dr inż. Marian Wrzesień
inż. Piotr Ryszawa
Przemysłowy Instytut Automatyki i Pomiarów

MULTI-INSTANCYJNY, WIELOWĄTKOWY SYSTEM BAZODANOWY MYSQL, W ŚRODOWISKU OS FEDORA KONTROLOWANYM PRZEZ SELINUX

Zaprezentowano wieloinstancyjną implementację bazodanowego systemu MySQL w serwerze pracującym pod systemem operacyjnym Linux – Fedora Core 11, kontrolowanym przez SELinux (Secure-Enhanced Linux). Istotą i celem prezentowanego rozwiązania jest zapewnienia podwyższonego bezpieczeństwa zasobów informatycznych poprzez udostępnienie zestawu subserwerów MySQL wraz z narzędziami do ich obsługi, wykorzystanie firewalla oraz wykorzystanie nadzoru SELinux – zaawansowanego systemu bezpieczeństwa wewnętrznego serwera. Wskazane powyżej podejście jest niezależne od ustanowionych dedykowanych praw dostępu do odpowiednich zasobów serwera zdefiniowanym użytkownikom systemu. Wprowadzanie szczególnych środków bezpieczeństwa jest podyktowane coraz większą otwartością systemów, zwłaszcza w środowisku z dostępem wykorzystującym narzędzia internetowe i zdalny dostęp on-line do zasobów informatycznych. Zaprezentowano tok postępowania oraz konfigurację serwera, zapewniające współbieżny, niezależny dostęp on-line do bazy danych MySQL.

THE MULTIINSTANCE, MULTI THREAD MySQL DATABASE SYSTEM, IN THE OS FEDORA ENVIRONMENT CONTROLLED BY SELINUX

The multi-instance MySQL database implementation in server running on the OS Fedora Core 11, which is controlled by SELinux (Secure-Enhanced Linux), is presented. The essence and the purpose of the presented solution is to provide increased security of the IT resources by sharing the set of sub- MySQL servers along with tools for theirs maintaining, use of the firewall as well as the SELinux use of surveillance – enhanced internal security system of the server. The above mentioned approach is independent of the established rights of access to a dedicated server IT resources defined for the respective system users. Placing special security measures is dictated by an increasingly open systems, especially in an environment with access using Internet tools and remote access on-line resources. The course of action, and server configuration, providing concurrent, independent on-line access to MySQL database is presented.

1. WSTĘP

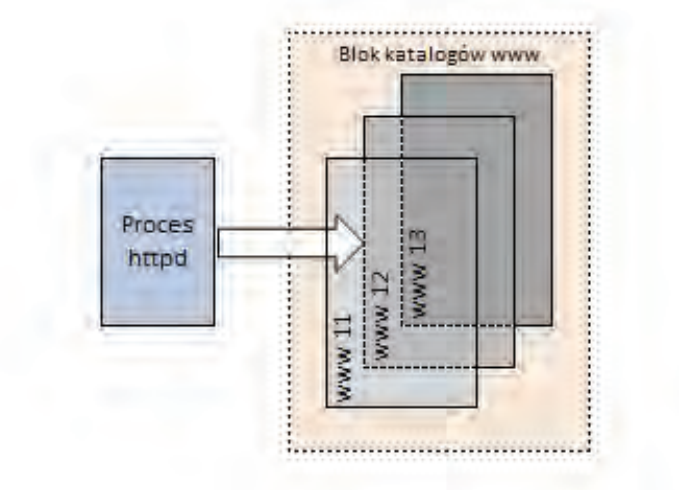
Celem opracowania wieloinstancyjnego bazodanowego system u MySQL w serwerze pracującym pod system em ope racyjnym Linux by ło zapewnienia podw yższonego bezpieczeństwa systemu obejmującego zasoby informatyczne implementowanych w systemie witryn internetowych. Przyj ęcie szczególnych środków bezpiecze ństwa jest podyktowane coraz większą otwartością systemów, zwłaszcza w środowisku z dost ępem wykorzystującym narzędzia internetowe i zdalny dostęp on-line do zasobów. Dla osiągnięcia postawionego celu zaimplementowano zestaw subserwerów MySQL wraz z narz ędziami do ic h obs ługi, wykorzystano firewall o raz zastoso wano nadzór zaawanso wanego system u bezpiecze ństwa wewnętrznego SELinux serwera. Utworzenie wielu instancji polega na zwielokrotnieniu i uniezależnieniu od siebie procesó w inicjowanych przez niezale żne dem ony MySQL, we współpracy z obsługującym je demonem httpd.

