ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta – Katedra informačních technologií

Bakalářská práce

Správa a zabezpečení přístupů MySQL

Vypracoval: **Holubovský Petr**

Vedoucí práce: **Ing. Havránek Martin**

Praha

2011 ©

Zadání BP

Čestné prohlášení

Čestně prohlašuji, že jsem bakalářskou práci na téma „Správa a zabezpečení přístupů MySQL“ vypracoval samostatně pouze za pomoci uvedených použitých zdrojů a po konzultacích s vedoucím bakalářské práce.

V Praze dne 30.3.2011

Podpis…….

Poděkování

Chtěl bych poděkovat svému vedoucímu bakalářské práce Ing. Martinovi Havránkovi za konzultace, dohled při tvorbě bakalářské práce a cenné rady.

Správa a zabezpečení přístupů MySQL

Souhrn

Tato bakalářská práce je zaměřena na rozbor správy přístupových oprávnění a zabezpečení přístupů k serveru MySQL. Práce obsahuje dvě základní části: teoretický rozbor zkoumané problematiky a rozpracování problematiky z praktického hlediska. V první části je rozebrán systém přístupových oprávnění a možnosti bezpečného připojení. Největší pozornost je věnována struktuře databáze mysql, do které se ukládají jednotlivá přístupová oprávnění. V druhé části práce jsou tyto teoretické poznatky demonstrovány na jednoduchém příkladu, prostřednictvím krátkých sekvencí kódů.

Klíčová slova

MySQL, přístupová oprávnění, grant, revoke, SSL, TCP/IP, my.cnf.

Summary

This bachelory essay is focused on the analysis of administrativ of access privileges and security access to the MySQL server. There are two basic parts included: the theoretical analysis of the research and development issues from the practical aspects. The first part has analyzed the system of access privileges and secure options for connection. The main attention is devoted to MySQL database, where the individual access privileges are stored. In the second part, there was these theoretical knowledge demonstrated for a simple example, through short sequences of code.

Key words

MySQL, access privileges,grant, revoke, SSL, TCP/IP, my.cnf

Obsah

Seznam tabulek

1 Úvod

Před tím než se začali počítače využívat pro vytváření a spravování databází, se dalo za databázi považovat třeba telefonní seznamy nebo kartotéku u lékaře. V dnešní době jsou však databáze integrovány do povědomí o počítačích a můžeme se s nimi setkat takřka kdekoliv. Od databází umístěných na pracovních stanicích, až po firemní databáze s podporou WWW sítě. Databáze mohou nabývat různých velikostí a podob, od těch, které jsou spravovány malými subjekty, až po ty, které spravují rozsáhlé společnosti.

Jedním z prostředků pro tvorbu a správu databáze je MySQL. MySQL je relační databázový systém. To znamená, že má podobu tabulek, které jsou vzájemně propojeny pomocí klíčů (sloupce v tabulce, které obsahují jedinečné, neopakující se záznamy).

Tato práce se bude zabývat správou a zabezpečením a přístupů k MySQL databázím. Dříve byly databáze umístěny pouze na stanici nebo lokální síti, proto se řešily pouze přístupová oprávnění k jednotlivým tabulkám. Zvláště nástup internetu nastolil nové požadavky na zabezpečený přístup k databázím prostřednictvím WWW sítě.

V první kapitole bude rozebrána problematika přístupových oprávnění. Možnosti nastavení, struktura databáze mysql, která se automaticky vytvoří při instalaci MySQL, do které se tato přístupová oprávnění zapisují a uchovává je. Pro přidělování a odebírání přístupových oprávnění je k dispozici několik možností. Buď prostřednictvím grafických nástrojů pro správu databází MySQL jako jsou například MySQL Query Browser a MySQL Administrátor. Nebo používáním standardních příkazů SQL. Jednotlivé výhody a úskalí budou vysvětlena v této kapitole. Dále budou rozebrány možnosti nastavení bezpečného připojení k databázím prostřednictvím SSL šifrování komunikace.

V druhé kapitole budou jednotlivé možnosti týkajících se problematik rozebíraných v kapitole první podrobněji popsány a demonstrovány pomocí krátkých ukázek příkazů.

V poslední, třetí kapitole bude poukázáno na to že, systém správy a zabezpečení MySQL postrádá některé prvky běžné z jiných systémů. Například zcela chybí možnost vytvářet skupiny uživatelů. Nebo jsou zde propracovány věci, které se dají řešit efektivněji na úrovni systému serveru MySQL.

2 Cíl práce a metodika práce

Cílem mé práce je informovat o možnostech zabezpečení serveru MySQL a jeho komunikaci s klientem tak, aby neproběhlo neoprávněné vniknutí do systému nebo znehodnocení, popřípadě i ztráta dat. Dále uvést možnosti správy serveru MySQL, a to především správy uživatelských účtů a jejich přístupových oprávněních.

Pro tvorbu práce byla použita odborná literatura a důvěryhodné internetové zdroje, související se zabezpečováním a správou serveru MySQL. Převážná část mé práce je tvořena literární rešerší z těchto zdrojů. V první části jsou obecněji vysvětleny a popsány základní okruhy týkající se správy a zabezpečení přístupů MySQL, jako jsou systém přístupových oprávnění, správa uživatelů a bezpečná připojení. V druhé části jsou tyto okruhy detailněji popsány spolu s krátkými ukázkami kódů pro práci se serverem MySQL.

3 Přehled řešené problematiky

Zabezpečení je prevence proti zcizení věcí a prostředků. Když odcházíte z bytu, tak uzamknete zámek, popřípadě zámky a zapnete alarm, když jej vlastníte. Protože víte, že pravděpodobnost krádeže je o mnoho větší, když necháte otevřeno. Zabezpečování věcí je automatická činnost člověka. Je proto ironií, že se zdá, že průmysl informačních technologií se chová ve značné míře naprosto opačně. V informačních systémech a aplikacích je mnoho sporných míst a děr, pomocí nichž se dají provést elektronické útoky. To vede ke ztrátě interních firemních dat nebo k jejich poškození. Mnohdy k tomu dochází ne proto, že by nebyly v informačních systémech možnosti pro zabezpečí, aby se takové věci nestávaly, ale proto, že tvůrce jednoduše nevyužil možnosti, které se mu nabízí (Gilmore, 2007,s. 575).

V této části práce budou rozebrány možnosti účinného zabezpečovacího modelu MySQL. Podrobně bude rozebrán systém uživatelských oprávnění MySQL, vytváření uživatelů, administrační oprávnění a změna hesla. Dále budou ukázány schopnosti bezpečného připojení MySQL, možnosti při nastavování limitů pro konzumaci prostředků.

3.1 Systém přístupových oprávnění MySQL

Ve většině případů je nežádoucí, aby všichni uživatelé měli neomezený přístup v celé databázi. Musí existovat alespoň jeden správce systému, který bude mít všechna práva. Ale bylo by velmi nebezpečné, kdyby bylo možno všemi uživately mazat nebo měnit záznamy v celé databázi, jak by si usmysleli (Kofler, 2007,s. 303).

„Existuje několik úrovní přístupových práv. V souvislosti s databází osobních údajů zaměstnanců je rozumné, aby zaměstnanci mohli číst jen určitou část databáze (aby třeba našli telefonní číslo jiného zaměstnance), ale zbytek jim zůstal skrytý (jako soukromá data).“ (Kofler, 2007,s. 303)

3.1.1 Systém oprávnění

Systém přístupových oprávnění MySQL se odehrává okolo dvou pojmů, autentizace (ověřování totožnosti) a autorizace.

- Autentizace : rozhoduje o tom, zda se povolí uživateli připojit k serveru.
- Autorizace : rozhoduje o tom, zda má uživatel dostatečná práva pro požadavky dotazů (Gilmore, 2007,s. 578).

„Protože k autorizaci nemůže dojít bez úspěšného ověření totožnosti, můžeme celý proces chápat jako dvoufázový.“ (Gilmore, 2007,s. 578)

Jak bylo uvedeno, systém řízení přístupových práv má dvě části. Ověření totožnosti při připojení a ověření platnosti požadavku zadaných uživatelem. Tyto dvě části se skládají z pěti oddělených kroků. (Gilmore, 2007,s. 578)

1. Jestli se má přicházející požadavek o připojení přijmout nebo odmítnout, určuje tabulka user. A to tak, že se testuje, jestli existuje souhlasný záznam v tabulce se specifikovaným hostitelem a uživatelem. Provedením tohoto kroku se dokončí ověřování totožnosti procesu řízení přístupových oprávnění. (Gilmore, 2007,s. 578)

2. Tímto druhým krokem se zahájí fáze prověřování platnosti požadavku procesu řízení přístupových oprávnění. Pokud byl požadavek o připojení k serveru přijat, zjišťuje se v tabulce user, zda má uživatel přidělena nějaká přístupová oprávnění. Pokud jsou zapnuta (nastaveny na y), potom má uživatel globální práva ke všem databázím, které oprávnění uděluje. Ve většině případů jsou tato oprávnění vypnuta, a přejde se ke kroku tři. (Gilmore, 2007,s. 578)

3. Projede se tabulka db, a zjistí se, s jakými databázemi může uživatel pracovat. Jakékoliv oprávnění, pokud je zapnuté, určuje, že uživatel má udělena tato oprávnění ke všem tabulkám dané databázi, se kterými může pracovat. (Gilmore, 2007,s. 578)

4. Jestliže se zjistí v tabulce db souhlasný uživatel, ale hostitel není uveden, projede se tabulka host. Pokud se zjistí souhlasný záznam hodnoty host, bude uživatel vlastnit k databázi taková práva, jaká jsou v tabulce host uvedena, a ne v tabulce db. To je proto, aby byl umožněn přístup k určité databázi určitému hostiteli. (Gilmore, 2007,s. 578)

5. Nakonec, jestliže uživatel se pokouší vykonat nějaký příkaz, ke kterému nemá přidělena oprávnění v tabulkách user, db nebo host, projedou se tabulky columns_priv a tebles_priv, aby se zjistilo, zda uživatel může vykonat daný příkaz nad sloupcem (sloupci) nebo tabulkou (tabulkami), kterých se daný příkaz týká. (Gilmore, 2007,s. 578)

Z rozboru těchto jednotlivých kroků se dá usoudit, že systém při ověřování oprávnění uživatele postupuje od těch nejširších k těm nejkonkrétnějším. (Gilmore, 2007,s. 578) Seznam MySQL oprávnění s jejich významem jsou uvedeny v tabulce 1.

