

Logowanie w PHP

W tym tutorialu postaram się opisać, jak w prosty sposób stworzyć logowanie na stronie. Jest ono przydatne, np. gdy chcemy udostępnić jakieś informacje

lub funkcjonalność tylko zarejestrowanym użytkownikom – np. dać możliwość dodawania nowości, zdjęć etc. Potrzebne będą:

- Serwer z obsługą PHP, np. WAMP lub XAMPP lub konto hostingowe
- Podstawowe wiadomości dotyczące PHP i HTML
- W dalszej części dodatkowo serwer MySQL
- Zapał do pracy :)

Tekst został podzielony na dwie części: logowanie z użyciem plików tekstowych, oraz z użyciem bazy danych MySQL, gdzie także zostanie umieszczone porównanie obu sposobów. Pomijam tu element konfiguracji serwera jak i bazy danych, jako iż było to opisywane w sieci wiele razy i nie dotyczy tego tekstu.

Część pierwsza – logowanie z użyciem plików tekstowych

Na początku będziemy potrzebowali trzech plików: *login.php* – zawierającego formularz logowania, *db.php* – zawierającego informacje o użytkownikach, oraz *secret.php* – czyli pliku, który chcemy zabezpieczyć przed wścibskimi użytkownikami.

Zaczynamy – tworzymy plik *login.php* i umieszczamy w nim nasz formularz z danymi, które musi wpisać użytkownik, aby uzyskać dostęp do tajnego pliku:

<http://ideone.com/4Wmqm>

Jak widać powyżej, w formularzu zostały umieszczone 2 pola: *user*, czyli nazwa użytkownika oraz *passwd*, czyli hasło tegoż użytkownika. Następnie musimy stworzyć bazę danych zawierającą dane użytkowników (plik *db.php*):

<http://ideone.com/vldGC>

Kilka słów wyjaśnień: w pierwszej linii otwieramy kod PHP i sprawdzamy (funkcja *defined()*), czy stała

SECRET_CONST

jest zadeklarowana. Jeśli nie – zostanie wywołana funkcja *die()*

, która kończy wykonanie skryptu. Jest to zabezpieczenie przed odczytaniem przez niepowołanych użytkowników zawartości tego pliku oraz przed próbą dołączenia go do niepowołanego skryptu. Jest to skrócona forma, wykorzystująca pewne cechy języka PHP, równie dobrze można by tę linię zapisać w taki sposób:

<http://ideone.com/qOxnM>

Lecz, jak widać tamten kod jest znacznie krótszy. Dalej mamy definicję dwuwymiarowej tablicy – pierwsza kolumna oznacza ID użytkownika (w wypadku pozostawienia pustego nawiasu ID będzie ustalane automatycznie od 0). Druga kolumna oznacza odpowiednio nazwę użytkownika (pole *user*) oraz jego hasło (pole *passwd*), które jest zakodowane za pomocą wybranej funkcji haszującej (w tym wypadku SHA-256) – w tym wypadku hasz odpowiada hasłu „*adminpasswd*”. Jest to niezbędne, gdyż zabezpiecza nas w dużym stopniu przed uzyskaniem nieautoryzowanego dostępu nawet w wypadku wykradzenia pliku z informacjami o użytkownikach (patrz: mój artykuł o *bezpieczeństwie danych osobowych w serwisach internetowych*, a w szczególności sekcja dot. odzyskiwania haseł). Dodanie następnego użytkownika ogranicza się do skopiowania drugiej i trzeciej linii oraz zmianie nazwy użytkownika i hasła. Do wygenerowania haszu z hasła można skorzystać z dostępnych w Internecie narzędzi, np. <http://www.insidepro.com/ashes.php>

Pozostał jeszcze jeden plik – *secret.php*, do którego dostęp ma być ograniczony. Oto elementy, które musi zawierać:

- Definicja naszej stałej *SECRET_CONST*, dzięki której będziemy mogli skorzystać z naszej bazy danych
- Sprawdzenie, czy użytkownik istnieje w bazie danych

Logowanie w PHP - część pierwsza

Wpisany przez Tomasz Stasiak
piątek, 22 kwietnia 2011 22:02

- Zapisanie w sesji informacji o tym, że użytkownik ma prawo do dostępu (aby nie trzeba było się logować po każdym odświeżeniu strony)
- Wyświetlenie naszej tajnej treści

Zabieramy się do pracy – tworzymy plik *secret.php* o następującej treści:

<http://ideone.com/gkQuP>

Wyjaśnienie linijka po linijce: na początku otwieramy tag PHP aby móc wykonać nasz kod, w następnej linii rozpoczynamy sesję, aby można było z niej korzystać. Następnie definiujemy naszą stałą *SECRET_CONST*, która umożliwi nam korzystanie z bazy użytkowników; w czwartej i piątej linii upewniamy się, czy aby na pewno użytkownik wpisał dane do formularza i czy nie jest już przypadkiem zalogowany, jeśli nie – wyświetlamy odpowiedni komunikat oraz kończymy działanie skryptu. Jeśli już jesteśmy upewnieni, że użytkownik wpisał dane do formularza lub jest zalogowany możemy przystąpić do sprawdzania poprawności wpisanych przez niego danych. Dołączamy plik z informacjami o użytkownikach (dyrektywa *include()*

) oraz w pętli

for()

sprawdzamy, dane wszystkich użytkowników (do czasu napotkania prawidłowych danych). Jeśli użytkownik podał prawidłowe dane, ustawiamy wartość

\$_SESSION['logged_in']

na

true

, aby oznaczyć informację o tym, że użytkownik został zalogowany. W następnej sekcji kodu PHP (wewnątrz sekcji HTML BODY) sprawdzamy, czy użytkownik jest zalogowany; jeśli tak – wyświetlamy naszą tajną treść, jeśli nie – także odpowiedni komunikat.

To by było na tyle – w następnej części opiszę jak wykonać logowanie z wykorzystaniem bazy danych MySQL.

//edit: [paczka z kodem do pobrania](#)