

Web App Security Alerter

1. Scopul proiectului

Proiectul a avut ca scop realizarea unei aplicatii ce trimite in timp real notificari userului despre diferitele vulnerabilitati software aparute pentru o clasa de aplicatii. Pentru a vedea informatiile prezentate de catre aceasta aplicatie, userul trebuie sa se logheze sau sa-si creeze un cont daca nu are deja unul. Pagina principala este pagina unde sunt toate exploiturile preluate cu ajutorul unui API si oferite utilizatorului intr-un mod frumos si interactiv.

2. Etapele intermediare ale dezvoltarii proiectului

I. Realizarea fisei cerintelor pentru a putea dezvolta ideea de ansamblu a proiectului

a. Construirea diagramei use-case, care urmareste modelul fisei.

b. Construirea diagramei de clase - actorii vor fi clasele, iar scenariile de utilizare vor fi in mare parte transformate in metode.

II. Luarea deciziilor privind abordarea problemei

a. Alegerea limbajului de programare (PHP)

b. Alegerea design-ului website-ului - aceasta alegere se va face in concordanta cu nevoile utilizatorului si cerintele problemei : responsive, intuitiv, atragator si simplu de folosit.

c. Brainstorming privind securitatea aplicatiei web - abordarea situatiilor de genul SQL Injection ...

d. Documentare privind API-urile ce vor fi folosite, bibliotecile si framework-ul de dezvoltare.

e. Impartirea task-urilor pe module pentru dezvoltarea ulterioara.

III. Dezvoltare

a. Dezvoltarea front-end-ului (impartirea task-urilor pe module; fiecare dezvoltator va realiza cate un segment; header-ul si footer-ul vor fi template-uri

care vor fi incluse la fiecare pagina din website, schimbandu-se doar main-ul)

b. Construirea bazei de date (realizarea diagramei de clase, pachetelor, tabelelor)

c. Realizarea conexiunii între baza de date, front-end și back-end.

d. Dezvoltarea back-end-ului.

i. Instalarea server-ului Apache.

ii. Testarea conexiunii cu baza de date și front-end-ul.

iii. Scrierea codului.

IV. Integrarea: preluarea modulelor de la dezvoltatori și combinarea lor în vederea rezultării unei aplicații funcționale.

V. Testarea: după integrare, testarea va urmări eventualele bug-uri rezultate din integrare (poate unele module nu vor fi compatibile, anumite use-case-uri neacoperite, etc)

VI. Rezolvarea bug-urilor

Obs. Etapele V și VI se vor face ciclic, însemnând că după rezolvarea bug-urilor, se vor face un nou rând de teste deoarece există posibilitatea ca atunci când bug-urile vor fi, altele vor apărea.

3. Considerații preliminare privind structura (modelarea) datelor și proveniența lor (servicii web, API-uri publice)

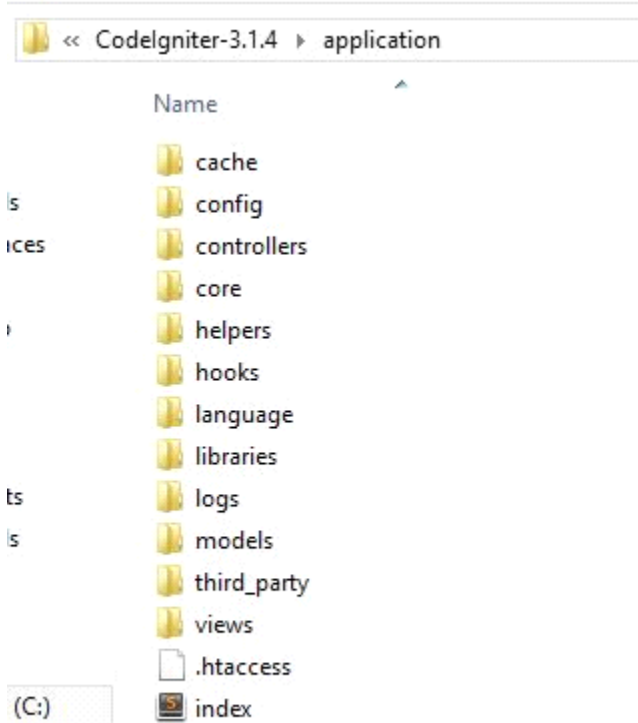
I. Folosirea API-ului website-ului exploit-db și a SearchSploit-ului : <https://github.com/offensive-security/exploit-database>