MySQL oprávnění	Význam
	Pro přístup k tabulkám
Select	Smíte číst data (dotaz SELECT).
Insert	Smíte vkládat nové záznamy (INSERT).
Update	Smíte měnit záznamy (UPDATE).
Delete	Smíte mazat záznamy (DELETE).
Lock Tables	Smíte zamykat tabulky (LOCK TABLES).
	Změna databází, tabulek a pohledů
Create	Smíte vytvářet nové databáze a tabulky.
Create Temporary Table	Smíte vytvářet dočasné tabulky.
Alter	Smíte měnit názvy tabulek a jejich strukturu.
Index	Smíte přidávat a odstraňovat indexy.
References	Nepopsáno, pravděpodobně se může v budoucnu týkat vytváření vztahů mezi tabulkami.
Drop	Smíte odstraňovat databáze a tabulky.
Create Views	Smíte definovat pohledy (od verze MySQL 5.0).
Show Views	Smíte prohlížet definice pohledů příkazem SHOW CRETE VIEW (od verze MySQL 5.0).
	Pro uložené procedury (od verze MySQL 5.0)
Alter Routine	Smíte měnit uložené procedury.
Create Routine	Smíte definovat uložené procedury.
Execute	Smíte spustit uložené procedury.

	Pro přístup k datům
File	Smíte číst a měnit soubory v lokálním souborovém systému.
Create User	Smíte vytvářet nové MySQL uživatele (od verze MySQL 5.0.3).
	Pro správu MySQL
Grant Option	Smíte ostatním uživatelům udílet práva.
Show Databases	Smíte vidět záznamy všech databází (SHOW DATABASES).
Process	Smíte vidět MySQL procesy dalších uživatelů (SHOW PROCESSLIST)
Super	Smíte ukončovat MySQL procesy jiných uživatelů (KILL), vytvářet uložené procedury a triggery a měnit a spouštět některé příkazy pro správu (CHANGE/PURGE MASTER, SET GLOBAL).
Reload	Smíte spouštět různé příkazy (reload,reflesh, flush-xxx).
Replication Client	Smíte získat informace o účastnících systému replikace.
Replication Slave	Smíte číst údaje o serveru MySQL pomocí replikace.
Shutdown	Smíte ukončit MySQL.

Tabulka 1 - MySQL oprávnění (Kofler, 2007,s. 312).

3.1.2 Nastavení přístupových práv

K dispozici je několik způsobů, jak se dají přístupová práva nastavit.

- Nejschůdnější cestou jak nastavovat přístupová práva, je používání programu pro správu MySQL s grafickým uživatelským rozhraním. K dispozici je například MySQL Administrator a phpMyAdmin. Ale i to nejlepší grafické uživatelské rozhraní bude k ničemu, pakliže se nepochopí princip přístupu MySQL.

- MySQL databázi je možno přímo měnit pomocí příkazů UPDATE a INSERT.

- Dále je možné použít příkazy REVOKE a GRANT, nabízející další možnosti.

- Dále *„Může být použit skript jazyka Perl mysql_setpermission.pl.“* (Kofler, 2007,s. 303) . Použití tohoto skriptu je jednodušší, nežli použití příkazů REVOKE a GRANT, pochopitelně za předpokladu, že je nainstalován Perl. (Kofler, 2007,s. 303)

3.1.3 Uložení informací o přístupových oprávnění

V databázi MySQL je mnoho tabulek pro nejrůznější účely správy. Z toho sedm tabulek slouží pro správu přístupových oprávnění (Kofler, 2007,s. 315). Tyto tabulky se standardně nainstalují zároveň při instalaci databázového serveru (Gilmore, 2007,s. 579). Výčet těchto tabulek s jejich stručným významem je uveden v tabulce 2.

Název	Význam
user	Kontroluje, kdo (uživatelské jméno) se smí přihlásit k MySQL a z jakého počítače (jméno hostitele). Tato tabulka také obsahuje globální oprávnění.
db	Určuje, který uživatel smí přistupovat ke kterým databázím.
host	Rozšiřuje tabulku db informacemi o povolených jménech hostitele (které nejsou přítomné v db).
tables_priv	Určuje, kdo smí přistupovat k jakým tabulkám databáze.
columns_priv	Určuje, kdo smí přistupovat k jakým sloupcům tabulky.
func	Umožňuje správu UDF (uživatelem definované funkce), zatím není zdokumentovaná.
procs_priv	Určuje, kdo smí spouštět jednotlivé uložené procedury.

Tabulka 2 - Tabulky mysql pro správu přístupových práv (Kofler, 2007,s. 315).

3.1.3.1 Tabulka user

Tabulka user určuje, kdo se může přihlásit na server MySQL. V této tabulce se dají nastavovat globální oprávnění. Názvy jednotlivých oprávnění se liší dle skutečných názvů sloupců této tabulky. Například oprávnění CREATE TEMPORARY TABLE náleží sloupec Create_tmp_table (Kofler, 2007,s. 315). Dále je tato tabulka výjimečná v tom smyslu, že hraje roli v obou fázích ověřování požadavku. Ve fázi autentizace rozhoduje, zda připojovaný uživatel dostane přístup k serveru MySQL. V druhé fázi autorizace říká, jestli uživatel, který dostal přístup, má nějaká globální oprávnění (Gilmore, 2007,s. 580).

Dále se v tabulce ukládají oprávnění související s administrací serveru. Určuje, kteří uživatelé budou oprávněni zadávat příkazy, které přímo ovlivňují všeobecné fungování serveru. Například to je opětovné načtení přístupových oprávnění nebo shození serveru.

„Je třeba říci, že tabulka hraje hodně důležitou roli v celém postupu ověřování přístupových oprávnění.“ (Gilmore, 2007,s. 580)

Tabulka user je nejrozsáhlejší ze všech tabulek pro ověřování přístupových oprávnění, což vyplývá z jejího širokého pole působnosti. Má na třicet sloupců. Informace o struktuře, datových typů, názvů, atributů a výchozích hodnot jsou uvedeny v tabulce 3.

Sloupec	Datový typ	Povolené NULL	Výchozí hodnota
Host	varchar(60) binary	Ne	Není
User	varchar(16) binary	Ne	Není
Password	varchar(16) binary	Ne	Není
Select_priv	enum ('N','Y')	Ne	N
Insert_priv	enum ('N','Y')	Ne	N
Update_priv	enum ('N','Y')	Ne	N
Delete_priv	enum ('N','Y')	Ne	N
Create_priv	enum ('N','Y')	Ne	N
Drop_priv	enum ('N','Y')	Ne	N
Reload_priv	enum ('N','Y')	Ne	N
Shutdown_priv	enum ('N','Y')	Ne	N
Process_priv	enum ('N','Y')	Ne	N
File_priv	enum ('N','Y')	Ne	N
Grant_priv	enum ('N','Y')	Ne	N
References_priv	enum ('N','Y')	Ne	N
Index_priv	enum ('N','Y')	Ne	N
Alter_priv	enum ('N','Y')	Ne	N
Show_db_priv	enum ('N','Y')	Ne	N
Super_priv	enum ('N','Y')	Ne	N
Create tmp_table_priv	enum ('N','Y')	Ne	N
Lock_tables_priv	enum ('N','Y')	Ne	N
Execute_priv	enum ('N','Y')	Ne	N

Repl_slave_priv	enum (' N ' , ' Y ')	Ne	N
ssl_type	enum (' ' , ' A - NY ' , ' X509 ' , ' SPECIFIED ')	Ne	0
ssl_cipher	blob	Ne	0
x509_issuer	blob	Ne	0
x509_subject	blob	Ne	0
max_questions	int (11) unsignet	Ne	0
max_updates	int (11) unsignet	Ne	0
max_conections	int (11) unsignet	Ne	0

Tabulka 3 – struktura tabulky user (Gilmore, 2007,s. 581)

3.1.3.1.1 Kontrola přístupu

Pro kontrolu kdo se přihlašuje k serveru MySQL, musí být vyplněny tři identifikační sloupce (jméno hostitele, uživatelské jméno a heslo).

- *Uživatelské jméno:* Při kontrole uživatelského jména je systém case sensitive, což znamená, že rozlišuje velká a malá písmena. Součástí uživatelského jména nesmí být zástupné znaky, jako je například %. Pokud je uživatelské jméno nevyplněno, znamená to, že se může přihlásit kdokoliv jako host a v druhé fázi ověřování bude uživatelské jméno nahrazeno prázdným řetězcem.

- *Heslo:* Heslo je uloženo ve sloupci Password a je zašifrováno systémovou funkcí *PASSWORD*. „Heslo nemůže být uloženo jako čistý text. Navíc nejsou povoleny zástupné znaky.“ (Kofler, 2007,s. 317) Jestliže ve sloupci není nic uloženo, dá se přihlásit bez hesla. Neznamená to ovšem, že by se dalo přihlásit jakýmkoliv řetězcem. Doporučení je takové, že heslo pro přihlášení do MySQL by se nemělo shodovat s heslem pro přihlášení do operačního systému.