2. ŚRODOWISKO SELINUX

Implementacja rozwi ązania m ulti-instancyjnego MySQL zosta ła zrealizowana w serwerze pracującym pod system em operacyjnym Linux, kontrolowanym przez SELinu x [1, 2] (*Secure-Enhanced Linux*). SELinux to technologia, która pozwala na podniesienie poziom u bezpieczeństwa OS Linux, jako ca łości. Bazuje ona na koncepcji w ymuszonej kontroli dostępu (MAC – *Mandatory Access Control*). Ten m echanizm zabezpiecze ń nak łada restrykcję poziom u kontroli nad obiektam i, sprawowanym i zwykle przez u żytkowników będących w łaścicielami kreowanych przez nich obiektów. W przeciwie ństwie do typowej kontroli dost ępu DAC (*Discretionary Access Control*), okre ślanej jedynie przez praw a dostępu do plików, czy wspó łdzielenia zasobów (autoryzacja), MAC dodaje dodatkowe atrybuty, do wszystkich obiektów w system ie plików. SELinux definiuje tym samym tzw. kontekst bezpieczeństwa, składający si ę z trzech atrybutów bezpiecze ństwa: to żsamości, roli i typu. Przy podejm owaniu decyzji co do dost ępu, u żywany jest ka żdy z ty ch trzech składników kontekstu bezpieczeństwa. Zanim użytkownicy lub procesy b ędą mogli wchodzić w interakcje z odpowiednim i obiektami okre ślonymi kontekstowo, m uszą mi eć ustanowiony właściwy kontekst odpowiadający obiektowi, do którego się odnoszą.

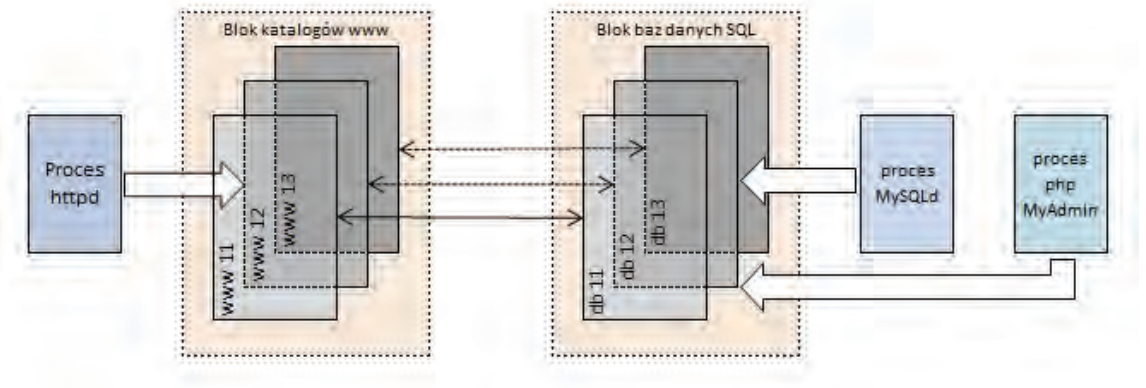
3. USYTUOWANIE PROCESU MYSQL W ARCHITEKTURZE WITRYNY WWW

Zwykle, podstawowym zasobem witryny internetowej jest jej katalog dom owy o strukturze umożliwiającej internautom nawi gowanie po udost ępnionych zasobach informacyjnych witryny (rys. 1).