II. Dezvoltarea unui API public care va oferi soluții pentru exploit-urile găsite pentru a minimiza șansa vulnerabilității software a aplicației noastre și de asemenea, pentru a face mai ușoară comunicarea dintre client și server și pentru o dezvoltare ulterioară facilă a server-ului.

III. Arhitectura aplicație și BackEnd

Pentru dezvoltare arhitecturii și funcționalității aplicației a fost folosit frameworkul php CodeIgniter care a oferit multe avantaje în acest scop.

Aplicația respectă arhitectura MVC, fiind structurată în modele, view-uri și controllere.



Model:

```
class User extends CI_Model{  
    function __construct() {  
        $this->userTbl = 'USERS';  
    }  
  
    function getRows($params = array()){  
        $this->db->select('*');  
        $this->db->from($this->userTbl);  
  
        //fetch data by conditions  
        if(array_key_exists("conditions",$params)){  
            foreach ($params['conditions'] as $key => $value) {  
                $this->db->where($key,$value);  
            }  
        }  
  
        if(array_key_exists("email",$params)){
```

Controller:

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Users extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
        $this->load->library('form_validation');
        $this->load->model('user');
    }
}
```

View:

```
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width">
    <meta name="description" content="Real-time security alerter web application for exploits">
    <meta name="keywords" content="real-time, security, exploits, alert, advices, article, solutions for exploits">
    <meta author="author" content="Paula Carp, Ana Bodnar">
    <title>ASA | Sign Up</title>
    <link rel="stylesheet" href="<?php echo base_url(); ?>assets/css/signup.css" type="text/css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
    <link rel="icon" href="/image/icon.png">
</head>
<header>
    <h1><center><a href = "<?php echo base_url(); ?>index.php/main/index">ASA | Application <span class = "highlight">Security Alerter</span>
    </a></center></h1>
</header>
<body>
    <div class="container">
        <h1> Sign Up for Free </h1>
        <form action="" method="post">
            <div class="mandatory">
                <input type="text" name="name" placeholder="Full name" required="" value="<?php echo !empty($user['name'])? $user['name'] : ''; ?>"
                <?php echo form_error('name', '<span class="help-block">', '</span>'); ?>
            </div>
            <div class="mandatory">
                <input type="text" name="username" placeholder="Username" required="" value="<?php echo !empty($user['username'])? $user['username'] : ''; ?>"
                <?php echo form_error('username', '<span class="help-block">', '</span>'); ?>
            </div>
            <div class="mandatory">
                <input type="text" name="email" placeholder="Email Address" required="" value="<?php echo !empty($user['email'])? $user['email'] : ''; ?>"
            </div>
            <div class="mandatory">
                <input type="text" name="birthday" placeholder="Birthday" onfocus="(this.type='date')" required="" value="<?php echo !empty($user['birthday'])? $user['birthday'] : ''; ?>"
            </div>
            <div class="mandatory">
                <input type="text" name="country" placeholder="Country" required="" value="<?php echo !empty($user['country'])? $user['country'] : ''; ?>"
            </div>
            <div class="mandatory">
                <input type="text" name="city" placeholder="City" required="" value="<?php echo !empty($user['city'])? $user['city'] : ''; ?>"
            </div>
        </form>
    </div>
</body>
</html>
```

Diagram:

