترسی تمامی کاربران بـه پایگـاه داده test و پایگاه‌های داده‌ی دیگری که با نام test_ آغاز می‌شوند، را فراهم می‌آورد. در حقیقت، فیلد USER در این سطرها مقداردهی نشده و در نتیجه با تمامی نام‌های کاربری تطابق دارد. این امر بدین معنی است که این پایگاه‌های داده در معرض دسترسی حساب‌هایی است که حتی دارای مجوز خاصـی نیسـتند. بـه منظـور محدودسازی دسترسی کاربران خاص به اینگونه پایگاه‌های داده، دستورات زیر را دنبال کنید.

```
mysql> DELETE FROM mysql.db WHERE Db LIKE 'test%';
mysql> FLUSH PRIVILEGES;
```

با استفاده از این تغییر، تنها کاربرانی که دارای مجوزهای عمومی یا مجوزهایی خاص برای پایگاه داده test هستند، می‌توانند از آن استفاده نمایند. علاوه بر این، در صورت عدم نیاز می‌توان این پایگاه داده را حذف نمود.

تنظیمات جدول اعطا:

پس از نصب MySQL در لینوکس می‌بایست جداول اعطا مقداردهی شوند، سِرور روشن شود و نحوه عملکرد آن مورد بررسی قرار گیرد. علاوه بر این می‌توان سِرور را به صورتی پیکربندی کرد که با روشن و خاموش شدن سیستم روشن و خاموش شود. علاوه بر این باید به حسابهای موجود در جداول اعطا گذرواژهایی را اختصاص داد.

در لینوکس، تنظیم جداول اعطا با استفاده از اسکریپت mysql_install_db انجام می‌شود. در برخی روش‌های نصب، در صورت عدم وجود یک پایگاه داده، این اسکریپت به صورت خودکار اجرا می‌شود. در سیستمهای عامل و انواع دیگر نصب باید به صورت دستی اجرا شود. رویه زیر چگونگی مقداردهی اولیه جداول اعطا و آغاز به کار سِرور را بیان می‌کند. علاوه بر این، دستوراتی نیز به منظور بررسی امکان دسترسی به سِرور و نیز نحوه عملکرد آن ارائه می‌شود. پس از اجرای کامل این گامها و آغاز به کار سِرور باید به حسابهای ایجاد شده توسط mysql_install_db گذرواژه‌هایی اختصاص داد و دسترسی به پایگاه‌های داده تست را محدود کرد. گام‌های مقدار دهی اولیه جداول اعطا و نحوه آغاز به کار سِرور به صورت زیر است:

۱) مسیر جاری را به بالاترین سطح دایرکتوری نصب MySQL ، که با نام BASEDIR نمایش داده می‌شود، تغییر دهید.

```
shell> cd BASEDIR
```

در حقیقت، BASEDIR دایرکتوری نصب MySQL است. این دایرکتوری می‌تواند چیزی شبیه /usr/local/mysql/ یا /usr/local باشد. فایلها و دایرکتوری‌های مختلفی در این دایرکتوری وجود ندارد. دایرکتوری‌های حائز اهمیت دراین میان bin و scripts هستند.

- دایرکتوری bin شامل برنامه‌های کاربر و سِرور است. آدرس کامل این دایرکتوری را به متغیر محلی PATH اضافه نمایید تا ارتباط با برنامه‌های MySQL به درستی امکانپذیر باشد. توجه داشته باشید که mysqld در برخی نسخه‌ها در دایرکتوری libexec نصب می‌شود.
- دایرکتوری scripts شامل اسکریپت mysql_install_db است که به منظور مقداردهی اولیه پایگاه داده mysql به کار می‌رود. در برخی نسخه‌ها این اسکریپت در دایرکتوری bin نصب می‌شود.

۲) فرض کنید که سِرور با استفاده از حساب کاربری mysql اجرا می‌شود. در صورت نیاز باید محتوای نصب در دسترس mysql قرار گیرد. در صورتی که نصب با استفاده از mysql انجام شده باشد، نیاز به انجام کار دیگری نیست. در صورتی که نصب با استفاده از root انجام شده باشد، محتوای آن در مالکیت root خواهد بود. تغییر مالکیت به mysql با استفاده از دستورات زیر در دایرکتوری نصب امکانپذیر است. دستور اول مالکیت فایل‌ها را در اختیار کاربر mysql قرار می‌دهد. دستور دوم مشخصه گروه را به گروه mysql تغییر می‌دهد.

```
shell> chown -R mysql
shell> chgrp -R mysql
```

۳) در صورت نیاز اسکریپت mysql_install_db را به منظور مقداردهی اولیه جداول اعطای MySQL اجرا کنید. به طور معمول، اجرای این اسکریپت تنها در اولین نصب MySQL لازم است. در نتیجه، در به روزرسانی‌های بعدی می‌توان از آن صرف نظر کرد. باید توجه داشت که این برنامه جداول اعطا را بازنویسی نکرده و استفاده از آن در تمامی شرایط امن است. مکان دقیق این اسکریپت به نحوه نصب بستگی دارد. به عنوان مثال در صورتی که این اسکریپت در دایرکتوری bin باشد، دستور زیر را باید اجرا کرد:

```
shell> bin/mysql_install_db --user=mysql
```

در صورتی که اسکریپت، موقعیت دایرکتوری‌های نصب یا داده را تشخیص ندهد، استفاده از گزینه‌هایی مانند basedir-- یا datadir-- ضروری است.

```
shell> scripts/mysql_install_db --user=mysql \
--basedir=/opt/mysql/mysql  --datadir=/opt/mysql/mysql/data
```

این اسکریپت دایرکتوری داده سِرور را با مالکیت mysql ایجاد می‌نماید. در دایرکتوری داده، زیر دایرکتوری‌هایی برای پایگاه داده mysql و نیز پایگاه داده test ایجاد می‌شود. این اسکریپت سطرهای کنترلی را برای root و دیگر حسابهای کاربری در جدول مجوزها ایجاد می‌نماید. توجه داشته باشید که این حسابها در ابتدا فاقد گذرواژه هستند. لازم به ذکر است که این مجوزهای اولیه به کاربر root اجازه انجام تمامی فعالیتها را اعطا کرده و مابقی کاربران امکان ایجاد و یا استفاده از پایگاه دادهای با نام test را دارا هستند.

نکته حائز اهمیت در اینجا آن است که دایرکتوری‌ها و فایل‌های پایگاه داده می‌بایست تحت مالکیت mysql باشند. در این صورت، سِرور به آنها دسترسی خواندن و نوشتن خواهد داشت. به منظور اطمینان، در صورتی که اسکریپت را با استفاده از کاربر root اجرا می‌کنید، از گزینه user-- استفاده نمایید. در غیر اینصورت باید با حساب کاربر mysql وارد شده باشید.

۴) در اکثر موارد، نصب MySQL می‌تواند با مالکیت root انجام شود. یکی از موارد استثناء دایرکتوری داده است که باید در مالکیت کاربر mysql باشد. به منظور تغییر مالکیت، دستورات زیر را در به عنوان کاربر root در دایرکتوری نصب اجرا نمایید.

```
shell> chown -R root
shell> chown -R mysql data
```

۵) اگر دایرکتوری مشخص شده با متغیر سیستمی plugin_dir برای سِرور قابل نوشتن باشد، یک کاربر می‌تواند با استفاده از SELECT … INTO DUMPFILE کدهای اجرایی را در یک فایل در این دایرکتوری بنویسد. برای پیشگیری از این امر می‌توان دایرکتوری مذکور را برای سِرور فقط-خواندنی کرد.

۶) در صورتی که MySQL با استفاده از فایل‌های منبع نصب شده باشد، می‌توان به صورت دلخواه یکی از فایل‌های پیکربندی را از دایرکتوری support-files به دایرکتوری /etc منتقل کرد. فایل‌های پیکربندی مختلفی برای کاربردهای متفاوت، انواع سِرور و تنظیمات سخت افزاری متفاوت وجود دارد. به منظور استفاده از فایل‌های پیکربندی باید آنها را در /etc/my.cnf یا /etc/mysql/my.cnf کپی کرد. باید توجه داشت در صورتی که از فایل‌های پیکربندی پیش فرض استفاده نمی‌شود، سِرور MySQL با تنظیمات پیش فرض اجرا می‌شود. به منظور اجرای خودکار MySQL با آغاز به کار سیستم، می‌توان support-files/mysql.server را در مکانی که فایل‌های آغاز به کار سیستم وجود دارد، کپی کرد.

۷) سِرور MySQL را اجرا کنید:

```
shell> bin/mysqld_safe --user=mysql &
```

نکته حائز اهمیت در اینجا اجرای سِرور MySQL با استفاده از یک حساب با مجوزهای محدود است. به منظور اطمینان از این امر، اگر mysqld_safe را با استفاده از کاربر root اجرا می‌کنید، از گزینه --user استفاده نمایید. در غیر اینصورت، باید این اسکریپت را در حالی که با حساب mysql وارد شده‌اید اجرا نمایید. در صورتی که جداول اعطا را با استفاده از mysql_install_db ایجاد نکرده باشید، با آغاز به کار سِرور، پیغام زیر در فایل رویدادنگاری ظاهر خواهد شد:

```
mysqld: Can't find file: 'host.frm'
```

این پیغام در صورتی که mysql_install_db را با استفاده از کاربر root و بدون گزینه --user اجرا کرده باشید نیز قابل مشاهده است. در هر صورت، دایرکتوری data را حذف نموده و mysql_install_db را با استفاده از گزینه --user اجرا نمایید.

۸) عملکرد سِرور را می‌توان با استفاده از mysqladmin بررسی نمود. با استفاده از دستورات زیر می‌توان نسخه سِرور و نیز پاسخ به ارتباطات را بررسی کرد.

```
shell> bin/mysqladmin version
shell> bin/mysqladmin variables
```

خروجی بسته به سیستم عامل و نسخه MySQL تا حدی متفاوت است:

```
shell> bin/mysqladmin version
mysqladmin Ver 14.12 Distrib 5.1.64, for pc-linux-gnu on i686
...
Server version 5.1.64
Protocol version 10
Connection Localhost via unix socket
unix socket /var/lib/mysql/mysql.sock
Uptime: 14 days 5 hours 5 min 21 sec
Threads: 1 Questions: 366 Slow queries: 0
Opens: 0 Flush tables: 1 Open tables: 19
Queries per second avg: 0.000
```

۹) از امکان خاموش کردن سِرور اطمینان حاصل نمایید:

```
shell> bin/mysqladmin -u root shutdown
```

۱۰) از امکان اجرای مجدد سِرور اطمینان حاصل کنید. این کار با استفاده از mysql_safe یا با فراخوانی mysqld به صورت مستقیم قابل انجام است. به عنوان مثال:

```
shell> bin/mysqld_safe --user=mysql &
```

۱۱) بررسی‌های ساده‌ای را به منظور اطمینان از قابلیت بازیابی اطلاعات از سِرور انجام دهید.

۱۲) در این مرحله سِرور در حال اجرا است. اما باید توجه داشت که هیچ یک از حساب‌های MySQL دارای گذرواژه نبوده و امکان دسترسی کامل به پایگاه‌های داده تست وجود دارد.

۱۳) با استفاده از اسکریپت bin/mysql_setpermission، و در صورت نصب پیمانه‌های DBI و DBD::mysql می‌توان حساب‌های جدید را ایجاد کرد.