- *Jméno hostitele:* Pro zadání jména hostitele jsou dvě možnosti (IP adresa nebo jméno počítače). V tomto případě jsou povoleny zástupné znaky % (jakýkoliv

textový řetězec) a _ (jakýkoliv znak). Jestliže zůstane jméno hostitele nevyplněno, pak se může uživatel přihlásit odkudkoliv.

(Kofler, 2007,s. 318)

3.1.3.2 Tabulka db

Za pomoci této tabulky se uživatelům přidělují přístupová práva k jednotlivým databázím na databázovém serveru. Tabulka se projíždí v tom případě, když uživatel, který zadává požadavek, nemá globální práva k tomuto požadavku. *„Najde-li se v tabulce db v nějakém řádku trojice uživatel/hostitel/databáze, která povoluje vykonat kladený požadavek, požadavek se vykoná.“* (Gilmore, 2007,s. 584) V opačném případě nastane jedna ze dvou možností:

1. Pokud se najde souhlasný záznam dvojice uživatel/databáze, ale hostitel není vyplněný, projede MySQL tabulku host. Tabulka host bude vysvětlena v další části této práce.

2. Pokud se najde souhlasný záznam trojice uživatel/hostitel/databáze, ale potřebné oprávnění není povoleno (hodnota Y), projede MySQL tabulku tables_priv. Tabulka tables_priv bude vysvětlena v další části této práce.

Zástupné symboly (_ , %) mohou být použity ve sloupcích host a Db, ale nikoli ve sloupci user (Gilmore, 2007,s. 584). Informace o struktuře, datových typů, názvů, atributů a výchozích hodnot jsou uvedeny v tabulce 4.

Sloupec	Datový typ	Povolené NULL	Výchozí hodnota
Host	char (60)	Ne	Není
Db	char (64)	Ne	Není
User	char (16)	Ne	Není
Select_priv	enum (' N ' , ' Y ')	Ne	N
Insert_priv	enum (' N ' , ' Y ')	Ne	N
Update_priv	enum (' N ' , ' Y ')	Ne	N
Delete_priv	enum (' N ' , ' Y ')	Ne	N

Create_priv	enum (' N ' , ' Y ')	Ne	N
Drop_priv	enum (' N ' , ' Y ')	Ne	N
Grant_priv	enum (' N ' , ' Y ')	Ne	N
References_priv	enum (' N ' , ' Y ')	Ne	N
Index_priv	enum (' N ' , ' Y ')	Ne	N
Alter_priv	enum (' N ' , ' Y ')	Ne	N
Create_tmp_teble_priv	enum (' N ' , ' Y ')	Ne	N
Lock_tables_priv	enum (' N ' , ' Y ')	Ne	N

Tabulka 4 – struktura tabulky db (Gilmore, 2007,s. 585)

Pro rychlé provádění bezpečnostních kontrol má MySQL různé tabulky mysql nahrané a seřazené v paměti RAM. To s sebou, ale nese ten následek, že bude-li provedena nějaká změna v těchto nahraných tabulkách, musíme poté MySQL říci, že je musí znovu nahrát do paměti, aby byly aktuální. To se dělá za pomoci příkazu FLUSH PRIVILEGES a nebo v grafickém uživatelském prostředí mysqladmin za pomoci reload (Kofler, 2007,s. 323).

3.1.3.3 Tabulka host

Tabulka host se používá pouze tehdy, pokud je hodnota ve sloupci host tabulky db prázdná. Prázdná hodnota ve sloupci host tabulky db se ponechávají prázdné pouze tehdy, pokud konkrétní uživatel potřebuje mít přístup z více míst. Bylo by zbytečné mít několik souhlasných trojic v tabulce user. Proto je tato hodnota prázdná a jednotliví hostitelé jsou vypsáni v tabulce host ve sloupci host. Stejně jako v předchozích případech se zástupné symboly (_ , %) mohou používat ve sloupcích Host a Db, ale nikoli ve sloupci User (Gilmore, 2007,s. 585). *„MySQL používá tabulku host poměrně vzácně (většinou nastavení v db splní všechny požadavky). To je také dáno tím, že příkazy GRANT a REVOKE neovlivní tabulku host. Čili tabulka host je zpočátku prázdná.“* (Kofler, 2007,s. 324) Informace o struktuře datových typů, názvů, atributů a výchozích hodnot jsou uvedeny v tabulce 5.

Sloupec	Datový typ	Povolené NULL	Výchozí hodnota
Host	char (60)	Ne	Není
Db	char (64)	Ne	Není
Select_priv	enum (' N ' , ' Y ')	Ne	N
Insert_priv	enum (' N ' , ' Y ')	Ne	N
Update_priv	enum (' N ' , ' Y ')	Ne	N
Delete_priv	enum (' N ' , ' Y ')	Ne	N
Create_priv	enum (' N ' , ' Y ')	Ne	N
Drop_priv	enum (' N ' , ' Y ')	Ne	N
Grant_priv	enum (' N ' , ' Y ')	Ne	N
References_priv	enum (' N ' , ' Y ')	Ne	N
Index_priv	enum (' N ' , ' Y ')	Ne	N
Alter_priv	enum (' N ' , ' Y ')	Ne	N
Create_tmp_teble_priv	enum (' N ' , ' Y ')	Ne	N
Lock_tables_priv	enum (' N ' , ' Y ')	Ne	N

Tabulka 5 – struktura tabulky host (Gilmore, 2007,s. 586)

3.1.3.4 Tabulka tables_priv

V tabulce tables_priv jsou uložena oprávnění, která se týkají jednotlivých tabulek. Používá se pouze tehdy, pokud uživatel, který zadává požadavek, nemá dostatečná oprávnění vyplývající z tabulek user, db a host (Gilmore, 2007,s. 586). Informace o struktuře datových typů, názvů, atributů a výchozích hodnot jsou uvedeny v tabulce 6.

Sloupec	Datový typ	Povolené NULL	Výchozí hodnota
Host	char (60) binary	Ne	Není
Db	char (64) binary	Ne	Není
User	char (16) binary	Ne	Není
Table_name	char (60) binary	Ne	Není
Grantor	char (77)	Ne	Není
Timestamp	timestamp	Ano	Není
Table_priv	množina pro tabulku	Ne	Není
Column_priv	množina pro sloupec	Ne	Není

Tabulka 6 – struktura tabulky tables_priv (Gilmore, 2007,s. 587)

3.1.3.5 Tabulka columns_priv

V tabulce columns_priv jsou uložena oprávnění, která se týkají jednotlivých sloupců tabulky. Používá se pouze tehdy, pokud uživatel, který zadává požadavek, nemá dostatečná oprávnění vyplývající z tabulek user, db / host a tables_priv (Gilmore, 2007,s. 588). Informace o struktuře datových typů, názvů, atributů a výchozích hodnot jsou uvedeny v tabulce 7.

Sloupec	Datový typ	Povolené NULL	Výchozí hodnota
Host	char (60) binary	Ne	Není
Db	char (64) binary	Ne	Není
User	char (16) binary	Ne	Není
Table_name	char (60) binary	Ne	Není
Column_priv	char (64) binary	Ne	Není
Timestamp	timestamp	Ano	Není
Column_priv	množina pro sloupec	Ne	Není

Tabulka 7 – struktura tabulky columns_priv (Gilmore, 2007,s. 588)

3.1.3.6 Tabulka proc_priv

„Tato tabulka proc_priv je dostupná od verze MySQL 5.0.3.“ (Kofler, 2007,s. 326) Tato tabulka slouží uložení oprávnění pro práci s uloženými procedurami. Je nutná pouze pokud uživatel nemá tato oprávnění na globální úrovni, tedy uvedené v tabulce user nebo na úrovni tabulek, tedy uvedené v tabulce db. Nabízí tak možnost spuštění nebo změnu jen určitých procedur (Kofler, 2007,s. 326). Informace o struktuře datových typů, názvů, atributů a výchozích hodnot jsou uvedeny v tabulce 8.

Sloupec	Datový typ	Povolené NULL	Výchozí hodnota
Host	char (60) binary	Ne	Není
Db	char (64) binary	Ne	Není
User	char (16) binary	Ne	Není
Routine_name	char (64) binary	Ne	Není
Proc_priv	set(Execute,Alter Routine, Grant)	Ne	Není
Timestamp	timestamp(14)	Ano	Není
Grantor	char (77) binary	Ne	Není

Tabulka 8 – struktura tabulky proc_priv (Kofler, 2007,s. 326)

3.2 Správa uživatelů

Všechny tabulky, které jsou uloženy v databázi mysql mají stejnou strukturu, tedy relační. Znamená to, že jejich obsah se dá modifikovat obvyklými příkazy MySQL. Až do verze 3.22.11 se skutečně musely takto přesně spravovat informace, které byly uloženy v databázi mysql. S touto verzí však přišla inovace správy těchto klíčových dat. S pomocí těchto nových prvků lze přidávat nové nebo rušit staré (zapínat nebo vypínat) uživatele, přidělovat jim nebo jim odebírat oprávnění. „Jejich propracovaná syntax eliminuje případné možné katastrofální následky, které by mohly vzniknout vydáním špatně naformulovaného dotazu SQL (například kdyby uživatel zapomněl uvést v dotazu UPDATE klauzuli WHERE)“ (Gilmore, 2007,s. 588). Jak již bylo zmíněno, dříve se k obměně dat databáze mysql používaly běžné příkazy SQL jako například: UPDATE,

INSERT a DELETE. Toto je velmi únavné a pracné, proto je lepší používat příkazy GRANT a REVOKE (Kofler, 2007,s. 326).

Je důležité nezapomenout, že MySQL uchovává kopii tabulky mysql v paměti RAM, z důvodu zvýšení rychlosti. Při použití standardních příkazů SQL UPDATE, INSERT a DELETE se změny v tabulce mysql projeví, až po zadání příkazu SQL FLUSH PRIVILEGES nebo při použití externího programu mysqladmin reload. Ale při použití příkazů GRANT a REVOKE se tyto změny načtou do tabulky v paměti automaticky (Kofler, 2007,s. 326).

