aplikacja webowa sprzedaj.pl

Amadeusz Gunia, Marek Kwak

Plan demonstracji

1. Przygotowanie środowiska
2. Demonstracja aplikacji
3. Zabezpieczenia
4. Podatności
5. Zadania

Przygotowanie środowiska

1. Zainstaluj XAMPP
   - Apache
   - MySQL
2. Uruchom XAMPP i ww. usługi
3. Otwórz panel phpMyAdmin
4. Utwórz nową bazę danych o nazwie bawim
5. Zaimportuj zrzut bazy danych bawim.sql
6. Usuń domyślną zawartość folderu htdocs i przenieś do niego pliki naszej aplikacji
7. Otwórz przeglądarkę internetową (np. Chrome) i wpisz w niej adres ***localhost*** lub ***adres IP*** swojego komputera
8. Dodatkowo (jeśli korzystasz w Windowsa) przygotuj maszynę wirtualną z zainstalowanym systemem Linux

Demonstracja aplikacji
live demo

Zabezpieczenia

- walidacja danych wejściowych przy dodawaniu ogłoszenia

```
if (strlen($new_title) < 5) {
    $message = "Tytuł musi mieć co najmniej 5 znaków.";
} else {
    if (strlen($new_title) > 30) {
        $message = "Tytuł może mieć co najwyżej 30 znaków.";
    } else {
        if (empty($_FILES["picture"]["name"])) {
            $message = "Nie wybrano zdjęcia.";
        } else {
            if (!getimagesize($_FILES["picture"]["tmp_name"])) {
                $message = "Przesyłane zdjęcie jest nieprawidłowe.";
            } else {
                if ($_FILES["picture"]["size"] > 5000000) {
                    $message = "Przesyłane zdjęcie jest za duże.";
                } else {
                    if (strlen($new_description) < 20) {
                        $message = "Opis musi mieć co najmniej 20 znaków.";
                    } else {
                        if (strlen($new_description) > 9000) {
                            $message = "Opis może mieć co najwyżej 9000 znaków.";
                        } else {
                            if (!(preg_match('/^[0-9]{0,7}[,][0-9]{2}$/', $new_price) || preg_match('/^[0-9]{0,7}$/', $new_price))) {
                                $message = "Nieprawidłowa cena, format ceny: 1234567,50";
                            } else {
                                if (!preg_match('/^([0-9]{2}-[0-9]{3})$/D', $new_postcode)) {
                                    $message = "Podany kod pocztowy jest niepoprawny.";
                                } else {
                                    if (strlen($new_location) > 30) {
                                        $message = "Lokalizacja może mieć co najwyżej 30 znaków.";
                                    } else {
                                        if (!preg_match('/^[0-9]{9}$/', $new_phone_number)) {
                                            $message = "Podany numer telefonu jest niepoprawny.";
```

Zabezpieczenia

- nadawanie losowej nazwy przesyłanym plikom

```php
$new_path = "uploads/" . generateRandomString() . "." . pathinfo($_FILES["picture"]["name"])['extension'];

function generateRandomString($length = 20) {
    $characters = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';
    $charactersLength = strlen($characters);
    $randomString = '';
    for ($i = 0; $i < $length; $i++) {
        $randomString .= $characters[rand(0, $charactersLength - 1)];
    }
    return $randomString;
}
```

- automatyczne usuwanie ogłoszeń w przypadku usunięcia konta

```sql
ALTER TABLE `advertisements`
  ADD CONSTRAINT `uid` FOREIGN KEY (`uid`) REFERENCES `users` (`uid`) ON DELETE CASCADE ON UPDATE CASCADE;
COMMIT;
```

Podatności

- wadliwe logowanie podatne na SQL Injection

```php
if (isset($_POST['login']) && isset($_POST['password'])) {
    if ($_POST['login'] == '' || $_POST['password'] == '') {
        $message = "Pola nie mogą być puste.";
    } else {
        $login = $_POST['login'];
        $password = crypt($_POST['password'], '$1$salt$');
        $sql = $dbh->query("SELECT * FROM users WHERE email = '$login' AND password = '$password'");
        $user = $sql->fetch(PDO::FETCH_ASSOC);
        if ($user) {
            $_SESSION['uid'] = $user['uid'];
            $_SESSION['email'] = $user['email'];
            header('Location: /');
        } else $message = "Niepoprawne dane.";
    }
}
```