در ادامه برخی از مهمترین مسائل و مشکلاتی را که به هنگام استفاده از mysql_install_db با آن روبرو هستیم را مورد بحث و بررسی قرار می‌دهیم. هدف استفاده از اسکریپت mysql_install_db ایجاد جداول مجوز جدید در MySQL است. این اسکریپت جداول جاری را بازنویسی نکرده و تغییری در داده‌های دیگر ایجاد نمی‌کند. به منظور ایجاد مجدد جداول مجوز، در ابتدا سِرور mysqld را متوقف نمایید. سپس نام دایرکتوری data/mysql را تغییر داده و mysql_install_db را اجرا نمایید. فرض کنید که دایرکتوری جاری شما دایرکتوری نصب MySQL بوده و mysql_install_db در دایرکتوری bin قرار دارد و دایرکتوری داده

نیز data است. به منظور تغییر نام پایگاه داده MySQL و اجرای مجدد mysql_install_db از دستورات زیر استفاده نمایید:

```
shell> mv data/mysql data/mysql.old
shell> scripts/mysql_install_db --user=mysql
```

در اجرای mysql_install_db ممکن است با مشکلات زیر روبرو شوید:

- **اسکریپت قادر به نصب جداول مجوز نیست:** ممکن است این برنامه جداول مجوز را نصب نکرده و پیغام زیر تولید شود:

```
Starting mysqld daemon with databases from <XXXXXX>
mysqld ended
```

در این حالت، باید فایل رویدادنگاری خطا را به دقت بررسی نمایید. این فایل در دایرکتوری <XXXXXX> قرار داشته و شامل دلایل عدم اجرای mysqld است.

- **یک پردازه mysqld در حال اجرا است:** در این حالت سِرور فعال است و جداول اعطا پیش از این ایجاد شده‌اند. برنامه mysql_install_db تنها یک بار باید اجرا شود و در این شرایط نیازی به اجرای مجدد آن نیست.
- **در حالتی که یک سِرور در حال اجرا است امکان نصب یک سِرور mysqld دیگر وجود ندارد:** این مشکل زمانی بروز می‌کند که یک MySQL نصب شده وجود دارد و نیازمند نصب یک نمونه دیگر در مکانی دیگر هستیم. به عنوان مثال، ممکن است در صورت وجود یک پایگاه داده تولید، نیازمند نصب یک پایگاه داده تست باشیم. در این حالت سِرور دوم سعی در استفاده از درگاه سِرور اول را داشته و در این حالت یکی از پیغام‌های خطای زیر بروز می‌کند:

```
Can't start server: Bind on TCP/IP port:
Address already in use
Can't start server: Bind on unix socket…
```

- **به دایرکتوری /tmp دسترسی نوشتن ندارید:** در صورتی که امکان ایجاد فایلهای موقت را در مکان پیش فرض (دایرکتوری /tmp) نداشته باشید با اجرای mysql_install_db یا سِرور با خطا مواجه می‌شوید. در این صورت می‌توان مکانی دیگر را برای فایل‌های موقت مشخص نمود. در دستور زیر some_temp_dir نام و مسیر کامل دایرکتوری‌هایی را مشخص می‌کند که به آن دسترسی نوشتن دارید:

```
shell> TMPDIR=/some_tmp_dir/
shell> MYSQL_UNIX_PORT=/some_tmp_dir/mysql.sock
shell> export TMPDIR MYSQL_UNIX_PORT
```

در ادامه امکان اجرای mysql_install_db و سِرور با استفاده از دستورات زیر وجود دارد:

```
shell> scripts/mysql_install_db --user=mysql
shell> bin/mysqld_safe --user=mysql  &
```

در صورتی که mysql_install_db در دایرکتوری bin نصب شده باشد، دستور را به bin/mysql_install_db تغییر دهید.

گزینه‌های دیگری نیز برای اجرای اسکریپت mysql_install_db وجود دارد:

- به منظور تغییر مجوزها از حالت پیش فرض، می‌توان mysql_install_db را پیش از اجرا تغییر داد. اما استفاده از GRANT و REVOKE برای تغییر مجوزها پس از ایجاد جداول اعطا، گزینه مناسب‌تری است. به عبارت دیگر، می‌توان mysql_install_db را اجرا کرده و با استفاده از mysql -u root mysql به سِرور متصل شد و دستورات GRANT و REVOKE را اجرا کرد.
- به منظور نصب MySQL روی چندین ماشین با مجوزهای یکسان می‌توان دستورات GRANT و REVOKE را در یک فایل ذخیره نمود و پس از اجرای mysql_install_db آن را با استفاده از mysql به عنوان یک اسکریپت اجرا کرد. با استفاده از این مکانیزم نیازی به اجرای دوباره دستورات روی هر ماشین نیست. به عنوان مثال:

```
shell> scripts/mysql_install_db --user=mysql
shell> bin/mysql -u root < your_script_file
```

- می‌توان جداول اعطا را مجددا ایجاد نمود. بدین منظور تمامی فایلهای frm.، MYI. وMYD. را از دایرکتوری پایگاه داده mysql حذف و mysql_install_db را دوباره اجرا کنید.
- می‌توان mysql را به صورت دستی و با استفاده از گزینه skip-grant-tables-- اجرا و اطلاعات مجوزها را با استفاده از mysql اضافه نمود:

```
shell> bin/mysqld_safe --user=mysql --skip-grant-tables  &
shell> bin/mysql mysql
```

با استفاده از mysql دستورات SQL موجود در mysql_install_db را اجرا نمایید. باید توجه داشت که در انتها، به منظور بارگذاری مجدد جداول اعطا، دستورات flush-privileges یا mysqladmin reload را اجرا نمایید. در صورت عدم استفاده از mysql_install_db نه تنها باید مجوز را به صورت دستی ایجاد کنید بلکه باید جداول اعطا را نیز ایجاد نمایید.

اجرا/توقف خودکار سمپاد:

به صورت کلی، اجرای سمپاد mysqld از طرق زیر امکان‌پذیر است:

۱) اجرای mysqld به صورت مستقیم: این امکان در تمامی سیستم‌های عامل وجود دارد.
۲) اجرای سِرور MySQL به عنوان یک سرویس ویندوز: این سرویس می‌تواند به گونه‌ای تنظیم شود که با آغاز به کار ویندوز سِرور را اجرا نماید و یا امکان اجرای دستی آن را فراهم آورد.
۳) اجرای mysqld_safe: این اسکریپت گزینه‌های مناسب برای mysqld را مشخص و آن را اجرا می‌نماید. این اسکریپت در سیستم‌های عامل مبتنی بر لینوکس قابل استفاده است.
کن mysql.server را فراخوانی نمایید. این اسکریپت سِرور را با استفاده از mysqld_safe اجرا می‌کند.
۵) استفاده از چارچوب مدیریت سرویس (SMF) Solaris یا OpenSolaris به منظور مقداردهی اولیه و کنترل شروع به کار MySQL.

به منظور روشن و خاموش کردن دستی سِرور با استفاده از اسکریپت mysql.server از آرگومانهای start و stop استفاده نمایید:

```
shell> mysql.server start
shell> mysql.server stop
```

پیش از اینکه mysql.server سِرور را اجرا نماید، مسیر را به دایرکتوری نصب MySQL تغییر داده و سپس mysqld_safe را فراخوانی نماید. برای اجرای سِرور با یک کاربر خاص، گزینه user را به گروه [mysqld] در فایل /etc/my.cnf اضافه نمایید.

فراخوانی mysql.server stop، سِرور را با ارسال سیگنالی به آن متوقف می‌نماید. همچنین می‌توان سِرور را به صورت دستی با اجرای mysqladmin shutdown متوقف نمود.

به منظور روشن و خاموش کردن MySQL به صورت خودکار باید دستورات start و stop را در مکان‌های مناسب در فایلهای /etc/rc* اضافه کرد. در صورتی که از بسته RPM، سِرور لینوکس استفاده می‌کنید، اسکریپت mysql.server ممکن است در دایرکتوری /etc/init.d با نام mysql نصب شده باشد. در صورتی که MySQL را از روی فایل منبع و یا یک فرمت دودویی نصب می‌کنید که mysql.server را به صورت خودکار نصب نمی‌کند، می‌توانید آن را به صورت دستی نصب نمایید. این اسکریپت از دایرکتوری support-files قابل دسترسی است. به منظور نصب mysql.server به صورت دستی آن را با نام mysql در دایرکتوری /etc/init.d کپی و سپس آن را اجرا کنید.

```
shell> cp mysql.server /etc/init.d/mysql
shell> chmod +x /etc/init.d/mysql
```

پس از نصب اسکریپت، دستورات مورد نیاز برای فعال‌سازی آن در شروع به کار سیستم، به سیستم عامل شما بستگی دارد. در لینوکس می‌توان از chkconfig استفاده نمود:

```
shell> chkconfig --add mysql
```

در برخی سیستم‌های لینوکس، استفاده از دستور زیر به منظور فعال‌سازی کامل اسکریپت mysql مورد نیاز است:

```
shell> chkconfig --level 345 mysql on
```

در FreeBSD اسکریپت‌های شروع به کار معمولا در /usr/local/etc/rc.d/ ذخیره می‌شوند. باید توجه داشت که اسکریپت‌های این دایرکتوری تنها در صورتی اجرا می‌شوند که دارای پسوند .sh باشند. به عبارت دیگر، در FreeBSD ، اسکریپت mysql.server باید به صورت usr/local/etc/rc.d/mysql.server.sh/ نصب شود. به عنوان راه‌کاری دیگر، برخی سیستم‌های عامل از /etc/rc.local یا /etc/init.d/boot.local به منظور راه‌اندازی سرویس‌های اضافی در زمان آغاز به کار سیستم استفاده می‌کنند. برای راه‌اندازی MySQL با استفاده از این روش می‌توان دستوری معادل دستور زیر را به فایل راه‌انداز مناسب اضافه نمود.

```
/bin/sh -c 'cd /usr/local/mysql; ./bin/mysqld_safe --user=mysql ' &
```

می‌توان گزینه‌های دیگری را برای mysql.server در یک فایل /etc/my.cnf سراسری اضافه نمود. یک فایل /etc/my.cnf معمولاً به صورت زیر است:

```
[mysqld]
datadir=/usr/local/mysql/var
socket=/var/tmp/mysql.sock
port=3306
user=mysql
[mysql.server]
basedir=/usr/local/mysql
```

اسکریپت mysql.server از گزینه‌های زیر پشتیبانی می‌نماید: basedir، datadir و pid-file. باید توجه داشت که این گزینه‌ها باید در یک فایل تنظیمات قرار گیرند و استفاده از آنها در خط فرمان امکانپذیر نیست. این اسکریپت تنها از آرگومانهای start و stop در خط فرمان پشتیبانی می‌نماید. جدول زیر، گروه‌های گزینه مورد استفاده توسط سِرور و هر یک از اسکریپت‌های راه‌انداز را نمایش می‌دهد:

اسکریپت	گروه‌های گزینه
mysqld	[mysqld] [server] [mysqld-major_version]
mysqld_safe	[mysqld] [server] [mysqld_safe]
mysql.server	[mysqld] [mysql.server] [server]

در این جدول، گروه [mysqld-major_version] به این معنا است که گروه‌هایی مانند [mysqld-5.0] و [mysqld-5.1] توسط سِرورهایی با نسخه‌های x.5.0 و x.5.1 خوانده می‌شوند.

راه اندازی سمپاد MySQL توسط یک کاربر عادی:

در سیستم عامل ویندوز سِرور را می‌توان به عنوان یک سرویس ویندوز و یا استفاده از یک حساب کاربری عادی اجرا کرد. در سیستم عامل لینوکس، امکان اجرای سِرور MySQL با نام mysqld توسط تمامی کاربران وجود دارد. در این صورت سِرور MySQL نباید تحت کاربر root اجرا شود. اما باید توجه داشت که، به دلایل امنیتی، سِرور مذکور نباید با استفاده از کاربر root اجرا شود. به عنوان مثال در این صورت، یک کاربر MySQL با مجوز FILE می‌تواند فایل‌هایی را به عنوان root ایجاد نماید (به عنوان مثال، /root/.bashrc). به منظور جلوگیری از این مشکل، mysql تنها در حالتی به عنوان کاربر root اجرا می‌شود که از پارامتر --user=root برای راه‌اندازی آن استفاده شده باشد.