Rys. 1. Podstawowa konfiguracja struktury witryny internetowej

W wielu wypadkach rozwi ązanie to jest wystarczaj ące do realizacji p rzekazu ok reślonych informacji. Jednak że udost ępnianie wi ększej ilo ści informacji i w łaściwe ni ą zarz ądzanie wymaga zastosowania baz danych. W tym celu, w system ie OS Linux m ożna zastosowa ć MySQL. Rys. 2 przedstawia strukturę takiego rozwiązania.



Rys. 2. Konfiguracja struktury witryny internetowej MySQL

Ogólnie, każdemu katalogowi dom owemu www jest wtedy przypisan a baza danych MySQL. Jej uruchom ienie wym aga funkcjonowania procesu *mysqld*. Poniżej om ówiono istot ę funkcjonowania takiego rozwi ązania w aspekcie przygotowa ń do tworzenia kolejnych instancji MySQL.

3.1. Pliki konfiguracyjne i nadzorujące proces MySQL

Proces MySQL [3, 4] wym aga nadzoru i kontroli, które s ą okre ślone przez port oraz parametry zawarte w poni ższych p likach (warto ści dom yślne), przy czym wprowadzono oznaczenia uwzględniające planowane do implementacji cztery instancje.

- port domyślny *3306*
- plik konfiguracyjny */etc/mysql[1-4]/my.cnf*
- katalog z danymi */var/lib/mysql[1-4]*
- gniazdo (socket) */var/lib/mysql[1-4]/mysql.sock*
- katalog z logiem */var/log/mysql[1-4]*
- pid (process id) */var/run/mysqld/mysqld[1-4].pid*

Każda z kolejno definiowanych instancji procesu MySQL powinna mieć powyżej określone cechy.

3.2. Atrybuty SELinux portów i plików obsługujących MySQL

Jak wy żej wspom niano, SELinux definiuje tzw. kontekst bezpiecze ństwa, który ustanawia wzajemne relacje pom iędzy procesam i i zasobam i systemu. W łaściwe dzia łanie kolejnych instancji M ySQL wymaga ć będzie okre ślenia atrybu tów bezpiecze ństwa odpowiednio dla zasobów i procesów przypisanych tym instancj om. Zgodnie z wzorcam i predefiniowanym i w systemie SELinux, atrybuty te powinny być przypisywane jak następuje:

port 33307, 33308,33309,33310	*typ SELinux: http_port_t, protokół: tcp*
/etc/mysql[1-4]/my.cnf	*unconfined_u:object_r:etc_t:s0*
/var/lib/mysql[1-4]/	*system_u:object_r:mysqld_db_t:s0*
/var/lib/mysql[1-4]/mysql.sock	*system_u:object_r:mysqld_var_run_t:s0*
/var/log/mysqld[1-4]/	*system_u:object_r:mysqld_db_t:s0*
/var/run/mysqld/mysqld[1-4].pid	*system_u:object_r:mysqld_var_run_t:s0*

Każda z kolejno definiowanych instancji procesu MySQL powinna mieć – przypisane portom i plikom – atrybuty rozszerzone zgodnie z powyższym zapisem.

3.3. Porty MySQL w firewall

Domyślnym portem dla procesu MySQL jest port 3306. Dla zapewnienia właściwego działania procesu port ten nie musi być udostępniany w firewall. Jedynie w przypadkach bezpośredniego połączenia z lokalizacji zewnętrznej z MySQL, powinien on być w firewall udostępniany. Podobnie, powinno to być realizowane dla portów zdefiniowanych dla każdej z kolejnych instancji.

3.4. Pliki konfiguracyjne i nadzorujące proces phpMyAdmin

phpMyAdmin [5, 6] to narzędzie służące do łatwego, zdalnego zarządzania bazą danych MySQL, napisane w języku PHP. Oprogramowanie umożliwia między innymi tworzenie i/lub usuwanie baz danych, dodawanie i/lub kasowanie relacji, zarządzanie użytkownikami, oraz edycję ich struktury i zawartości. Wszystkie operacje mogą być wykonywane z poziomu przeglądarki internetowej, w graficznym środowisku, bez konieczności pracy z domyślnym interfejsem tekstowym.