3.2.1 GRANT a REVOKE

Již zmíněná inovace se odehrává v režii dvou příkazů GRANT a REVOKE. Pomocí nich se přidávají noví a odebírají staří uživatelé, udělují a odnímají jejich přístupová práva. Jejich propracovaná syntax nám umožní do detailu určit, kdo bude moci dělat jakoukoliv operaci nad každou částí databáze. Kdo a odkud bude moci spravovat server nebo kdo bude moci a odkud měnit jakákoliv data v každém sloupci každé tabulky databáze (Gilmore, 2007,s. 588). V tabulce č. 9 jsou uvedena všechna přístupová práva, která lze pomocí příkazů GRANT a REVOKE udělit, respektive odvolat.

Název	Popis
All privileges	Týká se všech oprávnění kromě *WITH GRANT OPTION*.
Alter	Týká se používání příkazu *ALTER TABLE*.
Create	Týká se používání příkazu *CREATE TABLE*.
Create Temporary Tables	Týká se používání příkazu *CREATE TEMPORARY TABLE*.
Delete	Týká se používání příkazu *DELETE*.
Drop	Týká se používání příkazu *DROP TABLE*.
Execute	Zda bude uživatel moci spouštět uložené procedury MySQL verze 5.0 .
File	Týká se používání příkazu *SELECT INTO OUTFILE* a *LOAD DATA INFILE*.
Grant Option	Zda bude uživatel moci udělovat (delegovat) svá oprávnění.

Index	Týká se používání příkazu *CREATE INDES* a *DROP INDEX*.
Insert	Týká se používání příkazu *INSERT*.
Lock Tables	Týká se používání příkazu *LOCK TABLES*.
Process	Týká se používání příkazu *SHOW PROCESSLIST*.
References	Zástupce pro nějakou budoucí schopnost MySQL.
Reload	Týká se používání příkazu *FLUSH*.
Replication Client	Zda se bude uživatel moci dotazovat na umístění replik a řídícího vzoru.
Replication Slave	Požadované oprávnění pro práci s replikami.
Select	Týká se používání příkazu *SELECT*.
Show Databases	Týká se používání příkazu *SHOW DATABASES*.
Shutdown	Týká se používání příkazu *SHUTDOWN*.
Super	Týká se používání administračních příkazů, jako jsou *CHANGE MASTER*, *KILL* vlákno, *mysqladmin debug*, *PURGE MASTER LOGS* a *SET GLOBAL*.
Update	Týká se používání příkazu *UPDATE*.

Tabulka 9 - Práva spravující příkazy GRANT a REVOKE (Gilmore, 2007,s. 589).

3.2.2 Prohlížení oprávnění

Jestliže není jisté, jaká práva má jaký uživatel, je možné je nechat vypsat. MySQL v tomto případě nabízí dvě možnosti. Příkazy SHOW GRANS FOR a mysqlaccess (Gilmore, 2007,s. 594).

3.2.2.1 SHOW GRANTS FOR

Tento příkaz dává možnost zjistit přístupová oprávnění pro konkrétního uživatele. Zde je uveden příklad pro ilustraci.

SHOW GRANTS FOR petr@localhost;
Grants for petr@localhost :
GRANT SELECT ON mojeknihovna.* TO ´petr´@´localhost´ (Kofler, 2007,s. 328)

3.2.2.2 mysqlaccess

„Příkaz mysqlaccess je skript Perlu, kterým se mohou prohlížet souhrnné informace o přístupových oprávněných uživatelů MySQL.“ (Gilmore, 2007,s. 594) Tento skript nám i odhalí případná bezpečnostní rizika, která mohla vzniknout špatnou konfigurací přiřazených přístupových práv jednotlivému uživateli. Zde je uveden příklad pro ilustraci.

mysqlaccess 192.168.1.103 rita book –u root –p

(Gilmore, 2007,s. 594)

3.2.3 Grafické nástroje pro správu

Mezi grafické nástroje pro správu MySQL patří MySQL Administrátor a MySQL Query Browser. Tyto nástroje jsou podobné těm, které nabízejí i jiné databázové platformy.

3.2.3.1 MySQL Administrátor

MySQL Administrátor má široké spektrum využití ve správě a sledování databází. Dá se používat k mnoha úlohám, od nastavení parametrů serveru a jeho sledování, k zálohování dat, ke zjišťování výkonu a prohlížení protokolů. Prostřednictvím něj se dají nastavovat a sledovat oprávnění pro jednotlivé uživatele. Přináší vyšší komfort do správy databází. Jeho výstupy (zprávy a grafy) jsou vysoce nastavitelné a přizpůsobitelné, což je často přehlíženo těmi, co s ním začínají (Schneider, 2006, s. 51). *„Mnoho funkcí program nabízí:*

- *jen pro správce serveru,*
- *jen pro uživatele s dostatečným oprávněním (root),*
- *jen pro uživatele s dostatečným oprávněním operačního systému.“*

(Kofler, 2007,s. 110)

3.2.3.2 MySQL Query Browser

Předtím než přišel tento nástroj, byly silně omezeny možnosti, co se týče zobrazení výsledků dotazů na konzoly mysql nebo grafické nástroje třetích stran, jako Microsoft

Access nebo jiné nezávislé nástroje pro databázové dotazy. Změna ale přišla s tímto nástrojem, umožňuje provedení velkého množství databázových dotazů. Je to velmi šikovný nástroj pro ladění, umožňuje přehledné zobrazení výsledků a plánů dotazů (Schneider, 2006, s. 52). Neslouží však pouze pro práci s dotazy. Mezi další funkce patří:

- testování regulárních výrazů (pro SQL operátor REGEX)
- vkládání dat do tabulek
- změna dat v tabulkách
- čtení nápovědy a další….

(Kofler, 2007,s. 117)

3.3 Pohledy

Nejen omezení přístupu prostřednictvím přístupových oprávnění, ale i pohledy jsou v zabezpečení MySQL důležité. Jestliže se pečlivě nadefinuje pohled a přidělí se přístupová oprávnění uživatelům pro jejich přístup k těmto pohledům, ale ne k tabulkám se zdrojovými daty, pak se může efektivně omezit přístup uživatelům pouze na vybrané sloupce a řádky zdrojových tabulek. Pohledy nabízejí přesnou kontrolu nad tím, jaká data jsou viditelná pro jaké uživatele. (Groff, Weinberg, 2005, s. 408)

Pohledy mohou být buď pouze vybrané sloupce a řádky z jedné tabulky, nebo se dají definovat i složitější pomocí skupinového dotazu. Za pomoci skupinového dotazu se mohou uživateli nabídnout souhrnná data a nemusí mít přístup k podrobným řádkům zdrojové tabulky. Může se nakombinovat pohled ze dvou a více tabulek, a tak poskytnout uživateli přístup jen k těm datům, která potřebuje. A tak zamezit přístup k ostatním datům. Použití pohledů pro začlenění do zabezpečení MySQL je určeno dvěma omezeními (Groff, Weinberg, 2005, s. 408):

- *„Omezení aktualizací. Právo SELECT se používá u pohledů, jež mají být určeny pouze ke čtení, aby se omezilo načítání dat, práva INSERT, DELETE a UPDATE však jsou pro tyto pohledy bezvýznamná. Musí-li uživatel aktualizovat data*

viditelná v pohledu, který je určen pouze ke čtení, musí mít uživatel přístup ke zdrojovým tabulkám a musí používat příkazy INSERT, DELETE a UPDATE, jež se odkazují na tyto tabulky.“ (Groff, Weinberg, 2005, s. 409)

- *Výkon.* Protože přístup k pohledům je databází překládán na příslušející přístupy k tabulkám se zdrojovými daty, mohou pohledy znamenat velký nárůst pracnosti databázových operací. To snižuje výkonnost serveru. A proto pohledy nemohou být použity pro omezování přístupů uživatelů do databáze, aniž by to znamenalo pokles výkonu databáze. (Groff, Weinberg, 2005, s. 409)

3.4 Limity využívání prostředků uživateli

Kontrolovat využívání prostředků jednotlivými uživateli je více než vhodné, zvlášť pokud se nabízí MySQL v hostitelském prostředí, například úložné servery. Toto omezování je možné od verze 4.0.2.. Tato omezování (limity) jsou spravovatelné stejně jako přístupová práva. Existují tři oprávnění týkající se těchto prostředků a jsou umístěna v tabulce user.

- *max_connections* – udává maximální počet připojení uživatelem k databázi za hodinu
- *max_questions* – udává maximální počet výběrových dotazů (SELECT) vykonaných uživatelem za hodinu
- *max_updates* – udává maximální počet aktualizací (UPDATE a INSERT) vykonané uživatelem za hodinu

(Gilmore, 2007,s. 594)

3.5 Bezpečné ověření hesla

Server MySQL používá od verze MySQL 4.1 vylepšenou metodu šifrování hesel. Textový řetězec uložený ve sloupci password tabulky user byl zvětšen z původních 16 znaků na 41 znaků. Ve sloupci password není uloženo heslo jako čistý text (pravá podoba hesla), ale je uloženo v zašifrované podobě. Proto počet znaků neodpovídá heslu, ale jeho zašifrované podobě. V souvislosti s touto změnou byl změněn i ověřovací protokol. Dříve klientský

program přenášel k serveru heslo v podobě čistého textu. Nyní se heslo přenáší v zašifrované podobě (Kofler, 2007,s. 329).

Obě tyto změny zvýšily významně bezpečnost při ověřování MySQL. *„Avšak některé starší programy nemusí po aktualizaci serveru na novější verzi fungovat správně. Typickou chybovou hláškou bývá „Client does not support authentication protocol requested by server“*. (Kofler, 2007,s. 329) Proto v této kapitole bude popsáno několik možností, jak toto překonat.

