OWASP 1 - SQL Injection

Veilig programmeren

Veilig programmeren is een belangrijk onderdeel van Software Development, zie afbeelding:

momdropstable.png

SQL Injection

In deze les gaan we een login form maken met minimale functionaliteit. We gaan kennis maken met SQL Injection en gaan op zoek naar manieren om dit te voorkomen. Let's code!

Benodigdheden voor deze les

- XAMPP
- Sublime Text
- PhpMyAdmin DB
- Browser

> **Opdracht 1**: Wat is SQL?

Step-by-step implementatie login form met SQL injectie

Tijdens deze les gaan we een database opzetten in PhpMyAdmin. Om dit te kunnen doen gaan we eerst XAMPP **[1]** downloaden zodat we php kunnen gebruiken, want we hebben e

Voor deze opdracht gaan we HTML, PHP en MySQL gebruiken. Om deze tools te gebruiken, hebben we een server nodig. Daarom beginnen we met het downloaden van XAMPP **[1]**. Als je XAMPP eenmaal hebt gedownload, open je XAMPP en start je de Apache - en SQL server. Nu kunnen we beginnen met het opzetten van onze database in PhpMyAdmin.

PhpMyAdmin

Open je browser en navigeer naar http://127.0.0.1/phpmyadmin/ . Maak een nieuwe database aan met daarin een tabel *user*. Dit tabel heeft 3 kolommen, namelijk:

- id
- username
- password

Insert op minstens 2 entries in tabel user.

Code

Nu onze servers eenmaal draaien, kunnen we aan de slag! Open de xampp-folder in je document browser en open de folder *htdocs*. Maak een nieuwe folder genaamd *veilig programmeren SQL injecties* aan in folder htdocs.

Open de Sublime Text editor en maak de onderstaande drie files aan en sla deze op in folder *veilig programmeren SQL injeties*:

1. index.html
2. login.php
3. db.php

De onderstaande secties leggen uit wat er wordt verwacht van je code.

index.html

De index.html file is, zoals de extensie van de file al aangeeft, een HTML document. Zorg ervoor dat dit document de HTML-skeleton bevat. De body tag van dit document moet een form bevatten, bestaande uit twee input-fields en een button.

Het form moet gebruik maken van de login.php file.

Let op! Zorg ervoor dat het wachtwoord veld het ingevoerde wachtwoord niet toont!

Als je de HTML-skeleton en form af hebt, zou je deze moeten kunnen zien in je browser. Dat doe je zo:

1. Open je index.html file
2. Click met je rechter muisknop in de file
3. Selecteer "Open in browser"

Merk op dat de zoekbalk een file-path toont. Klik in je zoekbalk en vervang het stukje `file:///C:/xampp/htdocs` met `localhost`. Deze stap is nodig om PHP te kunnen gebruiken.

login.php

De login file is een php document. Hieronder is een code-snippet toegevoegd. Neem deze over in je eigen login.php-file en werk de todo's uit:

```
<?php

include "db.php";

// todo 1: print de ingevoerde username en wachtwoord op aparte regels. Geef duidelijk aan welk van de twee de username en het wachtwoord is.

// todo 2: maak twee variabele aan en sla de ingevoerde username en wachtwoord op in deze variabelen.

$myConn = new DB;

// todo 3: include de variabele met de username in de onderstaande query, zodat deze alle data kan ophalen voor de ingevoerde username.
$query = "SELECT * FROM user WHERE ";

$result = $myConn->executeSQL($query);

// todo 4: vermeldt wat de datatype van variabele $result is. Dit kun je met behulp van een ingebouwde php functie doen.

if (!empty($result)) {
    echo "<br> Login as $username <br>";
    // todo 5: let uit wat de ingebouwde php functie print_r() doet en gebruik het om de result-variabele te printen.
} else {
    echo "<br> Invalid login! <br>";
}

?>
```

db.php

De db.php file bevat een database (DB) class. Deze zorgt voor een database connectie. Neem de onderstaande code-snippet over in je db.php file en werk de todo's uit.

```
<?php

// Define DB Params
// todo 1: zoek uit wat de host, user, password van je database en vul ze hieronder in om de connectie te kunnen maken met je db
define("DB_HOST", "");
define("DB_USER", "");
define("DB_PASS", "");

// todo 2: vul de naam van de database in op de plek van de empty string.
define("DB_NAME", "");

class DB{
protected $dbh;
protected $stmt;
    protected $resultSet;

public function __construct(){
$this->dbh = new PDO("mysql:host=".DB_HOST.";dbname=".DB_NAME, DB_USER, DB_PASS);
        $this->resultSet = [];
}

public function executeSQL($query){
        $this->stmt = $this->dbh->prepare($query);
$result = $this->stmt->execute();

if (!$result) {
    die('<pre>Oops, Error execute query '. $query .'</pre><br><pre>'.'Result code: '. $result .'</pre>');
   }

$row = $this->stmt->fetchAll();

if(!empty($row)){
$this->resultSet = $row;
```

```
        return $this->resultSet;
    }

    return $this->resultSet;
}
}

?>
```

Moment of truth

In het begin heb je in de *user* tabel (van je database) twee gebruikers aangemaakt. Voer de username en wachtwoord van deze gebruiker in en ga na of je de juiste gebruikers data wordt geprint op de pagina.

> **Opdracht 3:** Als je ervoor hebt kunnen zorgen dat de juiste data print, is het zover! Voer de volgende regel in voor de username, en leg uit wat je te zien krijgt: `' OR '1'='1`.

> **Opdracht 4:** Wat is de naam van de "techniek" die je in opdracht 3 hebt toegepast?

> **Opdracht 5:** Hoe zou je ervoor kunnen zorgen dat de regel uit opdracht 3 niet meer kan gebeuren?

Bronnen

[1] XAMPP: https://www.apachefriends.org/index.html

Revision #2
Created 2020-02-25 08:48:29 UTC
Updated 2020-02-25 10:18:58 UTC