بنابراین mysqld باید به عنوان یک کاربر عادی و بدون مجوزهای خاص اجرا شود. می‌توان یک حساب جداگانه تحت عنوان mysql ایجاد نموده و از این حساب به منظور انجام اعمال مدیریتی در MySQL استفاده نمود. به این منظور گام‌های زیر باید طی شود:

۱. سِرور را متوقف نمایید.
۲. دایرکتوری‌ها و فایل‌های پایگاه داده را به گونه‌ای تغییر دهید که کاربر مورد نظر امکان خواندن و نوشتن در آنها را داشته باشد. در صورتی که این کار را انجام ندهید، کاربر قادر به دسترسی به پایگاه‌های داده و جداول نخواهد بود.

```
shell> chown -R user_name /path/to/mysql/datadir
```

۳. سِرور را با استفاده از کاربر مورد نظر اجرا نمایید.

به منظور اجرای mysqld به عنوان یک کاربر دیگر، می‌توان نام کاربری را در گروه [mysqld] در فایل /etc/my.cnf یا فایل my.cnf در دایرکتوری داده سِرور مشخص نمود. به عنوان مثال:

```
[mysqld]

user = mysql
```

در این صورت سِرور با استفاده از کاربر مشخص شده شروع به کار می‌کند. این امر در مواردی که سِرور به صورت دستی، با استفاده از mysqld_safe یا mysql.server اجرا شود، صادق است.

۸-۲ رهنمون‌های امنیتی احراز اصالت کاربران

کنترل دسترسی در MySQL در دو گام انجام می‌شود (۱): احراز اصالت و (۲): بررسی مجوزها. در گام اول، حساب کاربری تشخیص و گذرواژه ارائه شده با آن تطبیق داده می‌شود. در گام دوم، مجوزهای کاربر با عملی که قصد انجام آن را دارد تطبیق داده می‌شود. در این بخش به امنیت احراز اصالت می‌پردازیم و در بخش بعد، مجوزها در MySQL مورد بحث و بررسی قرار می‌گیرد. در این راستا، تعیین نام‌های حساب، چگونگی تشخیص نام‌کاربری هنگام اتصال، امن‌سازی حساب‌ها، امنیت گذر واژه، درهم‌سازی گذر واژه، تخصیص گذر واژه به حساب‌های Root، تخصیص گذر واژه به حساب‌های بی‌نام مورد بحث و بررسی قرار می‌گیرد.

مشخص نمودن نام‌های حساب:

نام‌های حساب در MySQL شامل یک نام کاربری و یک نام میزبان است. این امر امکان ایجاد حساب و ارتباط برای کاربران با یک نام و از میزبان‌های مختلف را فراهم می‌آورد. در این زیر بخش به نحوه ایجاد نام‌های حساب می‌پردازیم. این امر در MySQL با استفاده از دستورات CREATE USER، GRANT و SET PASSWORD انجام می‌شود:

- نام‌های حساب به صورت 'user_name'@'host_name' ایجاد می‌شوند.
- یک نام حساب که فقط شامل نام کاربری باشد هم‌ارز با 'user_name'@'%' است. به عنوان مثال، 'me' هم‌ارز با 'me'@'%' است.
- نام کاربری و نام میزبان در صورتی که مجاز باشد می‌تواند بدون علامت نقل قول استفاده شود. علامت نقل قول تنها در صورتی اجباری است که در نام کاربری یا میزبان از کاراکترهای خاص استفاده شود. به عنوان مثال: 'test-user'@'%.com'
- از علامت نقل قول برای نام کاربری و میزبان باید به صورت جداگانه استفاده شود.

نام‌های حساب در جداول اعطا در پایگاه داده mysql در ستون‌های مجزا (نام کاربری و نام حساب) ذخیره می‌شوند:

- جدول user شامل یک سطر برای هر حساب است. ستون‌های User و Host نام کاربری و نام میزبان را ذخیره می‌نمایند. این جدول مجوزهای سراسری هر حساب را نیز نمایش می‌دهد.
- جداول اعطای دیگر مجوزهای یک حساب را برای پایگاه‌های داده و اشیاء هر یک ارائه می‌نمایند. این جداول حاوی ستون‌های User و Host به منظور ذخیره‌سازی نام‌های حساب است. هر سطر از این جداول به یک حساب در جدول user مرتبط است که مقادیر User و Host مشابهی را دارا است.

مقادیر خاص مختلفی در نام‌های کاربری و میزبان قابل استفاده هستند. یک نام کاربری یا یک مقدار غیر تهی است که باید با نام کاربری ارتباط ورودی تطبیق داده شود و یا یک رشته تهی است که با تمامی نام‌های کاربری تطبیق می‌یابد. یک حساب با نام کاربری تهی یک کاربر بی‌نام است. به منظور مشخص نمودن یک کاربر بی‌نام در عبارات SQL می‌توان از یک نام کاربری تهی، همانند ''@'localhost' استفاده کرد. نام میزبان یک حساب می‌تواند به شکل‌های مختلفی نوشته شود:

- میزبان می‌تواند یک نام و یا یک IP باشد.
- می‌توانید از کاراکترهای "٪" و "_" در نام میزبان و یا آدرس IP استفاده نمایید.

از آنجا که می‌توان از کاراکترهای ویژه در نام میزبان استفاده کرد، به عنوان مثال '192.168.1.٪'، یک مهاجم می‌تواند از این امکان سوء استفاده نموده و میزبانی با نام somewhere.com.192.168.1 را فراخوانی نماید. به منظور پیشگیری از چنین شرایطی، MySQL از استفاده از نام‌های میزبانی که با رقم و نقطه آغاز می‌شوند پیشگیری می‌نماید. برای نام میزبانی که به صورت آدرس IP مشخص شده است می‌توان تعداد بیت‌های مربوط به قسمت شبکه آن را تعیین نمود. فرمت دستور به شکل host_ip / netmask است. به عنوان مثال:

```
CREATE USER 'david'@'192.58.197.0/255.255.255.0';
```

netmask می‌تواند آدرسهای ۸، ۱۶، ۲۴ یا ۳۲ بیتی باشد. به عنوان مثال:

- ۱۹۲,۰,۰,۰ / ۲۵۵,۰,۰,۰ : نمایشگر یک میزبان روی کلاس A یک شبکه (۱۹۲)
- ۱۹۲,۱۶۸,۰,۰ / ۲۵۵,۲۵۵,۰,۰ : نمایشگر یک میزبان روی کلاس B
- ۱۹۲,۱۶۸,۱,۰ / ۲۵۵,۲۵۵,۲۵۵,۰: نمایشگر یک میزبان در کلاس C
- ۱۹۲,۱۶۸,۱,۱ : یک میزبان با یک آدرس IP خاص

باید توجه داشت که سِرور با استفاده از مقدار برگشتی از DNS نام میزبان ماشین کاربر را با نام میزبان موجود در نام حساب تطبیق می‌دهد.

چگونگی تشخیص نام‌کاربری هنگام اتصال:

کنترل دسترسی در MySQL در دو گام انجام می‌شود. در گام اول و در هنگام ارتباط با سِرور MySQL، سِرور ارتباطات را بر مبنای شناسه و گذرواژه احراز هویت می‌نماید. در صورت موفقیت احراز هویت، سِرور به گام دوم کنترل دسترسی وارد می‌شود و منتظر پرس‌وجوها می‌ماند. تصمیم‌گیری در گام دوم با توجه به مدل کنترل دسترسی سمپاد و بر اساس مجوزهای تعیین شده انجام می‌شود. شناسه کاربر که در گام اول استفاده می‌شود همانطور که در بخش مربوط به حساب‌های کاربری توضیح داده شد، مبتنی بر اطلاعات زیر است:

- میزبانی که ارتباط از سمت آن برقرار می‌شود.
- نام کاربری MySQL

احراز هویت با استفاده از ستون‌های حوزه جدول user انجام می‌شود. این ستون‌ها عبارتند از: Host، User و Password. ارتباط در صورتی پذیرفته می‌شود که ستون‌های User و Host در جدول user با نام میزبان و شناسه کاربری مطابقت داشته باشند و کاربر گذرواژه درستی را ارسال کرده باشد. در صورتی که ستون User تهی نباشد، نام کاربری ارتباط ورودی باید دقیقا مطابق این مقدار باشد. اما در صورتی که این مقدار تهی باشد با تمامی نام‌های کاربری مطابقت دارد. چنین کاربری در سیستم به صورت یک کاربر بی‌نام شناخته می‌شود. این امر بدین معنی است که از یک نام کاربری تهی برای تمامی بررسی‌های کنترل دسترسی بعدی استفاده خواهد شد.

ستون Password می‌تواند تهی باشد. این مقدار به این معنی نیست که تمامی گذرواژه‌ها با آن تطابق دارند، بلکه به معنای آن است که کاربر باید با گذرواژه تهی به سمپاد متصل شود. مقادیر غیر تهی گذرواژه در جدول user گذرواژه‌های رمزگذاری شده هستند. در نتیجه، گذرواژه وارد شده از سوی کاربر با استفاده از تابع PASSWORD رمز و با این مقدار رمز شده مقایسه می‌شود. جدول زیر ترکیب‌های مختلف از مقادیر User و Host را در جدول user نمایش می‌دهد:

نام میزبان	نام کاربر	ارتباطات مجاز
‘alpha.example.com’	‘john’	john از alpha.example.com
‘alpha.example.com’	‘’	هر کاربر از alpha.example.com
‘%’	‘john’	کاربر john از تمامی میزبان‌ها
‘%’	‘’	تمامی کاربران از تمامی میزبان‌ها
‘%.example.com’	‘john’	کاربر john که از هر میزبانی زیر دامنه ‘example.com’ متصل می‌شود

ممکن است نام میزبان و نام کاربری در ارتباط با بیش از یک سطر در جدول user تطابق یابد. به عنوان مثال، در جدول ارائه شده، تعداد زیادی از سطرها با کاربر john که از دامنه ‘alpha.example.com’ متصل شده است تطابق دارند. در صورت وجود سطرهای تطابق یافته‌ی مختلف، سِرور باید یکی از آنها را به منظور استفاده، مشخص نماید. این کار به صورت زیر امکان‌پذیر است:

- سِرور در هنگام بارگذاری جدول user در حافظه سطرها را مرتب سازی می‌نماید.
- هنگام برقراری ارتباط یک کاربر، سِرور سطرهای جدول را به ترتیب بررسی می‌نماید.
- سِرور از سطر اولی که با نام میزبان و نام کاربری تطابق داشته باشد استفاده می‌نماید.

سِرور سطرها را بر اساس خاص‌ترین نام میزبان مرتب می‌نماید. نام‌های میزبان و آدرس‌های IP بدون کاراکترهای ویژه خاص‌ترین نام‌ها هستند. کاراکتر '٪' به معنای تمامی میزبان‌ها است و در پایین‌ترین درجه اهمیت قرار دارد. رشته خالی '' نیز به معنای تمامی میزبان‌ها است ولی بعد از '٪' مرتب می‌شود. سطرهایی با مقادیر میزبان یکسان بر اساس نام کاربری خاص‌تر مرتب می‌شوند.

در صورتی که امکان برقراری ارتباط با سِرور وجود دارد اما مجوزهای اعطا شده مطابق انتظار نیست، این امکان وجود دارد که احراز هویت با حسابی دیگر انجام شده است. به منظور مشخص نمودن حسابی که با آن احراز هویت انجام شده است از تابع CURRENT_USER استفاده کنید. این تابع مقداری به صورت user_name@host_name باز می‌گرداند که نمایشگر مقادیر کاربر و میزبان از جدول user است.

تذکر امنیتی: در صورتی که به DNS خود اعتماد ندارید، در جداول اعطای حقوق به جای نام میزبان از آدرس IP استفاده نمایید.

امن‌سازی حساب‌های MySQL:

بخشی از فرآیند نصب MySQL برپاسازی پایگاه داده mysql است که شامل جداول اعطا می‌باشد. محتوای اولیه این جداول بسته به شرایط متفاوت است:

- نسخه‌های ویندوزی شامل جداول از پیش مقداردهی شده هستند.
- در لینوکس، برنامه mysql_install_db جداول اعطا را مقداردهی می‌نماید. برخی روش‌های نصب، این برنامه را به صورت خودکار اجرا می‌نمایند، این در حالی است که در برخی دیگر از روش‌ها باید آن را به صورت دستی اجرا کرد.

جدول اعطای mysql.user حساب‌های کاربری mysql و مجوزهای دسترسی آنها را تعریف می‌کند:

- برخی حساب‌ها دارای نام کاربری root هستند. این حساب‌ها دارای تمامی مجوزها بوده و قادر به انجام هر فعالیتی هستند. گذرواژه اولیه برای حساب‌های root خالی است. در نتیجه، تمامی کاربران می‌توانند به عنوان root به پایگاه داده متصل شوند.
- در ویندوز حساب‌های root به گونه‌ای ایجاد می‌شوند که تنها اجازه برقراری ارتباط برای میزبان محلی را فراهم می‌آورد. در صورتی که کاربر گزینه ENABLE ROOT ACCESS FROM REMOTE MACHINES را در فرآیند نصب انتخاب کرده باشد، نصب‌کننده

ویندوز، حساب root دیگری را ایجاد کرده که امکان برقراری ارتباطات از راه دور را نیز فراهم می‌آورد.

- در لینوکس، تمامی حساب‌های root امکان برقراری ارتباط از میزبان محلی را فراهم می‌آورد.

- برخی حساب‌ها برای کاربران بی‌نام طراحی شده‌اند و دارای نام کاربری خالی هستند. حساب‌های بی‌نام گذرواژه ندارند، در نتیجه تمامی کاربران می‌توانند از آنها به منظور ارتباط با MySQL استفاده کنند.

- در ویندوز، یک حساب بی‌نام وجود دارد که اجازه ارتباطات از میزبان محلی را فراهم می‌آورد. برقراری این ارتباطات با نام localhost امکانپذیر است. این حساب دارای هیچگونه مجوز سراسری نیست.

- در لینوکس، تمامی حساب‌های بی‌نام امکان برقراری ارتباط را از میزبان محلی فراهم می‌آورند. به منظور برقراری ارتباطات می‌توان از نام میزبان localhost برای یکی از حساب‌ها و از آدرس IP برای دیگر حساب‌ها استفاده نمود.

علاوه بر این، جدول mysql.db شامل سطرهایی است که به تمامی حساب‌ها امکان دسترسی به پایگاه داده test و پایگاه‌های داده دیگر که با test_ شروع می‌شوند را می‌دهد. این امر در مورد حساب‌هایی که دارای مجوزهای خاصی نیستند، مثل حساب‌های بی‌نام نیز صادق است. این ویژگی، تست پایگاه داده را تسهیل می‌نماید اما استفاده از آن در پایگاه داده تولید توصیه نمی‌شود. به مدیرانی که تمایل به استفاده از یک خط‌مشی بسته دارند توصیه می‌شود که سطرهای جدول mysql.db را حذف نمایند.

در ادامه دستورالعمل‌هایی برای ایجاد گذرواژه برای حساب‌های اولیه MySQL ارائه می‌شود. علاوه بر این، راهکارهایی برای حذف حساب‌های بی‌نام نیز ارائه خواهد شد. در مثال‌های ارائه شده، پارامتر newpwd را با گذرواژه دلخواه جایگزین نمایید. پارامتر host_name نیز باید به نام میزبان سِرور تغییر یابد.

امنیت گذرواژه در MySQL:

استفاده‌های مختلفی از گذرواژه در MySQL وجود دارد. در ادامه به ارائه راهکارهایی برای امن‌سازی آنها خواهیم پرداخت. علاوه بر این، درهم‌سازی گذرواژه توسط سمپاد MySQL نیز مورد بررسی قرار خواهد گرفت. در ادامه راهکارهای امنیت گذاره واژه برای دو طیف متفاوت از کاربران سمپاد (مدیران و کاربران عادی) بحث و بررسی شده است.

مدیران پایگاه داده می‌توانند از راهکارهای زیر برای امن‌سازی گذرواژه‌ها استفاده نمایند:

- MySQL گذرواژه‌ها را در جدول mysql.user ذخیره می‌کند. دسترسی به این جدول باید از تمامی حساب‌های غیر مدیریتی گرفته شود.
- کاربری که امکان تغییر پوشه plugin (مقدار متغیر سیستمی plugin_dir) یا فایل my.cnf که مکان این پوشه را مشخص می‌نماید دارا است، می‌تواند plugin ها را جایگزین کرده و امکاناتی را که آنها در اختیار قرار می‌دهند، تغییر دهد.
- گذرواژه‌ها می‌توانند به صورت متنی ساده در دستورات SQL مانند CREATE USER، GRANT، و SET PASSWORD یا دستوراتی که از تابع PASSWORD استفاده می‌کنند، ظاهر شوند. در صورت ثبت این دستورات توسط سمپاد، گذرواژه‌ها برای تمامی افرادی که به فایل‌های رویدادنگاری دسترسی دارند، قابل رویت خواهد بود. به منظور رفع این مشکل باید فایل‌های رویدادنگاری را در پوشه‌ای ذخیره کرد که تنها مدیران سِرور و پایگاه داده به آن دسترسی دارند. در صورتی که اطلاعات رویدادنگاری در جداول پایگاه داده mysql ذخیره می‌شود، دسترسی به جداول نباید به هیچ یک از حساب‌های غیر مدیریتی اعطا شود.
- نسخه‌های تکرار شده از پایگاه داده، گذرواژه نسخه اصلی را در فایل master.info نگهداری می‌کنند. لازم به ذکر است که دسترسی به این فایل تنها می‌بایست برای مدیر پایگاه داده وجود داشته باشد.
- دسترسی به نسخه‌های پشتیبان از جداول و فایل‌های رویدادنگاری که شامل گذرواژه هستند را محدود نمایید.

کاربران عادی نیز می‌توانند از راهکارهای زیر به منظور امن‌سازی گذرواژه‌های خود استفاده نمایند:

- پس از اجرای یک برنامه کاربر به منظور ارتباط با سِرور MySQL، گذرواژه باید به گونه‌ای مشخص شود که امکان دسترسی کاربران دیگر به آن وجود نداشته باشد. در ادامه روش‌های مشخص نمودن گذرواژه و نیز ریسک مرتبط با هر روش، بیان شده است. به صورت خلاصه، امن‌ترین روش‌ها روش‌هایی هستند که برنامه کاربر گذرواژه را درخواست می‌کند و یا گذرواژه در یک فایل محافظت شده قرار دارد.

استفاده از p pass-- یا password pass-- در خط فرمان. به عنوان مثال:

```
shell> mysql -u john -pjohn123 db_name
```

استفاده از این روش آسان اما نا امن است؛ گذرواژه ارسالی در معرض دید کاربرانی قرار دارد.

- استفاده از p-- یا password-- در خط فرمان بدون مشخص کردن گذرواژه. در این حالت برنامه کاربر، گذرواژه را به صورت صریح درخواست می‌نماید:

```
shell> mysql -u john -p db_name
Enter password: *******
```

استفاده از این روش نسبت به روش قبل که گذرواژه را در خط فرمان مشخص می‌کند، امن‌تر است. اما باید توجه داشت که این روش تنها برای برنامه‌هایی قابل استفاده است که گذرواژه را به صورت تعاملی دریافت می‌کنند.

- ذخیره‌سازی گذرواژه در یک فایل. به عنوان مثال، در لینوکس می‌توان گذرواژه را در قسمت client از فایل .my.cnf در پوشه home ذخیره نمود. به منظور نگهداری امن گذرواژه در این حالت، فایل مورد نظر نباید برای کاربران دیگر قابل دسترسی باشد. برای اطمینان از این امر، حالت دسترسی فایل را به 400 یا 600 تغییر دهید. به عنوان مثال:

```
shell> chmod 600 .my.cnf
```

به منظور فراخوانی این فایل از خط فرمان، از گزینه --defaults-file=filename استفاده کنید. در اینجا filename مسیر کامل فایل مورد نظر است. به عنوان مثال:

```
shell> mysql --defaults-file=/home/john/mysql-opts
```

- ذخیره‌سازی گذرواژه در متغیر محیطی MYSQL_PWD : این روش مشخص نمودن گذرواژه در MySQL بسیار نا امن است و استفاده از آن توصیه نمی‌شود. برخی از نسخه‌های ps گزینه‌ای را برای نمایش محیط پردازه‌های در حال اجرا دارند، از این‌رو در صورت مقداردهی این متغیر، گذرواژه شما در دسترس کاربرانی که ps را اجرا می‌کنند نیز قرار می‌گیرد.
- در لینوکس، mysql دستورات اجرا شده را در یک فایل تاریخچه ثبت می‌نماید. نام این فایل به صورت پیش فرض، mysql_history. است و در دایرکتوری home ذخیره می‌شود. باید توجه داشت که گذرواژه‌ها در دستوراتی مانند CREATE_USER، GRANT و SET PASSWORD به صورت متنی ساده ظاهر می‌شوند. از این‌رو در صورت استفاده از این دستورات، گذرواژه نیز در فایل تاریخچه ذخیره می‌شود. به منظور نگهداری امن این فایل، دسترسی به آن را محدود نمایید.
- در صورتی که مفسر دستورات مورد استفاده به گونه‌ای پیکربندی شده باشد که تاریخچه‌ای از دستورات را ذخیره‌سازی کند، فایل مورد نظر شامل گذرواژه‌هایی است که در خط فرمان وارد شده است. به عنوان مثال، bash از فایل bash_history. در پوشه home استفاده می‌نماید. دسترسی به اینگونه فایل‌ها نیز باید محدود شود.

درهم‌سازی گذرواژه در MySQL:

سمپاد MySQL حساب‌های کاربری را در جدول user در پایگاه داده mysql ذخیره می‌نماید. تمامی حساب‌ها در این سمپاد دارای گذرواژه هستند که در ستون Password نگهداری می‌شود. باید توجه داشت

که مقدار درهم گذرواژه در این ستون ذخیره می‌شود. این مقدار با استفاده از تابع PASSWORD محاسبه می‌شود. از گذرواژه‌ها در دو فاز از ارتباط کاربر و سِرور استفاده می‌شود:

۱) در مرحله ارتباط کاربر با سِرور، یک مرحله احراز اصالت اولیه وجود دارد که در آن کاربر مقدار درهم گذرواژه خود را به سِرور ارائه می‌دهد.
۲) پس از برقراری ارتباط، کاربر می‌تواند گذرواژه‌ی ذخیره شده در جدول user را تغییر دهد. این امر با استفاده از تابع PASSWORD و ایجاد مقدار درهم‌سازی شده از گذرواژه قابل انجام است.

مکانیزم درهم‌سازی مورد استفاده در نسخه ۴,۱ این سمپاد به روزرسانی شده و امنیت بالاتری را فراهم می‌آورد. اما باید توجه داشت که تنها نسخه‌های ۴,۱ و جدیدتر با این مکانیزم تطابق دارند. این امر باعث بروز برخی مشکلات در ارتباطات کاربر و سِرور می‌شود. به عنوان مثال، کاربری با نسخه پایین‌تر از ۴,۱ در ارتباط با سِروری با نسخه ۴,۱ و جدیدتر با پیغامی مشابه پیغام زیر روبرو می‌شود:

```
shell> mysql -h localhost -u root
Client does not support authentication protocol requested by server; consider upgrading MySQL client
```

در ادامه تفاوت میان مکانیزم‌های جدید و قدیم گذرواژه بررسی می‌شود. پیش از MySQL نسخه ۴,۱، مقادیر درهم‌سازی شده گذرواژه که با استفاده از تابع PASSWORD محاسبه می‌شود، ۱۶ بایت است. ستون Password نیز که به منظور ذخیره‌سازی این گذرواژه‌ها از آن استفاده می‌شود ۱۶ بایتی است. پس از نسخه ۴,۱ این تابع به گونه‌ای تغییر داده شد که مقادیر ۴۱ بایتی را ایجاد می‌نماید. پیرو آن نیز ستون Password در جدول user نیز به گونه‌ای تغییر داده شد که امکان ذخیره‌سازی مقادیر ۴۱ بایتی را داشته باشد:

- در صورت نصب نسخه ۵,۱، ستون Password به صورت خودکار طولی معادل ۴۱ بایت خواهد داشت.
- به روزرسانی از نسخه۴,۱ (نسخه ۴,۱,۱ و نسخه‌های بالاتر) به نسخه ۵,۱ دارای مشکل خاصی نیست زیرا این نسخه‌ها از مکانیزم‌های یکسانی برای درهم‌سازی گذرواژه استفاده می‌کنند.