3.5.1 Aktualizace knihoven klienta

Nejlepším způsobem je instalace nových verzí klientských knihoven. To ale může být větší problém, než se to na první pohled může zdát. Vlastní zásah do Perlu a instalací PHP v Linuxu může být dost komplikovaný, jestliže distributor nedodá vhodnou aktualizaci, což se stane málokdy. Jestliže změna distribuce není žádoucí, postačí někdy nainstalovat balík z jiné distribuce. Ale tento krok může mít za následek velké problémy se závislostí balíků, což ale vyžaduje dobrou znalost Linuxu. Další možností je komprimace daného programu. To ale pravděpodobně vyžaduje znalost Linuxu na profesionální úrovni (Kofler, 2007,s. 330).

3.5.2 Režim starých hesel

V případě,že aktualizace knihoven klienta není možná, tak se server MySQL dá spustit v režimu starých hesel. Tento krok vyžaduje pouze do konfiguračního souboru my.ini nebo my.cnf vložit možnost old-password v části mysqld. Po restartu se bude server chovat jako při verzi 4.0. To znamená že:

- Funkce PASSWORD bude šifrovat hesla starým algoritmem.
- Příkaz SQL GRANT bude používat starý šifrovací algoritmus.
- Server MySQL přijímá pro připojení starý ověřovací protokol.

Toto opatření se však týká pouze nově vytvořených hesel. Stará hesla (definovaná před aktualizací) se musí opětovně vytvořit (pokud se toto neudělá, server MySQL přejde na ověřování podle nového protokolu a ne na režim starých hesel) (Kofler, 2007,s. 330).

3.5.3 Současné použití starých a nových hesel

Režim starých hesel je nevýhodný v tom, že zvýšení bezpečnosti pomocí aktualizace na verzi 4.1 se neprojeví. Pakliže někteří uživatelé používají programovací jazyky umožňující použití nových ověřovacích protokolů, zatímco ostatní spoléhají na starý způsob ověřování hesel, potom by se mělo umožnit používání obou typů hesel. V takovém případě se nemusí dělat žádné změny v konfiguračních souborech my.ini a my.cnf. To, jaký ověřovací protokol použít, se MySQL rozhodne na základě délky zašifrovaného hesla. Pro hesla 16 znaků dlouhá se použije starý ověřovací protokol, pro ty delší se použije novější (Kofler, 2007,s. 330).

Jediné úskalí spočívá v tom, že správce si musí uvědomit, jaký uživatel preferuje starou metodu a kdo novou. Protože pro starou metodu se hesla definují jinak než pro novou. Hesla pro ty, kteří trvají na staré metodě, se definují pomocí OLD_PASSWORD(´secret´) nebo při použití mysqladmin old-password. Jestliže se definuje heslo pomocí příkazu GRANT, musí se nechat část IDENTIFIED BY nevyplněna a heslo se musí zadat manuálně (Kofler, 2007,s. 330).

3.6 Bezpečná připojení MySQL

Výměna dat mezi serverem MySQL a klientem je velmi podobná jakémukoli jinému přenosu dat po síti. Proto existuje potencionální nebezpečí, že tento provoz bude zachycen, nebo dokonce pozměněn nějakou třetí stranou, která nebude mít dobré úmysly. V mnoha případech se touto otázkou ani nemusí zabývat, protože server MySQL a klient jsou připojeni na uzavřené interní síti, nebo dokonce jsou oba na jednom počítači. Pokud ale požadavky a okolnosti směřují k tomu, že budou data přenášena přes nebezpečné komunikační kanály, měly by se využít zabudované bezpečnostní schopnosti MySQL, kterými lze připojení šifrovat. Je možné od verze 4.0.0 šifrovat veškerý provoz mezi klientem a démonem mysqld pomocí SSL (Secure Sockers Layer) a šifrovacího standardu X 509 (Gilmore, 2007,s. 594). Toto téma bude podrobněji rozebráno v další části této práce.

4 Správa a zabezpečení přístupů k DB MySQL

Pro zabezpečení serveru MySQL existuje řada zásad. Jednou z nejdůležitějších je, že by se služba mysqld neměla spouštět se systémovým uživatelem root. Hlavně proto, že když se spustí jako systémový uživatel root, mohou všichni uživatelé, kteří mají přístupové oprávnění FILE, pracovat se soubory souborového systému počítače, na kterém je server MySQL nainstalován. Proto službu mysqld spouštějte spíše jako jiný uživatel prostřednictvím třeba mysqladmin nebo mysql Query browser. To zajistí, že žádný uživatel se systémovými oprávněními nebude moci pracovat v souborovém systému hostitelského počítače. (Maslakowski, 2001, s. 309)

Jako další je zabezpečení podadresáře data. Ten je umístěn v domovském adresáři mysql. Všechny soubory, které se nacházejí v tomto podadresáři, by měly být ve vlastnictví uživatele, který spouští službu mysqld. Pouze tento uživatel by měl vlastnit přístupová oprávnění pro čtení a editaci souborů zde uložených. To zaručuje, že datové soubory MySQL nebude moci nikdo jiný měnit. Dále je vhodné omezovat uživatelské účty s hostitelským názvem '%'. A tam, kde je to možné, je definovat jako IP adresy. (Maslakowski, 2001, s. 309)

„Vždy, když se věc týká zabezpečení, měl by se na počátku nastavit co nejpřísnější režim. Teprve po nějaké době se mohou přísné zásady zmírnit.“ (Maslakowski, 2001, s. 310) Vždy, když se udělují přístupová oprávnění, by se měla přidělovat pouze taková, která uživatel nezbytně potřebuje pro svou práci. Čím vyšší bude zabezpečení databáze, tím bude méně pravděpodobný jakýkoliv útok na data v ní obsažená. (Maslakowski, 2001, s. 310)

V této části práce bude ukázána praktická stránka tohoto tématu. Budou uvedeny praktické příklady, jak dané operace provést. Tyto operace budou uvedeny na jednoduché databázi psi_utulek skládající se ze dvou tabulek psi (c_psa, jmeno_psa, rasa, stari, kotec, osetrovatel) a osetrovatele (rc, jmeno, prijmeni, plat)

4.1 Co udělat jako první

Zde bude ukázáno několik primárních, ale velmi nevyhnutelných úkolů, které je nutno udělat ihned po instalaci a konfiguraci.

- *Záplatování operačního systému a softwaru, který je nainstalován*. Výzvy k zabezpečení softwaru jsou takřka na denním pořádku. I když jsou tyto výzvy otravné, je nutné, kontrolovat systém, zda je důkladně zazáplatován. Do děravého systému se totiž dostane i uživatel, který s takovými věcmi nemá valné zkušenosti. Návodů na atak serveru je plný internet. Proto je nutné si vybrat jistou filozofii záplatování systému a tu dodržovat bez ohledu na to, jaký máte server. Nestačí se utěšovat, že pracujete v prostředí Unixu. I na této platformě se ukazují záplaty téměř stejně často jako u Windows (Gilmore, 2007,s. 576).

- *Vypnout všechny systémové služby, které se nepoužívají.* Neustále je potřeba apelovat na to, abyste vyloučili všechny možné cesty, kterými by se dal podniknout útok na server ještě před tím, než bude připojen na síť. Útoky jsou takřka výlučně vedeny přes nezabezpečené systémové služby. Často tyto služby běží na pozadí systému a administrátor o nich ani neví. Proto nejjednodušší pravidlo je: „co nepoužívám, tak vypnu“ (Gilmore, 2007,s. 576).

- *Zazáplatovat firewall databázového serveru*. I když vypnutí všech systémových služeb, které se nehodlají používat, je dobré zabezpečení, neuškodí, když se přidá ještě úroveň zabezpečení. Takovou další úrovní je zavření všech portů, které se nehodlají používat. „ U dedikovaného databázového serveru je obvyklé uzavření všech portů pod 1024, s výjimkou 22 (SSH), 3306 (MySQL), a utilitních portů, jako je 123 (NTP) (Gilmore, 2007,s. 576).

- *Revize uživatelských účtů databázového serveru*. *„ Konkrétně, jestliže byl pro udržování databáze dané organizace určen jako hostitel nějaký dříve existující server, přesvědčte se, že všichni neprivilegovaní uživatelé byli vypnuti nebo, což je lepší, odstraněni.“* (Gilmore, 2007,s. 576) I když uživatelé MySQL nemají s uživateli operačního systému žádný vztah, ale protože mají přístup do prostředí, existuje možnost, že poškodí databázový server nebo data v něm uložená, vědomě i

nevědomě. Pro absolutní jistotu, že se nic neopomene, je vhodné naformátovat všechny jednotky, které jsou připojeny a přeinstalovat operační systém (Gilmore, 2007,s. 576).

- *Nastavení hesla uživatele root MySQL*. Explicitně je ponecháno heslo uživatele root (administrátor) prázdné. „Přestože to považuji za problematické, dlouho už je to standardní postup a mám podezření, že to tak ještě nějakou dobu zůstane“ (Gilmore, 2007,s. 576). Heslo je více než vhodné nastavit ihned! Pro nastavení se dá použít příkaz *SET PASSWORD* .

  Heslo se volí samozřejmě složitější nežli secret. Nevýhodou u MySQL je, že přijme primitivní hesla jako například 1234 nebo jméno manželky. Jako dostačující se dá považovat heslo o 8 znacích. Skládající se z kombinace písmen (velká i malá), číslic a speciálních znaků (@,÷, atd.) (Gilmore, 2007,s. 577).

4.2 Přidělování a odebírání přístupových oprávnění MySQL

Struktura databáze mysql, ve které jsou uložena přístupová oprávnění je rozebrána v předchozí části práce a je zobrazena v tabulkách 2 – 8. Práva, která lze přidělovat a odebírat, jsou zobrazena v tabulce 1. Nyní bude demonstrováno, jak tato práva přidělit nebo odebrat. Bude také rozebrána filozofie přístupových oprávnění.