Proces phpMyAdmin wymaga nadzoru i kontroli, które są określone przez port oraz parametry zawarte w poniższych plikach (wartości portów przyjęte w rozwiązaniu):

- port dostępowy do MySQL *33307, 33308,33309,33310*
- plik konfiguracyjny */etc/phpMyAdmin[1-4]/config.inc.php*
- katalog domowy */var/www/html/phpMyAdmin[1-4]*

Każda z kolejno definiowanych instancji procesu MySQL powinna mieć zaimplementowane narzędzie phpMyAdmin, zgodnie z powyższym zapisem. O właściwym działaniu phpMyAdmin decyduje konfiguracja zawarta w plikach config.inc.php, zawierająca informacje m.in. o porcie dostępowym, hoście (w przedstawionym rozwiązaniu 127.0.0.1), oraz o gnieździe (socket) odpowiedniego procesu MySQL administrowanego za pomocą phpMyAdmin.

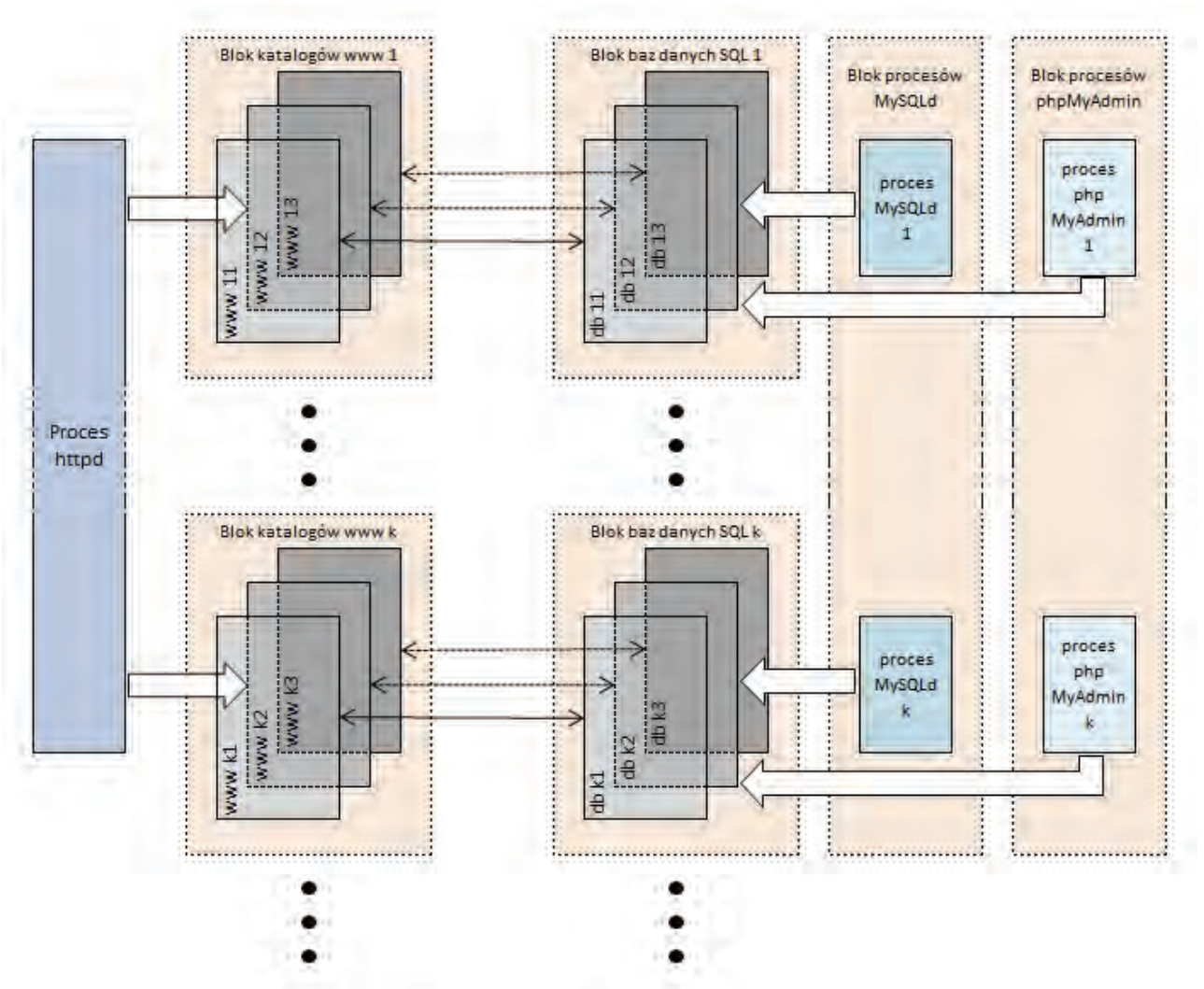
Dostęp do poszczególnych instancji MySQL poprzez phpMyAdmin jest zdefiniowany w pliku konfiguracyjnym serwera www Apache */etc/httpd/conf/httpd.conf*. W pliku tym, dla każdej instancji został zdefiniowany wirtualny host (VirtualHost), wskazujący właściwe zasoby przy wybranym, szyfrowanym (https) wywołaniu procesu phpMyAdmin. Poniżej przedstawiono konfigurację dla przykładowej instancji (nr 3):

```
<VirtualHost *:80>
   DocumentRoot /var/www/html/phpMyAdmin3
   ServerName mysql3.piap.pl
   Redirect /https://mysql3.piap.pl:4433
</VirtualHost>
<VirtualHost *:4433>
   DocumentRoot /var/www/html/phpMyAdmin3
   ServerName mysql3.piap.pl
   SSLEngine on
   SSLCertificateFile /etc/pki/https/server.pem
   SSLCertificateKeyFile /etc/pki/https/server.pem
</VirtualHost>
```