ستون جدید Password توانایی نگهداری گذرواژه‌های قدیمی و جدید را دارا است. تشخیص قالب مقادیر درهم‌سازی شده‌ی گذرواژه به دو صورت امکانپذیر است:

- تفاوت اصلی در طول این مقادیر است (۱۶ بایت و ۴۱ بایت)
- مقادیر درهم‌سازی شده در قالب جدید با کاراکتر x شروع می‌شود، در حالی که این امر در مورد مقادیر قدیمی صادق نیست.

نحوه استفاده سِرور از مقادیر درهم‌سازی شده گذرواژه در فرآیند احراز اصالت به طول ستون Password بستگی دارد:

- در صورتی‌که طول ستون کوتاه باشد، تنها از احراز اصالت مبتنی بر مقادیر درهم‌سازی شده کوتاه استفاده می‌شود.
- در صورتی‌که طول ستون طولانی باشد، توانایی ذخیره هر دو نوع مقدار درهم‌سازی شده را دارا است و سِرور می‌تواند از هر دو قالب استفاده نماید:
  - امکان ارتباط برای کاربری با نسخه ۴,۱ و پایین‌تر وجود دارد. از آنجا که این نسخه‌ها تنها با مکانیزم قدیمی درهم‌سازی سازگاری دارند، تنها حساب‌هایی که مقادیر درهم‌سازی شده با طول کوتاه دارند را احراز اصالت می‌نمایند.
  - برنامه‌های کاربران با نسخه ۴,۱ و جدیدتر می‌توانند حساب‌هایی با مقادیر درهم‌سازی شده کوتاه و طولانی را احراز اصات نمایند.

نحوه تولید مقادیر درهم‌سازی شده گذرواژه توسط سِرور، بسته به طول ستون Password و گزینه --old-passwords متفاوت است. یک سِرور نسخه ۴,۱ و بعد از آن تنها در شرایط خاصی مقادیر درهم‌سازی شده طولانی را تولید می‌کند: ستون Password امکان ذخیره‌سازی این مقدار را داشته باشد و نیز از گزینه --old-passwords استفاده نشده باشد. هدف از استفاده از گزینه --old-passwords فراهم آوردن سازگاری رو به عقب با نسخه‌های پیش از ۴,۱ در شرایط خاص است.

تخصیص گذرواژه به حساب‌های root:

تخصیص گذرواژه به این حساب‌ها از سه راه مختلف امکانپذیر است:

- استفاده از عبارت SET PASSWORD
- استفاده از دستور UPDATE
- استفاده از برنامه mysqladmin

به منظور تخصیص گذرواژه با استفاده از SET PASSWORD، به سِرور متصل شده و برای هر یک از حساب‌های root که در جدول mysql.user لیست شده است، از یک دستور SET PASSWORD استفاده نمایید. توجه داشته باشید که گذرواژه را با استفاده از تابع PASSWORD رمزگذاری نمایید. در سیستم عامل ویندوز عملیات زیر را انجام دهید:

```
shell> mysql -u root
mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('newpwd');
mysql> SET PASSWORD FOR 'root'@'127.0.0.1' = PASSWORD('newpwd');
```

```
mysql> SET PASSWORD FOR 'root'@'%' = PASSWORD('newpwd');
```

در سیستم عامل لینوکس، اعمال زیر را انجام دهید:

```
shell> mysql -u root
mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('newpwd');
mysql> SET PASSWORD FOR 'root'@'127.0.0.1' = PASSWORD('newpwd');
mysql> SET PASSWORD FOR 'root'@'host_name' = PASSWORD('newpwd');
```

همچنین می‌توان از یک دستور که گذرواژه را به تمامی حساب‌های root اختصاص می‌دهد، استفاده کرد.

```
shell> mysql -u root
mysql> UPDATE mysql.user SET Password = PASSWORD('newpwd')  WHERE User = 'root';
mysql> FLUSH PRIVILEGES;
```

دستور FLUSH سِرور را وادار به بازخوانی جداول اعطا می‌کند. بدون استفاده از چنین دستوری، تغییرات گذرواژه تا زمان اجرای مجدد سِرور از دید آن مخفی می‌ماند. به منظور تخصیص گذرواژه به حساب‌های root با استفاده از mysqladmin، دستورات زیر را اجرا نمایید:

```
shell> mysqladmin -u root password "newpwd"
shell> mysqladmin -u root -h host_name password "newpwd"
```

این دستورات در سیستم‌های عامل لینوکس و ویندوز قابل استفاده است. استفاده از علامت نقل قول دوتایی برای گذرواژه، در تمامی موارد ضروری نیست. اما در صورتی که گذرواژه شامل فاصله خالی یا کاراکتر خاص باشد استفاده از این علامت ضروری است. استفاده از mysqladmin برای حساب 'root'@'127.0.0.1' کار نمی‌کند. در این حالت باید از روش SET PASSWORD استفاده کرد.

تخصیص گذرواژه به حساب‌های بی‌نام:

دستورات mysql زیر شامل یک گزینه p- و مبتنی بر این فرض هستند که گذرواژه حساب‌های root مقداردهی شده‌اند. به منظور تخصیص گذرواژه به حساب‌های بی‌نام با استفاده از حساب root، به سِرور متصل شده و از دستور SET PASSWORD یا UPDATE استفاده نمایید. لازم به ذکر است که می‌بایست از رمزگذاری گذرواژه با استفاده از تابع PASSWORD اطمینان حاصل نمایید. به منظور استفاده از SET PASSWORD در ویندوز اعمال زیر را انجام دهید:

```
shell> mysql -u root -p
Enter password: (enter root password here)
mysql> SET PASSWORD FOR ''@'localhost' = PASSWORD('newpwd');
```

به منظور استفاده از SET PASSWORD در لینوکس اعمال زیر را انجام دهید:

```
shell> mysql -u root -p
Enter password: (enter root password here)
mysql> SET PASSWORD FOR "@'localhost' = PASSWORD('newpwd');
mysql> SET PASSWORD FOR "@'host_name' = PASSWORD('newpwd');
```

به منظور تخصیص گذرواژه به حساب‌های بی‌نام با استفاده از دستور UPDATE، اعمال زیر را انجام دهید:

```
shell> mysql -u root -p
Enter password: (enter root password here)
mysql> UPDATE mysql.user SET Password = PASSWORD('newpwd')  WHERE User ='';
mysql> FLUSH PRIVILEGES;
```

دستور FLUSH منجر به بازخوانی جداول اعطا توسط سِرور می‌شود، در صورت عدم استفاده از این دستور، تغییر گذرواژه تا زمان آغاز به کار مجدد سِرور اعمال نخواهد شد.

۸-۳ رهنمون‌های امنیتی کنترل دسترسی کاربران

کنترل دسترسی در MySQL در دو گام انجام می‌شود (۱): احراز اصالت و (۲): بررسی مجوزها. در گام اول، حساب کاربری تشخیص داده می‌شود و گذرواژه ارائه شده با آن تطبیق داده می‌شود. در گام دوم، مجوزهای کاربر با عملی که قصد انجام آن را دارد تطبیق داده می‌شود. در بخش قبلی به امنیت احراز اصالت پرداختیم. در این بخش به مجوزها در MySQL می‌پردازیم. در این راستا، مجوزهای موجود در سمپاد MySQL، جداول اعطای سیستم‌مجوزدهی، نحوه بررسی مجوزها و بازه زمانی تاثیر مجوزها را مورد بحث و بررسی قرار می‌دهیم.

کارکرد اصلی سیستم مجوزدهی MySQL احراز اصالت کاربری و مجوزدهی به او به منظور اجرای دستوراتی نظیر SELECT، INSERT، UPDATE و DELETE است. از دیگر وظایف این سیستم امکان وجود کاربران بی‌نام و اعطای مجوزها به توابع خاص MySQL مانند LOAD DATA INFILE و همچنین اعمال مدیریتی است. البته امکان انجام برخی از عملیات‌ها با استفاده از این سیستم مجوزدهی وجود ندارد که در ادامه برخی از آنها لیست شده‌اند:

- امکان لغو دسترسی برای یک کاربر خاص به صورت صریح وجود ندارد. این امر بدین معنی است که نمی‌توان یک کاربر خاص را مشخص نمود و ارتباط آن را با پایگاه داده قطع کرد.

- امکان اعطای مجوز ایجاد یا حذف جداول در یک پایگاه داده، بدون اعطای مجوز حذف یا ایجاد خود پایگاه داده، وجود ندارد.
- امکان تخصیص یک گذرواژه به یک حساب وجود دارد. این در حالی است که نمی‌توان گذرواژهای را به یک شی خاص همانند پایگاه داده، جدول، و غیره اختصاص داد.

سِرور اطلاعات مجوزها را در جداول اعطای پایگاه داده mysql ذخیره‌سازی می‌نماید. سِرور MySQL محتوای این جداول را در حافظه بارگذاری نموده و تصمیم‌گیری‌های کنترل دسترسی بر مبنای این نسخه از اطلاعات موجود در حافظه انجام می‌شود. سیستم مجوزهای MySQL این اطمینان را فراهم می‌آورد که تنها اعمال مجاز برای کاربران قابل انجام است. هویت یک کاربر بر مبنای نام کاربری و میزبانی که با استفاده از آن ارتباط برقرار می‌گردد، مشخص می‌شود. با ارائه یک درخواست، سیستم مجوزهای مربوطه را بر اساس هویت و درخواست مطرح شده اعطا می‌نماید.

برخی مجوزها از قبیل SHUTDOWN مستقل از پایگاه داده خاص هستند. سایر مجوزها در سطوح مختلفی قابل تعریف هستند: سراسری، پایگاه داده، جدول، ستون و روال ذخیره شده. در صورتی که مجوز عملی در سطح بالاتر داده شده باشد، سایر سطوح بررسی نمی‌شوند، در غیر این صورت سطوح پایین‌تر بسته به نوع عمل، بررسی می‌شوند. سِرور MySQL به صورت پیش فرض و به هنگام آغاز به کار، جداول مربوط به مجوزها را در حافظه واکشی می‌کند. باید توجه داشت که در صورت تغییر مجوزهای مربوط به یک کاربر، تغییر مربوطه بلافاصله اعمال نمی‌شود. در صورتی که تغییر مجوزها از طریق دستورات ویژه کنترل دسترسی انجام شود، مقادیر واکشی شده نیز اصلاح می‌شوند. اما اگر تغییرات با دستورات معمولی و مستقیماً در جدول‌های مربوطه اعمال شود، باید از دستور FLUSH PRIVILEGES یا دستور متناظر mysqladmin استفاده شود. تغییرات در سطوح جدول و ستون در اولین درخواست پس از آن لحاظ می‌شوند. اما تغییرات در سطح پایگاه داده پس از اولین استفاده از دستور USE و تغییر در مجوزهای سراسری و گذرواژه‌ها فقط در خصوص اتصال‌هایی که جدیداً برقرار می‌شوند، در نظر گرفته می‌شوند.