Uživatelům by měla být udělována pouze taková přístupová oprávnění, která opravdu potřebují. Pokud uživatel potřebuje pracovat jen s jednou databází, není vhodné mu nastavovat příslušná přístupová oprávnění v tabulce user. Protože, kdyby se to udělalo, měl by poté takto nastavená přístupová oprávnění ke všem databázím umístěných na příslušném MySQL serveru. Kdyby nějaký uživatel potřeboval prohlížet nějakou celou databázi (všechny tabulky i sloupce v ní obsažené), ale upravovat by směl pouze jediný sloupec. Potom by se nastavilo v tabulce db přístupové oprávnění Select_priv k té příslušně databázi a v tabulce columns_priv přístupové oprávnění UPDATE ve sloupci Columns_priv pro příslušný sloupec. (http://aleph.techlib.cz/, 9.2.2011)

4.2.1 Přidělování přístupových oprávnění MySQL

Pro přidělování přístupových oprávnění jsou dvě možnosti:

- Použití standardních příkazů pro práci s tabulkami (INSERT INTO, UPDATE, atd). Když se ale použije tento způsob, musí se po každém zásahu do tabulek s přístupovými oprávněními zadat ještě příkaz FLUSH PRIVILEGES pro opětovné načtení tabulek do paměti. Proč je to tak, je vysvětleno v předchozí kapitole.

- Použití příkazu GRANT, oprávnění, která lze pomocí tohoto příkazu udělit, jsou popsána v tabulce 9.

Po úspěšné instalaci se automaticky vytvoří tabulky přístupových oprávnění, do kterých je automaticky zapsán uživatel root. Tento uživatel má neomezená přístupová oprávnění na úplně vše, co se dá dělat na serveru MySQL. Po tomto zápisu má ale prázdné heslo, takže se jako root může přihlásit kdokoliv. Proto vložení hesla uživatele root by měl být vždy první krok. Provede se následovně: v příkazovém řádku počítače, kde běží MySQL, se spustí konzole mysql. Spustitelný soubor je uložen v podadresáři /bin a jmenuje se mysql.exe pro Windows a mysql pro Linux. Nastavení hesla pro uživatele root:

shell > mysql –u root –p (přihlášení uživatele root)
mysql > \u mysql (pro práci bude použita databáze mysql)
mysql > UPDATE user SET Password=PASSWORD('root_heslo') WHERE user='root'
(zápis hesla root do tabulky user)
mysql > FLUSH PRIVILEGES (znovunačtení přístupových oprávnění do paměti)
mysql > \q (konec práce)

(http://interval.cz, 9.2.2011)

Po tomto kroku má již uživatel root nastaveno heslo. Nyní bude vytvořeno několik nových uživatelů a budou jim přidělena přístupová práva. V prvním případě to bude pomocí příkazu GRANT a v druhém pomocí přímého přístupu do databáze mysql (tabulky s přístupovými oprávněními).

shell > mysql –u root –p (přihlášení uživatele root)
Enter password: (server vyžaduje heslo uživatele)

mysql > \u mysql (pro práci bude použita databáze mysql)
mysql > GRANT ALL PRIVILEGES ON psi_utulek.* TO lukas@localhost IDENTIFIED BY 'lukasovo_heslo' WITH GRANT OPTION;
mysql > GRANT SELECT , RELOAD ON psi_utulek.* honza@“%“ IDENTIFIED BY 'honzovo_heslo';
mysql > \q (konec práce)

(http://interval.cz, 9.2.2011)

Jako první byl vytvořen uživatel ‚lukas‘ s heslem ‚lukasovo_heslo‘. Lukas má povolený přístup pouze z počítače, na kterém běží SQL server (@localhost). Přidělena mu byla přístupová oprávnění k databázi psi_utulek a ke všem sloupcům v ní obsažených (ALL PRIVILEGES). Jako druhý byl vytvořen uživatel ‚honza‘ s heslem ‚honzovo_heslo‘. Honza se může připojit z kteréhokoliv počítače na celém světě (@“%“). Má právo select nad všemi tabulkami databáze psi_utulek a k administraci může spustit RELOAD. (http://interval.cz, 9.2.2011)

Nyní budou vytvořeni ti samí uživatelé, pouze s tím rozdílem, že bude využit přímý přístup do databáze mysql (tabulky přístupových oprávnění).

mysql > INSERT INTO user VALUES('localhost', 'lukas', PASSWORD('lukasovo_heslo'), 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y');
mysql > INSERT INTO user (Host, User, Password, Reload_priv) VALUES('%', 'honza', PASSWORD('honzovo_heslo'), 'Y');
mysql > INSERT INTO db (Host, Db, User, Select_priv) VALUES('%', 'psi_utulek', 'honza', 'Y',)
mysql > FLUSH PRIVILEGES

(http://interval.cz, 9.2.2011)

4.2.2 Odebírání přístupových oprávnění MySQL

Při odebírání přístupových oprávnění jsou k dispozici podobné možnosti jako při jejich udělování. Opět jsou dvě možnosti:

- Použití standardního příkazu pro práci s tabulkami UPDATE. Opět platí, že po použití tohoto příkazu se musí použít příkaz pro opětovné načtení tabulek s přístupovými oprávněními FLUSH PRIVILEGES.
- Použití příkazu REVOKE, oprávnění, která lze pomocí tohoto příkazu odebrat, jsou popsána v tabulce 9.

Pro interpretaci budou uvedeny dva příklady. Jeden s přímým přístupem do databáze mysql a druhý na ukázku příkazu REVOKE. Oba budou provádět to samé.

mysql > UPDATE user SET File_priv = 'N' WHERE User = 'lukas'; (1)
mysql > FLUSH PRIVILEGES; (1)
mysql > REVOKE FILE ON *.* FROM lukas@localhost; (2)

(http://interval.cz, 9.2.2011)

Příkazem (2) by se ale odebrala práva pouze v případě, že se uživatel připojoval pouze z počítače, na kterém běží MySQL server. Kdyby se mohl přihlašovat ještě ze vzdáleného počítače, musel by se ještě zadat příkaz (http://aleph.techlib.cz/, 9.2.2011) :

mysql > REVOKE FILE ON *.* FROM lukas@%;

4.2.3 Odstranění uživatele

Uživatele lze odstranit pouze za pomoci dotazu SQL nad tabulkami uživatelských účtů. Uživatele lze odstranit takto:

DELETE FROM user WHERE User = “lukas“ AND Host = “%“;
DELETE FROM db WHERE User = “lukas“ AND Host = “%“;

Atd. Samozřejmě, že se musí odstranit všechny záznamy, které příslušejí uživateli, kterého chceme zrušit. A to ve všech tabulkách. (http://aleph.techlib.cz/, 9.2.2011)

Pokud se ale počítá s tím, že uživatel bude znovu oživen, budou mu opět přidělena některá přístupová oprávnění. Nemusí se uživatel takto rušit, ale stačí mu pouze odebrat všechna práva:

mysql > REVOKE ALL PRIVILEGES ON lukas.* FROM lukas@localhost;

Při opětovném udělování přístupových oprávnění tak odpadne znovuzakládání uživatele. (Gilmore, 2007,s. 593)

4.3 Vytvoření pohledu

Vytvořením pohledu se dá vyhnout složitějšímu nastavování práv v tabulkách přístupových oprávnění, protože se nebude muset definovat, ke kterým sloupcům budou mít uživatelé přístup a ke kterým ne a rovnou se přiřadí příslušná práva k pohledu. Jako příklad bude uvedeno definování pohledu osetruje, ke kterému bude mít přístup kdokoliv s právem Select. Pro tento účel bude vytvořen uživatel, který bude moct prohlížet pouze tento pohled a nic jiného. Mohl by sloužit pro zájemce o psy, kteří by se chtěli ujmout daného psa.
Vytvoření pohledu:

mysql > CREATE WIEW osetruje AS SELECT jmeno, prijmeni, jmeno_psa, rasa, stari FROM osetrovatele, psi;

Vytvoření uživatele:
mysql > INSERT INTO user (Host, User,) VALUES('%', 'kdokoli',);
mysql > INSERT INTO tables_priv (Host, Db, User, Table_name, Grantor, Table_priv, Column_priv) VALUES('%', 'psi_utulek', 'kdokoli', 'osetruje', 'root', 'osetruje',ALL)
mysql > FLUSH PRIVILEGES

4.4 Bezpečné připojení MySQL

Pro zjištění, zda MySQL může zpracovávat bezpečná připojení, se přihlásí k serveru MySQL a zadá se příkaz:
mysql > SHOW VARIABLES LIKE ’have_openssl‘;
Pro používání bezpečného připojení ssl je nutné buď vytvořit nebo zakoupit certifikát, jak pro klienta, tak pro server. (Gilmore, 2007,s. 584)

4.4.1 Volby příkazu GRANT týkající se bezpečného připojení

Příkaz GRANT má několik voleb, které definují požadavky serveru na uživatele týkající se SSL. Následuje jejich výčet:

- *Require ssl* – přinutí uživatele se připojovat přes ssl. Pokud se pokusí připojit jinak než přes ssl (nezabezpečeným způsobem), pak skončí tento pokus chybou "Access denied" Přístup odmítnut. Příklad:
  mysql > grant insert, select, update on psi_utulek. to lukas@client.ochranci.cz*
  > identified by 'lukasovo_heslo' require ssl;
  require x509
  Tato volba donutí uživatele, aby předložil platný certifikát CA (Certificate Autority). To je požadováno, pokud se chce prověřit signatura certifikátu certifikátem CA. Toto ale nezpůsobí zájem MySQL o předmět, vydavatele nebo původ certifikátu. Respektive neurčuje, které CA jsou platné a které ne. Za platný bude určen jakýkoliv certifikát CA, který byl prověřen. (Gilmore, 2007,s. 596)

- *Require issuer* – tato volba donutí uživatele předložit platný certifikát CA vydaný platným vydavatelem. Proto je nutné doplnit několik informací, jako země původu, stát původu, město původu, jméno vlastníka a jeho kontakt. (Gilmore, 2007,s. 596)
  Přiklad:
  mysql > grant insert, select, update on psi_utulek. to lukas@client.ochranci.cz*
  > identified by 'lukasovo_heslo' require ssl require issuer 'C=EU, ST=CR,
  >L=Praha, O=LKOCOUREK, OU=ADMIN,
  >CN=db.ochranci.cz/Email=lukas@ochranci.cz'

- *Require subject* – tato volba donutí uživatele předložit platný certifikát CA včetně platného předmětu certifikátu. (Gilmore, 2007,s. 597) Příklad:
  mysql > grant insert, select, update on psi_utulek. to lukas@client.ochranci.cz*
  > identified by 'lukasovo_heslo' require ssl require subject 'C=EU, ST=CR,
  >L=Praha, O=LKOCOUREK, OU=ADMIN,
  >CN=db.ochranci.cz/Email=lukas@ochranci.cz'

- *Require cipher* – tato volba přinutí uživatele, aby použil nejnovější šifrovací algoritmus, protože ho donutí se připojit prostřednictvím konkrétní šifry.