Z powyższego wynika, że połączenie z serwerem www po porcie h ttp: 80 jest autom atycznie przekierowywane na bezpieczn e po łączenie https: 4433, które jest szyfrowane z wykorzystaniem protokołu SSL (*Secure Socket Layer*). Port, na który nast ępuje przekierowanie – w tym przypadku 4433 – musi by ć udost ępniony w system ie ochrony firewall.

4. REALIZACJA IMPLEMENTACJI MULTI-INSTANCYJNEJ MYSQL

Omawiane w artykule zwielokrotnienie zosta ło przepro wadzone dla czterech instan cji. Oznacza to, że przep rowadzono multip likację procesów MySQL i narz ędzi do zarz ądzania bazami danych phpMyAdm in oraz pogrupowano bazy danych oraz katalogi www przypisane odpowiednim instancjo m. Nie ma konieczno ści powielania procesu httpd odpowiadaj ącego za funkcjonowanie serwera www. Konfig uracja tak przebudowanego system u jest przedstawiona na rys. 3



Rys. 3. Konfiguracja multi-instancyjnej struktury witryny internetowej MySQL

4.1. Multiplikacja procesu MySQL

Podczas multiplikacji procesu MySQL, dla poszczególnych instancji:

- zdefiniowano porty *33307,33308, 33309,33310* i przypisano im atrybuty rozszerzone: *semanage port –a –t http_port_t –p tcp 333[07-10]*
- założono katalogi konfiguracyjne, okre ślono własność, skopiowano do nich odpowiednio zmodyfikowane pliki i przypisano im atrybuty rozszerzone:

```
mkdir   /etc/mysql[1-4]
chown –R root:root /etc/mysql[1-4]
cp –a /etc/my.cnf /etc/mysql[1-4]/my.cnf
semanage fcontext –a –t etc_t  " /etc/mysql[1-4](/.*)?"
restorecon  –R –v  /etc/mysql[1-4]/
```

- przeprowadzono modyfikacje plików my.cnf:

```
[mysqld]
port=333[07-10]
datadir=/var/lib/mysql[1-4]
socket=/var/lib/mysql[1-4]/mysql.sock
user=mysql
old_passwords=1
[mysqld_safe]
log-error=/var/log/mysql1/mysqld.log
pid-file=/var/run/mysqld/mysqld1.pid
[ndbd]
connect-string="nodeid=2;host=localhost:1186"
[ndb_mgm]
connect-string="host=localhost:1186"
```