Podatności

- cross-site scripting (XSS)

```
$title = $row['title'];
$category = $row['category'];
$path = $row['path'];
$description = $row['description'];
$status = $row['status'];
$price = $row['price'];
$postcode = $row['postcode'];
$location = $row['location'];
$phone_number = $row['phone_number'];
$username = $row['name'];

<div class="col-md-4">
    <img class="big_img" src="<?php echo $path ?>" alt="">
</div>
<div class="col-md-4">
    <br>
    <p class="adv-data" style="font-size: 32px; font-weight: bold; font-family: Helvetica;"><?php echo $title ?></p>
    <p class="adv-data" style="font-size: 30px; font-weight: bold;"><?php echo $price ?> PLN</p>
    <p class="adv-data" style="font-size: 15px;">Kategoria: <?php echo $category ?></p>
    <p class="adv-data" style="font-size: 15px;">Stan: <?php echo $status ?></p>
</div>
<div class="col-md-4">
    <br> <br>
    <p class="adv-info"> Użytkownik: <b style="font-size: 22px;"><?php echo $username ?></b></p>
    <p class="adv-info"> Telefon: <b style="font-size: 22px;"><?php echo $phone_number ?></b></p>
    <p class="adv-info"> Lokalizacja: <b style="font-size: 22px;"><?php echo $postcode . ', ' . $location ?></b></p>
</div>
```

Zadania

tzw. część praktyczna

Zadanie 1 - SQLi

- Zaloguj się na konto *sprzedawca@agh.pl* nie znając hasła. Wykorzystaj do tego podatność SQL Injection.

- Przydatne informacje dot. MySQL:
  - 'tralalala'
  - -- komentarz
  - ; koniec zapytania

- Jako odpowiedź do tego zadania prześlij zrzut ekranu zawierający stronę logowania z wypełnionymi i widocznymi polami email i hasło.

Zadanie 2 - naprawa logowania

- Napraw logowanie wykorzystując tzw. spreparowane instrukcje.

- Konstrukcja *prepared statement*:

```
$stmt = $dbh->prepare("SELECT * FROM table_name WHERE col = :var");
$stmt->execute([':var' => $php_var]);
```

- Można także przenieść weryfikację hasła poza zapytanie SQL.

- Jako odpowiedź do tego zadania prześlij zrzut ekranu zawierający fragment poprawionego kodu – plik ***login.php***.

Zadanie 3 - łamanie haseł

- Załóżmy, że udało Ci się przechwycić hasła użytkowników. Spróbuj je złamać mając na uwadze, że serwis używa przestarzałego algorytmu do hashowania – MD5. Hasła znajdują się w pliku passwords.txt.

- Potrzebne narzędzia:
  - Virtual Box lub VMware player + Linux (dowolny)
  - narzędzie do łamania haseł - John the Ripper, Hashcat lub inne
  - instalacja: `sudo apt-get install john -y`

- Jako odpowiedź do tego zadania prześlij zrzut ekranu zawierający rozszyfrowane hasła.

- Przydatne strony:
  - czy hasło kiedyś wyciekło - https://haveibeenpwned.com/Passwords
  - czy hasło jest mocne - https://www.passwordmonster.com

Zadanie 4 - cross-site scripting (XSS)

- Prześlij do bazy danych złośliwy kod i użyj go w niecny sposób wykorzystując podatność cross-site scripting.
- Idealnym miejscem do tego ataku będzie strona dodawania ogłoszeń. Znajdź pole gdzie możesz wprowadzić dużo tekstu. Napisz skrypt JS, który np. pokoloruje element strony lub wyświetli przerażający alert.
- Ważne: kodem pocztowym w Twoim ogłoszeniu musi być 00-000
- Jako odpowiedź do tego zadania prześlij zrzut ekranu obrazujący wykonanie Twojego złośliwego kodu – ***localhost/xss***.
- Podpowiedź: Jeśli nie masz pomysłu – poszukaj na stack'u.

Zadanie 5 - zabezpieczenie przed XSS

- Twig zapewnia zabezpieczenie przed XSS, jednak dla potrzeb tego zadania strona ***/xss*** celowo generuje front za pomocą czystego php.

- Znajdź i dopisz w odpowiednim miejscu funkcję, która konwertuje wszystkie znaki specjalne na encje HTML, nieinterpretowane przez przeglądarkę.

- Jako odpowiedź do tego zadania prześlij zrzuty ekranu zawierające:
  - poprawnie wyświetlane ogłoszenie - ***localhost/xss***
  - fragment poprawionego kodu – plik ***xss.php***

Bibliografia

- https://www.olx.pl/
- https://zaworski.pl/
- https://www.php.net/
- https://stackoverflow.com/
- https://www.w3schools.com/
- https://miloserdov.org/?p=5477
- https://www.apachefriends.org/pl/download.html
- https://mansfeld.pl/bezpieczenstwo/sql-injection-zabezpieczenie-php-mysql/
- https://www.theguardian.com/technology/2016/dec/15/passwords-hacking-hashing-salting-sha-2