V současnosti jsou k dispozici: RSA, DES, EDH, SHA a CBC3. (Gilmore, 2007,s. 597) Příklad:

mysql > grant insert, select, update on psi_utulek. to lukas@client.ochranci.cz > identified by 'lukasovo_heslo' require ssl require cipher 'EDH-SHA';*

4.4.2 Volby SSL

Volby uvedené v této kapitole používá jak připojující se klient tak server, na který se připojuje. Rozhodují o použití SSL (ano X ne), a pakliže ano, tak určují umístění certifikátu a souborů klíčů. (Gilmore, 2007,s. 597)

- --*ssl* – tato volba sama o sobě říká, že se má použít SSL. Jestliže se použije v součinnosti s mysqld, tak tato volba sděluje serveru, že má povolit připojení prostřednictvím SSL. Jestliže se použije v součinnosti s klientem, tak to signalizuje serveru, že klient použije k připojení SSL. Použitím této volby se ale nezajistí ani nepožaduje bezpečné připojení SSL. *„Skutečně, mé vlastní testy ukázaly, že se volba sama o sobě ani nepožaduje, chcete-li iniciovat připojení SSL. Zda bude úspěšně iniciováno bezpečné připojení SSL, určují přepínače, které je doprovázejí.“* (Gilmore, 2007,s. 597) Dále budou popsány tyto přepínače.

- --*ssl-ca* – tato volba říká, v jakém souboru a kde se nachází seznam důvěryhodných certifikačních autorit SSL. (Gilmore, 2007,s. 597) Příklad:
  --*ssl-ca=/home/lukas/openssl/cacert.pem*

- --*ssl-capath* – tato volba ukazuje cestu do adresáře, v němž jsou uloženy důvěryhodné certifikáty SSL ve formátu PEM (privacy-enhanced mail) (Gilmore, 2007,s. 597)

- --*ssl-cert* - tato volba říká, v jakém souboru a kde se nachází certifikát SSL, který byl použit při zřízení bezpečného připojení. (Gilmore, 2007,s. 597) Příklad:
  --*ssl-cert=/home/lukas/openssl/mysql-cert.pem*

- --*ssl-cipher* – tato volba určuje, jaké šifrovací algoritmy jsou povoleny. Seznam těchto šifer se shoduje se seznamem, který používá příkaz: *%>openssl ciphers.* Kdyby se například chtěly povolit jen šifrovací algoritmy Triple-DES a Blowfish,

nastaví se volba takto:
--ssl-cipher=des3:bf (Gilmore, 2007,s. 597)

- --ssl-key – tato volba určuje název a umístění klíče SSL, který byl použit při zřízení bezpečného připojení. (Gilmore, 2007,s. 597) Příklad:
  --ssl-key=/home/lukas/openssl/mysql-key.pem

4.4.3 Nastartování serveru MySQL se zapnutou podporou SSL

Když už jsou k dispozici certifikáty klienta i serveru, tak se může server MySQL nastartovat se zapnutou podporou SSL. Příklad:
„%>./bin/mysqld_safe –user=mysql –ssl-ca=$OPENSSL/cacert.pem
>--ssl-cert=$OPENSSL/server-cert.pem –ssl-key=$OPENSSL/server-key.pem“ (Gilmore, 2007,s. 597), kde $OPENSSL odkazuje na umístění certifikátu SSL. (Gilmore, 2007,s. 597)

4.4.4 Připojení pomocí klienta se zapnutým SSL

Připojit se k serveru MySQL s podporou SSL můžete takto:
%>mysql –ssl-ca=$OPENSSL/cacert.pem –ssl-cert=$OPENSSL/client-cert.pem
>--ssl-key=$OPENSSL/client-key.pem –u lukas –h www.ochranari.cz –p, kde $OPENSSL odkazuje na umístění certifikátu SSL. (Gilmore, 2007,s. 597)

4.4.5 Uložení voleb SSL do souboru my.cnf

Samozřejmě, že se volby SSL nemusí předávat prostřednictvím příkazového řádku. Mohou se uložit do souboru my.cnf. Ukázka souboru my.cnf:
[client]
ssl-ca = /home/lukas/openssl/cacert.pem
ssl-cert = /home/lukas/openssl/client-cert.pem
ssl-key = /hole/lukas/openssl/client-key.pem
[mysqld]
ssl-ca = /usr/local/mysql/openssl/ca.pem
ssl-cert = /usr/local/mysql/openssl/cert.pem
ssl-key = /usr/local/mysql/openssl/key.pem
(Gilmore, 2007,s. 597)

4.5 Problémy s připojováním

Jestliže nastanou problémy s připojením k serveru MySQL, bez ohledu na programovací jazyk, který se používá, mělo by se jako první otestovat programem mysql, zda se lze vůbec připojit k serveru MySQL. Jen pokud se podaří tímto programem připojit, mohou se začít hledat příčiny v programu. (Kofler, 2007,s. 330)

Změny v tabulkách přístupových oprávnění databáze mysql se projeví až po zadání příkazu FLUSH PRIVILEGES (nebo restartováním serveru MySQL), pokud se nepoužije příkazů GRANT a REVOKE. Mnoho programů, které se využívají pro správu MySQL, příkaz FLUSH PRIVILEGES nespouští automaticky. Pokud se tedy změny v tabulkách přístupových oprávnění neprojeví, musí se do mysql přihlásit jako root a zadat příkaz FLUSH PRIVILEGES (nebo při použití mysqladmin flush-privileges). (Kofler, 2007,s. 331)

Pokud se zadal příkaz FLUSH PRIVILEGES (přihlášen jako uživatel root), i přesto se změny mohou projevit až při příštím přihlášení. *„Platí následující pravidla:*

- *Změny globálních oprávnění se projeví až při dalším přihlášení.*
- *Změny databázových oprávnění se projeví po použití příkazu USE database.*
- *Změny oprávnění tabulek a sloupců se projeví provedením prvního příkazu SQL.“*

(Kofler, 2007,s. 331)

4.5.1 Možné příčiny problémů s připojením

V následujícím výčtu jsou uvedeny obvyklé příčiny problémů s připojováním. Jedno chybové hlášení nemusí být způsobeno pouze jednou příčinou.

- *Server MySQL neběží* – Pokud při pokusu o připojení je obdrženo chybové hlášení 2002 (Can´t connect to local MySQL server through societ /var/lib/mysql/mysql.sock.) nebo 2003 (Can´t connect to local MySQL server on 'hostname'). V případě, že server MySQL je nainstalován na Windows, snadno se zjistí, jestli server běží, a to pohledem na správce úloh. Jestliže je na Linuxu, může se zadat příkaz ps | grep –i mysql. Po tomto příkazu by se měl zobrazit seznam

s procesy (protože server MySQL běží jako více procesů, pro zvýšení efektivity). Pokud se nezobrazí, musí server spustit (na Linuxu příkazem /tec/init.d/mysql[d]) start. (Kofler, 2007,s. 331)

- *Klientský program nenašel socked soubor* – Tento případ je pouze pro Unix/Linux, protože klient se serverem (pokud běží na stejném počítači) komunikuje prostřednictvím socked souboru. Proto musí oba programy (klient i server) znát umístění tohoto souboru. Pokud se vyskytnou potíže, je dobré zkontrolovat, jestli nastavení socked=aktuální_cesta_k_souboru se nachází v konfiguračním souboru /tec/my.cnf v sekci [client]. Obvyklá cesta je /var/lib/mysql/mysql.sock. (Kofler, 2007,s. 331)

- *Klientský program nemůže přistupovat k socked souboru (SELinux) – „Spousta distribucí Linuxu (například RHEL 4) je nastavena tak, že Apache nemůže přistupovat k souborům mimo adresář htdocs. Tudíž nemůže ve skriptech PHP a Perl používat socked soubor. Nastane dříve zmíněná chyba 2002. Možnost nápravy – komunikace přes TCP/IP nebo deaktivování SELinux pro Apache (pod RHEL pomocí systém-config-security).“* (Kofler, 2007,s. 331)

- *Nefunguje síťové spojení mezi serverem a klientem* – Pokud server a klient jsou spuštěny na různých počítačích, zadá se na počítači, na kterém běží klient, příkaz *ping nazevpocitace-serveru*. Tím se otestuje dostupnost serveru. Pokud není dostupný, musí se přednastavit síť. (Kofler, 2007,s. 331)