- założono katalogi danych, okre ślono w łasność, skopiowano odpowiednie pliki i przypisano im atrybuty rozszerzone:

```
mkdir /var/lib/mysql[1-4]
chown -R mysql.mysql /var/lib/mysql[1-4]/
semanage fcontext –a –t mysql_db_t  " /var/lib/mysql[1-4](/.*)?"
restorecon  –R –v  /var/lib/mysql[1-4]/
```

- założono katalogi logów, nadano własność i przypisano im atrybuty rozszerzone:

```
mkdir /var/log/mysql[1-4]
chown -R mysql.mysql /var/log/mysql[1-4]
semanage fcontext –a –t mysqld_log_t  " /var/log/mysqld[1-4].log
restorecon –R  –v  /var/log/mysql[1-4].log
```

Do zainicjowania i uruchomienia instancji MySQL stosuje się poniższe polecenia:

```
mysql_install_db --user=mysql --datadir=/var/lib/mysql[1-4]/
mysqld_safe --defaults-file=/etc/mysql[1-4]/my.cnf &
```

W celu auto matycznego urucham iania zd efiniowanych instancji, po uruchom ieniu system u OS Linux, do pliku */etc/rc.local* wpisano poniższe polecenia:

```
mysqld_safe --defaults-file=/etc/mysql1/my.cnf &
mysqld_safe --defaults-file=/etc/mysql2/my.cnf &
mysqld_safe --defaults-file=/etc/mysql3/my.cnf &
mysqld_safe --defaults-file=/etc/mysql4/my.cnf &
```

4.2. Multiplikacja narzędzi phpMyAdmin

Podczas multiplikacji narzędzi phpMyAdmin, dla poszczególnych instancji:

- założono katalogi konfiguracyjne, okre ślono własność, skopiowano do nich odpowiednio zmodyfikowane pliki */etc/phpMyAdmin* i przypisano im atrybuty rozszerzone jak następuje:

```
cp –a /etc/phpMyAdmin /etc/phpMyAdmin[1-4]
chown –R root:root /etc/phpMyAdmin[1-4]
semanage fcontext –a –t etc_t  " /etc/phpMyAdmin[1-4](/.*)?"
restorecon  –R –v  /etc/mysql[1-4]/
```

- przeprowadzono następujące modyfikacje plików */etc/phpMyAdmin[1-4]/config.inc.php*:

```
$cfg['Servers'][$i]['host']          = '127.0.0.1';
$cfg['Servers'][$i]['port']          = '333[07-10]';
$cfg['Servers'][$i]['socket']        = '/var/lib/mysql[1-4]/mysql.sock';
[mysqld]
port=333[07-10]
```

- założono katalogi danych, określono własność, skopiowano odpowiednie pliki i przypisano im atrybuty rozszerzone:

```
cp –a usr/share/phpMyAdmin /var/www/html/phpMyAdmin[1-4]
chown -R mysql.mysql /var/lib/mysql[1-4]/
semanage fcontext –a –t httpd_sys_content_t "/var/www/html/phpMyAdmin[1-4](/.*)?"
restorecon –R –v /var/www/html/phpMyAdmin[1-4]
```

5. TESTOWANIE IMPLEMENTACJI MULTI-INSTANCYJNEJ MYSQL

Testowanie przeprowadzono dla każdej z zaimplementowanej instancji. Poniżej opisano czynności przeprowadzane dla – przykładowej – instancji Nr 3. Omówione czynności zostały przeprowadzone dla każdej z czterech instancji MySQL wprowadzonych do systemu.