مجوزهای موجود در MySQL:

مجوزهای موجود در زمینه‌های مختلف قابل دسته‌بندی هستند که در ادامه برخی از مهم‌ترین آنها آورده شده است:

- مجوزهای مدیریتی، مدیریت سِرور MySQL را برای کاربران امکان‌پذیر می‌سازد. این مجوزها سراسری بوده و منحصر به یک پایگاه داده خاص نیستند.
- مجوزهای پایگاه داده به یک پایگاه داده و تمامی اشیاء آن قابل اعمال است. این مجوزها می‌توانند به یک پایگاه داده خاص یا به صورت سراسری به تمامی پایگاه‌های داده اعمال شوند.

- مجوزهای اشیاء یک پایگاه داده مثل جداول، دیدها، شاخص‌ها و رویه‌های ذخیره شده می‌تواند به اشیاء یک پایگاه داده خاص، اشیائی از یک نوع در یک پایگاه داده یا به صورت سراسری برای تمامی اشیاء از یک نوع در تمامی پایگاه‌های داده اعطا شود.

اطلاعات مربوط به مجوزهای حساب‌ها در جداول user، db، host، tables_priv، columns_priv و procs_priv در پایگاه داده mysql ذخیره می‌شود. لذا به هیچ یک از کاربران (غیر از حسابهای root این سمپاد) حق دسترسی به جدول user در پایگاه داده‌ی mysql را اعطا نکنید. این امر از اهمیت ویژه‌ای برخوردار است و به هر حال اعطای بیش از حد مجوزها نادرست است.

در برخی نسخه‌های MySQL تغییراتی در ساختار جداول اعطا به منظور اضافه نمودن مجوزهای دسترسی و ویژگیهای جدید، ایجاد شده است. با به روزرسانی MySQL باید جداول اعطا را نیز به منظور بهره‌گیری از ویژگی‌های جدید به روزرسانی نمود. جدول زیر، نام مجوزهای استفاده شده در دستورات GRANT و REVOKE ، به همراه نام ستون مرتبط با هر مجوز در جداول اعطا و زمینه‌ای که مجوز در آن قابل اعمال است را نمایش می‌دهد:

مجوز	ستون	زمینه
CREATE	Create_priv	پایگاه‌های داده، جداول و شاخص‌ها
DROP	Drop_priv	پایگاه‌های داده، جداول و دیدها
GRANT OPTION	Grant_priv	پایگاه‌های داده، جداول و رویه‌های ذخیره شده
LOCK TABLES	Lock_tables_priv	پایگاه‌های داده
REFERENCES	References_priv	پایگاه‌های داده یا جداول
EVENT	Event_priv	پایگاه‌های داده
ALTER	Alter_priv	جداول
DELETE	Delete_priv	جداول
INDEX	Index_priv	جداول
INSERT	Insert_priv	جداول یا ستون‌ها
SELECT	Select_priv	جداول یا ستون‌ها

UPDATE	Update_priv	جداول یا ستون‌ها
CREATE TEMPORARY TABLES	Create_tmp_table_priv	جداول
TRIGGER	Trigger_priv	جداول
CREATE VIEW	Create_view_priv	دیدها
SHOW VIEW	Show_view_priv	دیدها
ALTER ROUTINE	Alter_routine_priv	رویه‌های ذخیره شده
CREATE ROUTINE	Create_routine_priv	رویه‌های ذخیره شده
EXECUTE	Execute_priv	رویه‌های ذخیره شده
FILE	File_priv	دسترسی به فایل‌ها
CREATE USER	Create_user_priv	مدیریت سِرور
PROCESS	Process_priv	مدیریت سِرور
RELOAD	Reload_priv	مدیریت سِرور
REPLICATION CLIENT	Repl_client_priv	مدیریت سِرور
REPLICATION SLAVE	Repl_slave_priv	مدیریت سِرور
SHOW DATABASES	Show_db_priv	مدیریت سِرور
SHUTDOWN	Shutdown_priv	مدیریت سِرور
SUPER	Super_priv	مدیریت سِرور
ALL		مدیریت سِرور
USAGE		مدیریت سِرور

فهرست زیر شرحی کلی از مجوزهای موجود در MySQL را ارائه می‌دهد:

- مجوز ALL به مفهوم تمامی مجوزهای موجود در یک سطح مجوز، غیر از GRANT OPTION است. به عنوان مثال، اعطای ALL در سطح سراسری یا جدول تمامی مجوزهای سراسری یا جدول را اعطا می‌نماید.

- مجوز ALTER استفاده از ALTER TABLE را به منظور تغییر ساختار جداول امکان‌پذیر می‌سازد. علاوه بر این، ALTER TABLE نیازمند مجوزهای CREATE و INSERT نیز می‌باشد. تغییر نام یک جدول نیازمند مجوزهای ALTER و DROP روی جدول قدیمی و مجوزهای ALTER، CREATE و INSERT روی جدول جدید است.
- مجوز ALTER ROUTINE برای تغییر یا حذف رویه‌های ذخیره شده ضروری است.
- مجوز CREATE ایجاد پایگاه‌های داده و جداول جدید را امکان‌پذیر می‌سازد.
- مجوز CREATE ROUTINE به منظور ایجاد رویه‌های ذخیره شده ضروری است.
- مجوز CREATE TEMPORARY TABLES ایجاد جداول موقت را با استفاده از دستور CREATE TEMPORARY TABLES امکان‌پذیر می‌سازد. اعمال دیگر روی جداول موقت، مانند INSERT، UPDATE یا SELECT نیازمند مجوزهای اضافی برای این اعمال در پایگاه داده‌ی شامل جدول موقت یا برای جدول غیر موقت با نام مشابه است. یک راه حل به منظور جداسازی مجوزهای جداول موقت و غیر موقت، ایجاد پایگاه داده خاص استفاده از جداول موقت است. در ادامه و در پایگاه داده‌ی ایجاد شده، اعطای مجوز CREATE TEMPORARY TABLES به همراه دیگر مجوزهای مورد نیاز برای انجام اعمال روی جداول موقت، امکان‌پذیر است.
- مجوز CREATE USER استفاده از مجوزهای CREATE USER، DROP USER، RENAME USER و REVOKE ALL PRIVILEGES را امکان‌پذیر می‌سازد.
- مجوز CREATE VIEW استفاده از CREATE VIEW را امکان‌پذیر می‌سازد.
- مجوز DELETE امکان حذف سطرها از یک جدول را فراهم می‌آورد.
- مجوز DROP امکان حذف پایگاه‌های داده، جداول، و دیدها را فراهم می‌آورد. از نسخه ۵,۱,۱۰ این مجوز برای استفاده از دستور ALTER TABLE ... DROP PARTITION روی جداول پارتیشن شده نیز ضروری است. پس از MySQL نسخه ۵,۱,۱۶، مجوز DROP برای استفاده از دستور TRUNCATE TABLE مورد نیاز است. در صورت اعطای مجوز DROP روی پایگاه داده mysql به یک کاربر، امکان حذف پایگاه داده‌هایی که در آن مجوزهای دسترسی MySQL ذخیره شده است برای کاربر مربوطه وجود دارد.
- مجوز EVENT برای ایجاد، حذف، تغییر و مشاهده رخدادهای برنامه مورد نیاز است. این مجوز در نسخه ۵,۱,۶ اضافه شده است.
- مجوز EXECUTE به منظور اجرای رویه‌های ذخیره شده ضروری است.

- مجوز FILE امکان خواندن و نوشتن فایل‌ها را بر روی سِرور با استفاده از دستورات LOAD DATA INFILE و SELECT ... INTO OUTFILE و نیز تابع LOAD_FILE فراهم می‌آورد. کاربری که مجوز FILE را دارد می‌تواند تمامی فایل‌های موجود در سِرور را بخواند. این امر بدین معنی است که مجوز خواندن تمامی فایل‌های موجود در دایرکتوری‌های پایگاه داده به این کاربر اعطا شده است. علاوه بر این، مجوز FILE امکان ایجاد فایل‌های جدید، در تمامی دایرکتوری‌هایی که سِرور به آنها دسترسی خواندن دارد را فراهم می‌آورد. لذا این مجوز نباید به کاربران غیر مدیریتی اعطا شود. چنین قابلیتی می‌تواند مورد سوء استفاده قرار گیرد. به عنوان مثال، با استفاده از LOAD DATA می‌توان /etc/passwd را در یک جدول بارگذاری و سپس محتویات آن را با دستور SELECT مشاهده نمود.
- مجوز GRANT OPTION امکان اعطا یا لغو مجوزهایی که مالک آن هستید را فراهم می‌آورد.
- مجوز INDEX امکان ایجاد یا حذف شاخص‌ها را فراهم می‌آورد. این مجوز به جداول موجود قابل اعمال است. در صورت دارا بودن مجوز CREATE برای یک جدول می‌توان تعاریف شاخص را در دستور CREATE TABLE لحاظ کرد.
- مجوز INSERT امکان درج سطرها را در جداول یک پایگاه داده فراهم می‌آورد. این مجوز برای اجرای دستورات ANALYZE TABLE، OPTIMIZE TABLE، و REPAIR TABLE ضروری است.
- مجوز LOCK TABLES استفاده از دستورات LOCK TABLES برای قفل کردن جداولی که مجوز SELET برای آنها موجود است را میسر می‌سازد. این دستورات شامل قفل‌های نوشتن که از دسترسی دیگر نشست‌ها به جدول قفل شده جلوگیری می‌نمایند، نیز می‌باشد.
- مجوز PROCESS با نمایش اطلاعات فرآیندهایی که در سِرور اجرا می‌شوند در ارتباط است. در حقیقت، این اطلاعات در مورد دستوراتی است که توسط نشست‌های مختلف اجرا می‌شوند. مجوز مذکور استفاده از SHOW PROCESSLIST یا mysqladmin processlist را به منظور مشاهده فرآیندهای متعلق به دیگر حساب‌ها امکانپذیر می‌سازد. لازم به ذکر است که این مجوز را نباید به کاربران غیر مدیریتی اعطا کرد. خروجی mysqladmin processlist و SHOW PROCESSLIST متن تمامی دستورات اجرا شده را نمایش می‌دهد. در نتیجه تمامی کاربرانی که قادر به مشاهده فهرست پردازه‌های سِرور هستند می‌توانند دستورات اجرا شده توسط کاربران دیگر را نیز مشاهده نمایند.
- مجوز REFERENCES در حال حاضر بدون استفاده است.

- مجوز RELOAD استفاده از دستور FLUSH را امکان‌پذیر می‌سازد. این مجوز استفاده از فرمان‌هایی که معادل دستور FLUSH هستند را نیز امکانپذیر می‌سازد: flush-hosts، flush logs، flush-privileges، flush-status، flush-tables، flush-threads، refresh و reload. دستور reload جداول اعطا را در حافظه بارگذاری می‌نماید. flush-privileges معادل reload است. دستور refresh فایل‌های رویدادنگاری را بسته و باز می‌نماید و تمامی جداول را خالی می‌کند. دستورات دیگر flush کارکردی مشابه refresh را دارا هستند، با این تفاوت که خاص‌تر بوده و در شرایط خاص کاربرد دارند. به عنوان مثال، برای پاک کردن فایل‌های رویدادنگاری، استفاده از flush-logs گزینه مناسب‌تری نسبت به refresh است.
- مجوز REPLICATION CLIENT استفاده از SHOW MASTER STATUS و SHOW SLAVE STATUS را امکان‌پذیر می‌سازد.
- مجوز REPLICATION SLAVE باید به حساب‌هایی که توسط سِروران فرعی به منظور ارتباط با سِرور جاری استفاده می‌شود اعطا شود. بدون این مجوز، سِرور فرعی نمی‌تواند بروزرسانی‌های سِرور اصلی را دریافت کند.
- مجوز SELECT امکان انتخاب سطرهای یک جدول را فراهم می‌آورد. این مجوز برای دستوراتی که مقادیر یک ستون را می‌خوانند نیز مورد نیاز است.
- مجوز SHOW DATABASES امکان مشاهده نام‌های پایگاه داده را با استفاده از دستور SHOW DATABASE فراهم می‌آورد. حساب‌هایی که دارای این مجوز نیستند تنها پایگاه‌های داده‌ای را می‌توانند مشاهده نمایند که مجوزی روی آنها دارند. در صورتی که پایگاه داده با استفاده از گزینه --skip-show-database- اجرا شده باشد، امکان استفاده از این دستور وجود ندارد. توجه داشته باشید که مجوزهای سراسری نیز جزو مجوزهای پایگاه داده محسوب می‌شوند.
- مجوز SHOW VIEW استفاده از SHOW CREATE VIEW را امکان‌پذیر می‌سازد.
- مجوز SHUTDOWN استفاده از دستور mysqladmin shutdown را امکان‌پذیر می‌سازد.
- مجوز SUPER استفاده از CHANGE MASTER TO، KILL یا mysqladmin kill را به منظور حذف فرآیندهای متعلق به حساب‌های دیگر فراهم می‌آورد. mysqld یک ارتباط اضافی به کاربران با مجوز SUPER اختصاص می‌دهد. در نتیجه، یک کاربر root می‌تواند در هر زمان (حتی اگر تمامی ارتباطات معمول در حال استفاده باشد) به سیستم وارد شده و فعالیت سِرور را بررسی نماید. از مجوز SUPER می‌توان به منظور قطع ارتباطات کاربران، تغییر کارکرد سِرور با تغییر مقادیر متغیرهای سیستمی و نظارت بر سِروران استفاده کرد لذا نباید به کاربران عادی داده شود. موارد دیگر استفاده از این مجوز به شرح زیر است:

- PURGE BINARY LOGS
- تغییرات پیکربندی با استفاده از SET GLOBAL به منظور تغییر متغیرهای سیستمی سراسری
- فرمان mysqladmin debug
- فعالسازی و غیر فعالسازی امکان رویدادنگاری
- انجام به روزرسانی اطلاعات حتی در صورتی که متغیر سیستمی read_only فعال شده باشد
- مشخص کردن تمامی حساب‌ها در خصیصه DEFINER برنامه‌های ذخیره شده و دیدها
- امکان برقراری ارتباط با سِرور در شرایطی که تعداد ارتباطات از کران بالای تعیین شده در max_connections بیشتر باشد.

- مجوز TRIGGER اعمال مرتبط با رهاناها را امکان‌پذیر می‌سازد. دارا بودن این مجوز روی یک جدول به منظور ایجاد، حذف یا اجرای یک رهانا روی آن جدول ضروری است. این مجوز در نسخه 5.1.6 اضافه شده است. پیش از این نسخه، این اعمال با استفاده از مجوز SUPER امکانپذیر بوده است.
- مجوز UPDATE امکان به روز رسانی سطرها را فراهم می‌آورد.
- مجوز USAGE در حقیقت به معنای عدم وجود مجوزی خاص است. از این مجوز در سطح سراسری به همراه GRANT به منظور تغییر ویژگی‌های حساب‌ها (همانند محدودیت روی منابع) بدون تأثیر روی مجوزها استفاد می‌شود.

در اعطای مجوز FILE و مجوزهای مدیریتی موارد زیر را در نظر داشته باشید:

- مجوز فایل می‌تواند مورد سوء استفاده واقع شده و از آن به منظور دسترسی به فایل‌هایی که سِرور روی میزبان به آنها دسترسی خواندن دارد استفاده شود. این فایل‌ها شامل فایل‌های با دسترسی خواندن سراسری و فایل‌های موجود در دایرکتوری داده سِرور است. دسترسی به جدول با استفاده از SELECT به منظور انتقال محتوای آن به کاربر انجام می‌شود.
- مجوز GRANT OPTION امکان اعطای مجوزهای یک کاربر را به دیگر کاربران فراهم می‌آورد.
- از مجوز ALTER می‌توان به منظور تغییر سیستم مجوزدهی یا تغییر نام جداول استفاده نمود.
- مجوز SHUTDOWN می‌تواند مورد سوء استفاده واقع شده و از آن برای ایجاد اختلال یا خاموش کردن سمپاد استفاده نمود.

- از مجوز PROCESS می‌توان به منظور مشاهده متن ساده‌ی دستورات در حال اجرا، شامل دستورات تغییر یا مقداردهی گذرواژه استفاده نمود.
- مجوز SUPER می‌تواند به منظور خاتمه دادن به نشست‌های دیگر و نیز تغییر کارکرد سِرور، مورد استفاده قرار گیرد.
- مجوزهای اعطا شده به پایگاه داده mysql می‌تواند به منظور تغییر گذرواژه‌ها و دیگر اطلاعات مجوزهای دسترسی مورد استفاده قرار گیرد. گذرواژه‌ها به صورت رمزشده ذخیره می‌شوند، در نتیجه امکان خواندن آنها وجود ندارد. اما یک کاربر با دسترسی نوشتن به ستون Passwords از جدول user می‌تواند گذرواژه حساب‌ها را تغییر دهد.

جداول اعطای سیستم مجوز دهی:

به صورت معمول، تغییر محتوای جداول اعطا در پایگاه داده mysql با استفاده از دستورات GRANT و REVOKE به صورت غیر مستقیم امکان‌پذیر است. در ادامه، ساختار جداول اعطا بررسی و نحوه استفاده از محتوای آنها توسط سِرور در تعامل با کاربران مورد بحث قرار می‌گیرد. این جداول پایگاه داده شامل اطلاعات مجوزها به صورت زیر هستند:

- user : شامل حساب‌های کاربری و مجوزهای سراسری است.
- db : شامل مجوزهای سطح پایگاه داده است.
- host : در ترکیب با جدول db به منظور تعیین دسترسی از میزبان‌های مختلف استفاده می‌شود.
- tables_priv : شامل مجوزهای سطح جدول است.
- columns_privileges :شامل مجوزهای سطح سطر است.
- process_priv : شامل مجوزهای رویه ذخیره شده و توابع است.

جداول دیگر در پایگاه داده‌های mysql شامل اطلاعات مربوط به اعطای مجوزها نیستند. تمامی جداول اعطا شامل ستون حوزه و مجوز هستند:

- ستون‌های حوزه، دامنه‌ی هر سطر از جدول را مشخص می‌نمایند.
- ستون‌های مجوز، مجوزهای اعطا شده توسط هر سطر از جدول را مشخص می‌نماید. سِرور اطلاعات دریافتی از جداول مجوز مختلف را با یکدیگر ترکیب کرده و مجوزهای نهایی را مشخص می‌نماید.

سِرور از جداول مجوز به صورت زیر استفاده می‌نماید:

- ستون‌های حوزه مربوط به جدول user امکان برقراری یـا رد ارتباطـات ورودی را تعیـین می‌نمایـد. مجوزهای اعطا شده در جدول user مجوزهای سراسری کاربر هستند و بـه تمـامی پایگاه‌هـای داده روی سِرور اعمال می‌شود.
- ستون‌های حوزه جدول db دسترسی کاربران از یک میزبـان خـاص بـه یـک پایگـاه داده را تعیـین می‌نماید. ستون مجوزها دستورالعمل‌های مجاز را مشخص می‌نماید. مجوز تخصیص یافته در سـطح پایگاه داده به آن پایگاه و تمامی اشیاء آن تخصیص می‌یابد.
- از جدول host به همراه جدول db به منظور اعمال یک سطر از جدول db بـه میزبان‌هـای مختلـف استفاده می‌شود. به عنوان مثال، در صورتیکه می‌خواهید یک کاربر از طریق میزبان‌های مختلـف بـا یک پایگاه داده در تماس باشد، مقدار Host را در جدول db خالی گذاشته و در جدول host هر سطر را به هر یک از میزبان‌ها اختصاص دهید.
- جداول tables_priv و columns_privs مشابه جدول db بوده ولی ریزدانه‌تر هستند. در حقیقت این جداول در سطح جدول و ستون اعمال می‌شوند.
- جدول procs_priv مرتبط با رویه‌های ذخیره شده است. مجوز اعطا شده در سطح یک رویه، تنهـا و تنها به همان رویه خاص اعمال می‌شود.

سِرور از جداول user، db و host در مراحل کنترل دسترسی استفاده می‌نماید. ستون‌های جداول user و db را در ادامه مشاهده می‌نمایید.

نام جدول	user	db
ستون‌های حوزه	Host	Host
	User	Db
	Password	User
ستون‌های مجوز	Select_priv	Select_priv
	Insert_priv	Insert_priv
	Update_priv	Update_priv
	Delete_priv	Delete_priv
	Index_priv	Index_priv
	Alter_priv	Alter_priv

Create_priv	Create_priv	
Drop_priv	Drop_priv	
Grant_priv	Grant_priv	
Create_view_priv	Create_view_priv	
Show_view_priv	Show_view_priv	
Create_routine_priv	Create_routine_priv	
Alter_routine_priv	Alter_routine_priv	
Execute_priv	Execute_priv	
Trigger_priv*	Trigger_priv*	
Event_priv*	Event_priv*	
Create_tmp_table_priv	Create_tmp_table_priv	
Lock_tables_priv	Lock_tables_priv	
References_priv	References_priv	
	Reload_priv	
	Shutdown_priv	
	Process_priv	
	File_priv	
	Show_db_priv	
	Super_priv	
	Repl_slave_priv	
	Repl_client_priv	
	Create_user_priv	
	ssl_type	ستون‌های امنیتی
	ssl_cipher	
	x509_issuer	
	x509_subject	
	max_questions	ستون‌های کنترل

منبع		
	max_updates	
	max_connections	
	max_user_connections	

تذکر: ستونهای Event_priv و Trigger_priv در MySQL نسخه ۵,۱۶ اضافه شده‌اند.

در مرحله دوم کنترل دسترسی، سِرور مجوزهای کافی برای درخواست دسترسی را بررسی می‌نماید. علاوه بر جداول user، db و host سِرور ممکن است از جداول tables_priv و columns_priv نیز در مورد درخواست‌های خاص استفاده نماید. این جداول مجوزهای ریزدانه‌تری را در سطح جدول و ستون ارائه می‌نمایند که در جدول زیر نمایش داده شده است:

نام جدول	tables_priv	columns_priv
ستون‌های حوزه	Host	Host
	Db	Db
	User	User
	Table_name	Table_name
		Column_name
ستون‌های مجوز	Table_priv	Column_priv
	Column_priv	
ستون‌های دیگر	Timestamp	Grantor
	Grantor	

به منظور بررسی درخواست‌هایی که شامل رویه ذخیره شده است،سِرور از جدول procs_priv استفاده می‌کند. ستون‌های این جدول در جدول زیر نمایش داده شده‌اند:

نام جدول	procs_priv
ستون‌های حوزه	Host
	Db

User	
Routine_name	
Routine_type	
Proc_priv	ستون‌های مجوز
Timestamp	ستون‌های دیگر
Grantor	

ستون Routine_type یک ستون شمارشی با مقادیر FUNCTION و PROCEDURE به منظور مشخص کردن رویه‌ای است که سطر مربوطه به آن اشاره دارد. این ستون امکان تخصیص مجزای مجوزها را به یک تابع و یک رویه‌ی همنام فراهم می‌آورد. ستون‌های حوزه در جداول اعطا شامل انواع رشته‌ای هستند که در جدول زیر قابل مشاهده است:

نام ستون	نوع
Host	CHAR(60)
User	CHAR(16)
Password	CHAR(41)
Db	CHAR(64)
Table_name	CHAR(64)
Column_name	CHAR(64)
Routine_name	CHAR(64)

در فرآیند کنترل دسترسی، فیلدهای User، Password، Db و Table_name حساس به بزرگی و کوچکی حروف است. این در حالی است که فیلدهای Host، Column_name و Routine_name از این قاعده مستثنی هستند.

