Ochrona bazy danych

Jeżeli udostępniamy bazę danych poprzez stronę WWW, to musimy zabezpieczyć serwer baz danych. Od razu należy powiedzieć, że administratorzy blokują możliwości działań ryzykownych, niemniej musimy znać zagrożenia.

MySQL w aspekcie serwera Apache

Należy unikać uruchamiania serwera MySQL z poziomu użytkownika *root*. Dałoby to każdemu użytkownikowi logującemu się do serwera baz danych, prawo do odczytywania i zapisywania wszystkich plików. Włamanie do witryny nie byłoby w takiej sytuacji dużym problemem. Powinno się zarejestrować nowego użytkownika, przeznaczonego do uruchamiania serwera MySQL.

Stosowanie haseł

Każdemu użytkownikowi powinniśmy nadać hasło dostępu, które powinno być często zmieniane i trudne do odgadnięcia. Najlepszym rozwiązaniem jest losowa kombinacja liter i cyfr. Jeżeli hasła są przechowywane w plikach, to pliki te powinny być dostępne tylko dla właścicieli tych haseł.

Skrypty PHP operujących na bazie danych, muszą najpierw połączyć się z danym serwerem, co wymaga podania hasła dostępu odpowiedniego użytkownika. Hasło oraz identyfikator tego użytkownika zapisuje się w pliku dołączanym do skryptu za pomocą funkcji `include()`. Plik ten powinien być zapisany w innej lokalizacji jak dokumenty WWW i dostępny, tylko dla zalogowanego użytkownikowi.

Jeżeli plik z hasłem jest przechowywany w tej samej lokalizacji co dokumenty WWW, to należy się upewnić że nie będą one analizowane przez interpreter PHP ani też wyświetlone przez przeglądarkę.

Formularz logowania:

```
<body>
  <form action="wynik.php" method="post">
    Login<input type="text" name="login" />
    Hasło<input type="password" name="haslo" />
    <input type="submit" name="loguj"  value="Zapisz w pliku" />
  </form>
</body>
```

Plik ze skryptem PHP:

```
  <?php
    $login=$_POST['login'];
    $haslo=$_POST['haslo'];
    $do_zapisania=md5($login).' '.md5($haslo);

    @$plik=fopen('haslo.txt','w');
    if (!$plik)
    {
      echo 'Wystąpił błąd podczas otwierania pliku!';
      exit;
    }

    if (!flock($plik, LOCK_EX))
    {
      echo 'Wystąpił błąd podczas zakładania blokady pliku!';
      fclose($plik);
      exit;
    }
```

```
    fwrite($plik,$do_zapisania);
    flock($plik, LOCK_UN);
    fclose($plik);
    echo 'Operacja zapisywania danych zakończona sukcesem!';
  ?>
```

Hasła przechowywane w bazie danych powinny być szyfrowane jednostronnie (bez możliwości odszyfrowania) za pomocą funkcji `md5()` lub `sha1()`. Należy wtedy pamiętać, że po zapisaniu zaszyfrowanego hasła za pomocą `INSERT`, w treści polecenia `SELECT` należy zastosować tę samą funkcję szyfrującą aby porównać hasła zaszyfrowane.

Przywileje użytkowników

Najważniejszą zasadą jest przyznawanie minimalnych, tylko niezbędnych uprawnień. Bardzo ryzykowne jest nadanie przywilejów `PROCESS`, `FILE`, `SHUTDOWN` i `RELOAD` użytkownikom, którzy nie są administratorami. Użytkownik z uprawnieniem `PROCESS`, ma możliwość przeglądania czynności i danych innych użytkowników, w tym również haseł dostępu, natomiast `FILE` uprawnia do edycji plików systemowych serwera. Ostrożność jest wskazania przy nadawaniu uprawnienia `GRANT`, ponieważ taki użytkownik może nadawać innym użytkownikom swoje uprawnienia.

Dane wysyłane za pomocą formularzy

Powinny być dokładnie weryfikowane co do treści jak również ich rozmiaru.

Dane poufne

Jeżeli użytkownicy mają wysyłać do bazy dane poufne lub podawać hasła dostępu, to powinno się zastosować protokół SSL (Secure Socets Layer), są one wtedy przekazywane z przeglądarki do serwera w postaci zaszyfrowanej, a nie jako zwykły tekst.

Tworzenie kopii zapasowej bazy danych MySQL

Tworzenie kopii zapasowych baz danych (im częstsze tym lesze; w pewnych granicach rozsądku) jest koniecznością.

Ćwiczenie.1. Tworzenie kopii bazy za pomocą polecenia *mysqldump*

Przy pomocy phpMyAdmina - zakładka **Użytkownicy/Dodaj użytkownika**, dodamy użytkownika o nazwie **admin**, haśle **admin123** ze wszystkimi uprawnieniami.

Następnie otwieramy panel kontrolny XAMPP-a i przyciskiem *Shell* otwieramy okienko wiersza poleceń. Chcemy utworzyć kopię bazy `samochody` i kopię tę zapisać w pliku `kopia_samochody.sql`. Musimy napisać polecenie:

```
mysqldump samochody -u admin -p > kopia_samochody.sql
```

i wcisnąć *Enter*. Dalej musimy podać hasło, ponownie wcisnąć *Enter* i to wszystko.

Znajdź w katalogu głównym XAMPP-a, plik `kopia_samochody.sql`.

Przywracanie bazy danych MySQL

Przywrócenie kopii zapasowej bazy danych jest równie proste, jak jej wykonanie. Wystarczy zalogować się do systemu bazy danych i wskazać plik z przechowywaną kopią. Ten plik wygenerowano podczas Ćwiczenia.1 (`kopia_samochody.sql`) i powinien się znajdować w katalogu głównym XAMPP-a. Zalogujemy się jako `admin` z hasłem `admin123`.

Ćwiczenie.2. Przywracanie bazy danych

Najpierw usuń za pomocą phpMyAdmina naszą bazę `samochody`, a następnie przywróć ją.

Następnie ponownie utworzymy bazę `samochody`, za pomocą polecenia SQL:

```
CREATE DATABASE samochody;
```

Powinniśmy mieć ponownie naszą bazę - pustą. Otwieramy panel kontrolny XAMPP-a i przyciskiem *Shell* otwieramy okienko wiersza poleceń. Chcemy przywrócić bazę `samochody` zapisaną w pliku `kopia_samochody.sql`. Wpisujemy polecenie:

```
mysql samochody -u admin -p < kopia_samochody.sql;
```

wciskamy *Enter*, wpisujemy hasło, ponownie wciskamy *Enter* i chwilę czekamy aż zostaną wykonane wszystkie zapytania zapisane w pliku zrzutu.