5.1. Uruchomienie instancji

Testowanie rozpoczyna się sprawdzeniem poprawności uruchomiania instancji. Przeprowadza się to z wykorzystaniem poniższych poleceń:

```
mysql_install_db --user=mysql --datadir=/var/lib/mysql3/
mysqld_safe --defaults-file=/etc/mysql3/my.cnf &
```

Weryfikację poprawności tego testu sprawdza się poleceniem:

```
ps -axf | grep mysql3,
```

w którego wyniku uzyskuje się informacje o zachowaniu procesu (numery procesów losowe)

```
2290 ?   S   0:00 /bin/sh /usr/bin/mysqld_safe --defaults-file=/etc/mysql3/my.cnf
2611 ?   Sl  1:36  \_ /usr/libexec/mysqld --defaults-file=/etc/mysql3/my.cnf --basedir=/usr --
                   datadir=/var/lib/mysql3 --user=mysql --log-error=/var/log/mysql3/mysqld.log --
                   pid-file=/var/run/mysqld/mysqld3.pid --socket=/var/lib/mysql3/mysql.sock --
                   port=33309
```

Z powyższego wynika, że instancja nr 3 została uruchomiona zgodnie z wcześniej zdefiniowanymi parametrami.

5.2. Logowanie się do bazy danych instancji MySQL

Po uruchomieniu instancji sprawdza się możliwość zalogowania się do bazy danych MySQL uruchomionej instancji. Można to przeprowadzić na dwa sposoby:

- Uruchamiając program ze wskazaniem odpowiedniego portu.
  mysql -h 127.0.0.1 -P 33309, lub
- Uruchamiając program ze wskazaniem odpowiedniego gniazda.
  mysql -S var/lib/mysql3/mysql.sock

O poprawności powyższego testu świadczy zalogowanie się do serwera MySQL, na wiązanie z nim komunikacji i przeprowadzanie operacji na bazie danych dostępnych w systemie.

5.3. Logowanie się do bazy danych z lokalizacji oddalonej

Następnym krokiem jest przetestowanie możliwości zalogowania się do bazy danych MySQL określonej instancji - przez przeglądarkę internetową - za pośrednictwem phpMyAdmin. W tym celu, w przeglądarce internetowej podaje się adres:

http://mysql2.piap.pl

Po wpisaniu adresu połączenie zostaje przekierowane – zgodnie z zapisem w pliku konfiguracyjnym */etc/httpd/conf/httpd.conf* – na adres:

https://mysql3.piap.pl:4433

Po akceptacji certyfikatu podaje się login i hasło. Zalogowanie się do odpowiedniej instancji świadczy o poprawnej konfiguracji phpMyAdmin.

6. WNIOSKI

1. Zaimplementowanie kilku instancji MySQL pozwala na udostępnienie praw administracyjnych niezależnym od siebie projektantom i administratorom witryn internetowych w tym samym serwerze. Liczba możliwych do zaimplementowania instancji nie jest ograniczona.
2. Każdemu z administratorów można przypisać prawa dostępu do zestawu zarządzanych przez niego katalogów domowych witryn internetowych oraz do zestawu baz danych MySQL związanych z tymi katalogami.
3. Bezpieczne, szyfrowane, zdalne zarządzanie odpowiednim i zestawami witryn internetowych, podlegającymi w łaściwym administratorom tych witryn uzyskuje się, wykorzystując niezależne – oddzielne dla każdego z zestawów witryn – procesy phpMyAdmin.

Kontekstowe określenie obiektów składających się na zestawy katalogów i baz danych witryn, zwiększa bezpieczeństwo poprzez zdefiniowanie wzajemnych relacji pomiędzy procesami, a zasobami, na które procesy te mogą oddziaływać.

BIBLIOGRAFIA

1. Bill Mc Carthy, SELinux, O'Reilly, 2004.
2. Red Hat Enterprise Linux Deployment Guide, Red Hat Enterprise Linux 5, Red Hat, Inc. , 2006.
3. Paul DuBois , MySQL Wydanie II, MIKOM PWN, 2004
4. MySQL 5.5 Reference Manual, Oracle Corporation, 2010
5. Mastering phpMyAdmin for Effective MySQL Management, Packt Publishing, 2004
6. Łukasz Sosna, phpMyAdmin – proste zarządzanie bazą MySQL, NAKOM, 2006.