- *MySQL nepřijímá pokus o připojení přes síť (TCP/IP)* – Toto může mít na svědomí nastavení *skip-networking* v konfiguračním souboru my.cnf. MySQL volí tento druh nastavení kvůli zvyšování bezpečnosti. To má ale za následek, že se klient může připojit pouze z lokální sítě. Toto se většinou projeví již zmíněnou chybou 2003. Pro odstranění této chyby se musí odstranit příslušné nastavení ze souboru my.cnf. (Kofler, 2007,s. 331)

- *MySQL nepřijímá připojení z našeho počítače* – Tento problém se vyskytne v případě, že na počítači poskytovatele internetového připojení běží i MySQL server. Kde je pravděpodobně nastaveno, že přístup je možný jen z lokální sítě

(nebo lokálního počítače). Pro správu databází se musí použít nějaký program, který je nainstalovaný na serveru a je dostupný z internetu (phpMyAdmin nebo další). (Kofler, 2007,s. 332)

- *Rozlišování názvu počítače ze jména hostitele probíhá špatně* – Při připojování (přes síť) vyskočí chybová hláška 1130 (Host 'n.n.n.n' is not allowed to connect to this MySQL server). Nejčastěji je způsobeno špatnou hodnotou hostitele v tabulce mysql.user, nebo je špatně nastavený server jmen. Podle toho jak je síť nastavena, musí být ve sloupci user.Host uložena hostitelská jména s doménovým jménem nebo bez něj (méně časté). Řešení je doménové jméno přidat nebo odebrat od názvu hostitele. Pokud toto nepomůže, nabízejí se příkazy host, hostname a resolveip k otestování, zda je vůbec problém s rozlišováním názvů. Poslední možností je vložit za název hostitele IP adresu, to je ale málo flexibilní. (Kofler, 2007,s. 332)

- *Nesprávné uživatelské jméno nebo heslo* – Musí se dávat pozor na překlepy! Nesmí se zapomínat na to, že musí také souhlasit název hostitele. Často je přístup omezen pouze z některých počítačů. Tento problém může také souviset s tím předcházejícím. (Kofler, 2007,s. 332)

- *Klientský program používá starou MySQL knihovnu pro ověřování hesla* – Pokud vyskočí chybová hláška *Client does not support authentication protocol requested by server,* je to důsledek pokusu se připojit k serveru verze 4.1 a vyšší, kdežto klientský program je určen pro verze 4.0 a nižší. Ve verzi 4.1 a vyšší je vylepšený algoritmus pro šifrování hesel. Ten bohužel není kompatibilní s tím z nižších verzí. Jsou dvě možnosti řešení. Za prvé aktualizovat klientský program a jeho knihovny, nebo za druhé spustit server MySQL v režimu starých hesel. Tato možnost je rozebrána v kapitole 3.6.2. (Kofler, 2007,s. 332)

- *Špatný záznam v tabulce mysql*.user – Při přihlašování uživatele u z počítače p se nejdříve porovná záznam s jedinečnou hodnotou ve sloupci Host, a až poté přijdou na řadu hodnoty se zástupnými znaky. Tento algoritmus platí u porovnání hodnot ve sloupci User. Důsledkem toho je, že se uživatel lukas nemůže přihlásit z počítače localhost, i když v tabulce mysql.user existuje záznam

Host='%'/User='lukas'. A to protože primární nastavení přístupových oprávnění je záznam Host='localhost'/User=''. Toto nastavení má přednost před sekundárními záznamy. Řešením je buď přidat záznam Host='localhost'/User='lukas' do tabulky mysql.user, nebo odstranit záznam Host='localhost'/User=''. Lepší je druhé řešení, protože záznam User='' představuje poměrně velké bezpečnostní riziko (na localhost se může přihlásit kdokoliv). (Kofler, 2007,s. 332)

- *Nebylo zadáno uživatelské jméno* – Pokud se nezadá přihlašovací jméno při přihlašování na server MySQL, tak se automaticky použije jméno účtu programu, pod kterým se přihlašujeme. Z bezpečnostních důvodů webové servery neběží na uživatelských účtech Administrator (Windows) nebo root (Linux), ale jako wwwrun nebo apache. Tyto záznamy nejsou v tabulce mysql.user, proto je nám odepřen přístup. Nesmí se proto zapomínat dávat do přihlašovacího skriptu přihlašovací jméno. (Kofler, 2007,s. 333)

- *Připojit se podaří, ale nedostaneme se k databázi* – Toto nastane, když během přihlašování zadáme název databáze, s níž chceme pracovat. Chybová hláška může vypadat nějak takto: Access denied for user ……. To database ……. Nejspíše uživatel nemá dostatečná přístupová oprávnění pro práci s požadovanou databází. Toto se dá napravit přidáním potřebných oprávnění, například pomocí příkazu GRANT. Nebo pokud server MySQL běží například na počítači našeho poskytovatele internetového připojení. Tato možnost už byla rozebrána zezačátku této kapitoly. (Kofler, 2007,s. 332)

- *Není možné se přihlásit lokálně přes TCP/IP* – Tento problém je většinou na operačním systému Unix/Linux. Lokálně se dá připojit pokud je v parametru –h nastavena IP adresa nebo název počítače. Nejčastější příčinou této chyby jsou problémy při rozpoznávání hostitelského jména. Pokud se do user.Host vloží jméno hostitele s doménovým jménem, mělo by to už fungovat. Další možnou příčinou je špatné nastavení sítě. Tato možnost už byla také probrána na začátku této kapitoly. (Kofler, 2007,s. 332)

„Další seznam možných příčin a tipů pro jejich řešení můžeme najít v dokumentaci MySQL – http://dev.mysql.com/doc/mysql/en/access-denied.html .“ (Kofler, 2007,s. 332)

5 Výsledky a diskuze

Přístupová oprávnění pro přístup k databázím MySQL jsou propracovaná na nejmenší prvek databáze, což je sloupec. To umožňuje velkou variabilitu při přidělování těchto oprávnění. Pokud je ale potřeba pro několik uživatelů nastavit stejný rozsah pravomocí, určitě by bylo dobré mít možnost vytvořit skupinu uživatelů. Propracované nastavení třeba jen nad určité sloupce různých tabulek, nebo třeba jen nad některé tabulky v rozsáhlé databázi je dosti pracné a úmorné, pokud to ještě musíme opakovat pro více uživatelů. Tato možnost však v MySQL chybí. A to je určitě škoda, mnoho správců by to jistě uvítalo. Někdo může namítnout, že nastavování oprávnění nad jednotlivými sloupci tabulek se dá obejít pomocí pohledů. A poté nastavit oprávnění pouze na tento pohled. To samozřejmě jde. Ale tato možnost vede vždy k většímu zatížení serveru MySQL, a to samozřejmě vede ke snížení výkonnosti serveru.

Na server MySQL se dá přistupovat pomocí WWW sítě prakticky odkudkoliv. Pro jasné definování, odkud se jaký uživatel může připojit, se používá definice hostitelských počítačů v tabulce mysql.host. Pokud se však potřebuje uživatel připojovat z několika počítačů, vede to k nárůstu záznamů a pracnosti správy uživatelských účtů. Nemluvě o tom, když se potřebuje uživatel připojovat odkudkoliv, třeba když hodně cestuje a má pokaždé k dispozici jiný počítač. Toto se dá v MySQL vyřešit, pokud se do sloupce host v tabulce mysql.user zapíše uživatel@%, poté se může připojit odkudkoliv, ale neřeší jiné aspekty bezpečného připojení. Na úrovni systému počítače, na kterém je server MySQL umístěn, by se dala tato problematika vyřešit mnohem lépe. Například prostřednictvím certifikátu, který by prověřil bezpečnost připojení vzdáleného počítače a až po úspěšném ověření by se přistoupilo k samotnému přístupu do databáze MySQL.

6 Závěr

Literární rešerší z odborné literatury a důvěryhodných internetových zdrojů byli popsány principy a možnosti správy a zabezpečení přístupů MySQL. Ze získaných poznatků lze vyvodit, že systém přístupových oprávnění je dobře propracován a umožňuje přesnou specifikaci přístupových oprávnění, jak nad celými databázemi, tak nad jejich částmi (tabulky, sloupce tabulek). V možnostech těchto oprávnění však chybí možnost vytváření skupin uživatelů. To značně přidělává pracnost při správě uživatelů serveru MySQL.

Specifikace hostitelských počítačů a uživatelů, které se mohou připojovat k serveru MySQL, je dle doporučení jednoznačné. Tyto údaje jsou zapisovány v tabulce mysql.host a upřednostňuje se přesná specifikace buď pomocí názvu nebo IP adresy hostitelského počítače. Pro přenositelnost oprávnění uživatele je lepší pro ověření připojovaného počítače, používat certifikáty pro ověření důvěryhodnosti.

7 Seznam použitých zdrojů

GILMORE, Jason W., Velká kniha PHP a MySQL 5: kompendium znalostí pro začátečníky i profesionály. 1. vydání Brno: Toner press, 2007. 864 s. ISBN 978-80-86815-53-4.

SCNEIDER, Robert D., MySQL: oficiální průvodce tvorbou, správou a laděním databází. 1. vydání Praha: Grada, 2006. 372 s. ISBN 80-247-1516-3.

KOFLER, Michael, Mistrovství v MySQL 5. 1. vydání Brno: Computer press, 2007. 805 s. ISBN 978-80-251-1502-2.

GROFF, James R.,WEINBERG, Paul N., SQL Kompletní průvodce. 1. vydání Brno: CP Books, 2005. 936 s. ISBN *80-251-0369-2*

MASLAKOWSKI, Mark, Naučte se MySQL za 21 dní. 1. vydání Praha: Computer Press, 2001. 478 s. ISBN 807-22-644-86

http://aleph.techlib.cz/F

http://interval.cz/clanky/uzivatele-a-pristupova-prava-v-mysql/