در جداول tables_priv، columns_priv و procs_priv ستون‌های مجوز به صورت مجموعه تعریف می‌شوند. مقادیر این ستون‌ها می‌تواند شامل هر ترکیبی از مجوزهای کنترل شده توسط این جدول باشند. این مقادیر در جدول زیر نمایش داده شده اند:

نام جدول	نام ستون	عناصر مجموعه
tables_priv	Table_priv	Select, Insert, Update, Delete, Create, Drop, Grant, References, Index, Alter, Create View, Show view, Trigger
tables_priv	Column_priv	Select, Insert, Update, References
columns_priv	Columns_priv	Select, Insert, Update, References
procs_priv	Proc_priv	Execute, Alter Routine, Grant

مجوزهای مدیریتی مانند RELOAD و SHUTDOWN، تنها در جدول user مشخص می‌شوند. اعمال مدیریتی، اعمال مختص سِرور بوده و خاص پایگاه داده نیستند. در نتیجه، نیازی به وجود این مجوزها در دیگر جداول اعطا وجود ندارد. به عبارت دیگر، سِرور برای تعیین امکان / عدم امکان انجام یک عمل تنها نیازمند استفاده از اطلاعات جدول user است. مجوز FILE نیز تنها در جدول user مشخص شده است. این مجوز یک مجوز مدیریتی به شمار نمی‌آید و امکان خواندن و نوشتن فایل‌ها روی میزبان به پایگاه داده مقصد بستگی ندارد.

سِرور mysqld محتویات جداول اعطا را در حافظه بارگذاری می‌نماید. با استفاده از دستور FLUSH PRIVILEGES می‌توان این جداول را مجددا بارگذاری نمود. پس از تغییر مجوزهای یک حساب می‌توان مجوزهای مربوطه را با استفاده از دستور SHOW GRANTS بررسی نمود. به عنوان مثال، به منظور مشخص نمودن مجوزهای اعطا شده به حساب با نام کاربری john و نام میزبان host1.example.com می‌توان از عبارت زیر استفاده نمود:

```
SHOW GRANTS FOR 'john'@'host1.example.com';
```

نحوه بررسی مجوزها:

همانطور که پیش از این یاد شد، کنترل دسترسی در MySQL در دو گام انجام می‌شود. در گام اول و در هنگام ارتباط با سِرور MySQL ، سِرور ارتباطات را بر مبنای شناسه و گذرواژه احراز هویت می‌نماید. در صورت موفقیت احراز هویت و برقراری ارتباط، سِرور وارد گام دوم کنترل دسترسی می‌شود. به ازای هر درخواست، سِرور درخواستی را بررسی کرده و با توجه به مجوزهای مجاز در زمینه رد یا اعطای حق دسترسی، تصمیم‌گیری می‌نماید. در این مرحله، از ستون مجوز در جداول اعطا استفاده می‌شود. این مجوزها می‌توانند از جداول user، db، host، tables_priv، columns_priv و procs_priv استخراج شود.

جدول user مجوزهایی که به صورت سراسری و بدون در نظر گرفتن یک پایگاه داده خاص تخصیص داده شده‌اند را مشخص می‌نماید. به عنوان مثال، اگر جدول user مجوز DELETE را به یک کاربر اعطا نماید، آن کاربر می‌تواند سطرهای دلخواه خود را از هر جدول موجود در پایگاه داده حذف نماید. در نتیجه، مجوزهای این جدول باید با دقت و به کاربران خاص، مثلا مدیر پایگاه داده، تخصیص یابد. برای دیگر کاربران نیز باید تمامی مجوزها در این جدول به 'N' مقداردهی شده و مجوزها در سطحی ریزدانه‌تر تخصیص یابد. مجوزها را می‌توان در سطوحی مانند پایگاه داده، جدول، ستون و رویه تخصیص داد.

جداول db و host مجوزهای خاص پایگاه داده را ذخیره‌سازی می‌نمایند. مقادیر ستون‌های حوزه در این جداول می‌تواند به صورت‌های زیر باشد:

- مقدار تهی برای نام کاربری در جدول db نشاندهنده‌ی یک کاربر بی‌نام است. یک مقدار غیر تهی باید دقیقا با نام کاربر مطابقت داشته باشد. استفاده از کاراکترهای ویژه برای نام‌های کاربری امکان‌پذیر نیست.
- از کاراکترهای ویژه '%' و '_' می‌توان در ستون‌های Host و Db استفاده کرد. به منظور استفاده از این کاراکترها در اعطای مجوزها باید آنها را با کاراکتر '\' به کار برد.
- استفاده از کاراکتر '%' در فیلد host در جدول db به معنای تمامی میزبان‌ها است. مقدار تهی در این فیلد به این معناست که باید از جدول host برای دریافت اطلاعات بیشتر کمک گرفت.
- استفاده از کاراکتر '%' و یا یک مقدار تهی در جدول host به معنای تمامی میزبان‌ها است.
- استفاده از کاراکتر '%' و یا یک مقدار تهی در جدول db به معنای تمامی پایگاه‌های داده است.

سِرور جداول db و host را در حافظه بارگذاری و مرتب‌سازی می‌نماید. به موازات این اعمال، جدول user نیز خوانده می‌شود. سِرور جدول db را بر اساس ستون‌های Host، Db و User مرتب‌سازی می‌نماید. علاوه بر این، جدول host بر اساس ستون‌های Host و Db مرتب می‌شود. همانند جدول user، مقادیر خاص‌تر در ابتدا و مقادیر عام‌تر در انتها قرار می‌گیرد. جداول tables_priv، columns_priv و procs_priv مجوزهای خاص جدول، خاص ستون و خاص رویه را اعطا می‌نماید. مقادیر ستون‌های این جداول می‌تواند به صورت‌های زیر باشد:

- کاراکترهای ویژه مانند '%' و '_' می‌تواند در ستون Host مورد استفاده قرار گیرد.
- مقدار '%' یا تهی برای میزبان به معنای تمامی میزبان‌ها است.

البته ستون‌های Db، Table_name، Column_name و Routine_name نمی‌تواند شامل کاراکترهای ویژه باشد. سِرور جداول tables_priv، columns_priv و procs_priv را بر اساس ستون‌های Host، Db و User مرتب می‌کند. این فرآیند مشابه فرآیند مرتب‌سازی جدول db است.

سِرور از این جداول مرتب شده به منظور تصمیم‌گیری در مورد درخواست دسترسی استفاده می‌نماید. برای درخواست‌های دسترسی که نیازمند مجوزهای مدیریتی مانند SHUTDOWN یا RELOAD هستند، سِرور تنها جدول user را بررسی می‌کند. در صورتی که این مجوز در سطر مربوطه در جدول user وجود داشته باشد با درخواست دسترسی موافقت می‌شود. در غیر اینصورت، درخواست دسترسی رد می‌شود.

برای درخواست مرتبط با پایگاه داده مانند INSERT، UPDATE سِرور در ابتدا جدول user را به منظور یافتن مجوزهای سراسری جستجو می‌کند. در صورت وجود چنین سطری در جدول با این درخواست موافقت می‌شود. در صورت عدم وجود مجوزهای سراسری، سِرور مجوزهای خاص پایگاه داده را با بررسی جداول db و host تعیین می‌نماید:

- سِرور در جدول db به دنبال یک مقدار خاص در ستون‌های Host، Db و User می‌گردد. ستون‌های Host و User با نام میزبان و نام کاربری MySQL تطبیق داده می‌شود. ستون Db با پایگاه داده‌های که درخواست دسترسی روی آن مطرح شده مطابقت می‌یابد. در صورت عدم وجود یک سطر برای Host و User با درخواست دسترسی مخالفت می‌شود.
- در صورتی که یک سطر همسان از جدول db پیدا شود و ستون Host نیز تهی نباشد، آن سطر مجوزهای ویژه پایگاه داده را تعریف می‌نماید.
- در صورتی که ستون Host تهی باشد در جدول host به جستجوی یک مقدار همسان در ستون‌های Host و Db پرداخته می‌شود. در صورتی که هیچ سطر همسانی در جدول host وجود نداشته باشد با درخواست دسترسی مخالفت می‌شود. در صورت وجود سطر همسان، مجوزهای خاص پایگاه داده برای یک کاربر بر اساس اشتراک مجوزهای جداول db و host محاسبه می‌گردد.

پس از تعیین مجوزهای خاص پایگاه داده، سِرور آنها را به مجوزهای سراسری جدول user اضافه می‌کند. در صورتی که بر اساس نتیجه به دست آمده عمل مورد نظر مجاز باشد، اجازه دسترسی صادر می‌شود. در غیر این صورت، سِرور مجوزهای جدول و ستون کاربر را بر اساس جداول tables_priv و columns_priv بررسی، آنها را به مجوزهای کاربر اضافه و برای درخواست دسترسی بر اساس نتیجه حاصل، تصمیم‌گیری می‌نماید. برای اعمال مربوط به رویه‌های ذخیره شده، سِرور از مجوزهای جدول procs_priv استفاده می‌نماید. نحوه محاسبه مجوزهای یک کاربر را می‌توان بر اساس عبارات منطقی، به صورت زیر خلاصه نمود:

- مجوزهای سراسری

- **یا** (مجوزهای پایگاه داده **و** مجوزهای میزبان)
- **یا** مجوزهای جدول
- **یا** مجوزهای ستون
- **یا** مجوزهای رویه

باید توجه داشت که در صورتی که یک دستور نیازمند بیش از یک مجوز باشد، مجوزهای دیگری غیر از مجوزهای موجود در جدول user به این جدول افزوده می‌شود. به عنوان مثال، به منظور اجرای دستور INSERT INTO ... SELECT مجوزهای INSERT و SELECT مورد نیاز است. در حقیقت، مجوزها ممکن است به گونه‌ای باشد که جدول user یک مجوز را اعطا نماید و جدول db نیز مجوز دیگری را اعطا کند. در این حالت، سِرور با بررسی تک تک جداول نمی‌تواند مجوزهای کافی برای انجام دستور را استخراج نماید. بدین منظور ترکیب مجوزهای این دو جدول ضروری می‌باشد.

لازم به ذکر است که در سمپاد MySQL می‌توان با استفاده از جدول host، میزبان‌های غیر امن را مشخص کرد. فرض کنید که ماشین public.your.domain در ناحیه‌ای عمومی و غیرامن در شبکه وجود دارد. می‌توان امکان دسترسی به تمامی میزبان‌های این شبکه را فراهم آورد و دسترسی به این میزبان خاص را محدود نمود. این امر به صورت زیر امکان پذیر است:

Host	Db	…
public.your.domain	%	(all privileges set to 'N')
%.your.domain	%	(all privileges set to 'N')

زمان تأثیر مجوزها:

در هنگام آغاز به کار mysqld محتوای تمامی جداول اعطا در حافظه بارگذاری می‌شود. در صورت تغییر جداول اعطا، به صورت غیر مستقیم و با استفاده از دستوراتی نظیر GRANT، REVOKE، SET PASSWORD، یا RENAME USER سِرور جداول اعطا را مجددا در حافظه بارگذاری می‌نماید. توجه داشته باشید که تأثیر تغییرات مستقیم جداول اعطا با استفاده از دستوراتی نظیر INSERT، UPDATE و DELETE تا شروع کار مجدد سِرور یا بارگذاری دستی مجدد آنها قابل مشاهده نیست. به منظور بارگذاری مجدد جداول اعطا می‌توان از دستوراتی نظیر FLUSH PRIVILEGES یا فرمان‌هایی نظیر mysqladmin flush-privileges یا mysqladmin-reload استفاده نمود.

بارگذاری مجدد جداول اعطا مجوزها را برای هر یک از ارتباطات کاربر به صورت زیر تغییر می‌دهد:

- مجوزهای جدول و ستون با درخواست بعدی کاربر تغییر می‌یابد.

- مجوزهای پایگاه داده با اجرای دستور USE db_name تغییر می‌یابد.
- مجوزهای سراسری و گذرواژه‌ها برای یک کاربرمرتبط، بدون تغییر هستند. تأثیر تغییرات در ارتباطات بعدی مشخص می‌شود.

در صورتی که سِرور با استفاده از گزینه skip-grant-tables-- سمپاد MySQL راه‌اندازی کرده باشد، جداول اعطا در حافظه بارگذاری نمی‌شود و همچنین کنترل دسترسی نیز اعمال نمی‌گردد.