Web dizajn i programiranje

Prof. dr.sc. Dragutin Kermek
Sveučilište u Zagrebu
Fakultet organizacije i informatike
Pavlinska 2, Varaždin 42000
dkermek@foi.hr

18. dio

Programiranje na strani poslužitelja

File Upload - preuzimanje lokalne datoteke
Funkcije datotečnog sustava
Virtualno vrijeme sustava
PHP sigurnost

File upload

Potreba za preuzimanjem datoteke s računala korisnika javlja u određenim situacijama.

Proces preuzimanja datoteke treba biti pod posebnom kontrolom zbog mogućih problema koji mogu proizaći iz njega. Prije svega misli se na zatrpavanje poslužitelja raznim "smećem". Drugi, ne manje značajan problem je moguće narušavanje sigurnosti poslužitelja kada se napad na sustav kanalizira kroz preuzetu datoteku.

PHP podržava upload/preuzimanje datoteke nakon podešavanja konfiguracijske datoteke php.ini

file_uploads = On
upload_max_filesize = 2M

Postoji mali skup funkcija za rad kod preuzimanja datoteke.

File upload / 2.

```
<form enctype="multipart/form-data" action="uploader.php"
method="post">
 <input type="hidden" name="MAX_FILE_SIZE" value="30000" />
 Preuzmi datoteku: <input name="userfile" type="file" /><br>
 <input type="submit" value="Pošalji" />
</form>
```

PHP omogućava da se postave dvije granice za veličinu datoteke koja se preuzima:

- **na razini sustava (upload_max_filesize u php.ini)**
- **na razini aplikacije (`MAX_FILE_SIZE` u obrascu).**

File uploader

```
<?php
$userfile = $_FILES['userfile']['tmp_name'];
$userfile_name = $_FILES['userfile']['name'];
$userfile_size = $_FILES['userfile']['size'];
$userfile_type = $_FILES['userfile']['type'];
$userfile_error = $_FILES['userfile']['error'];

if ($userfile_error > 0)  {
    echo 'Problem: ';
    switch ($userfile_error)  {
      case 1:  echo 'File exceeded upload_max_filesize';  break;
      case 2:  echo 'File exceeded max_file_size';  break;
      case 3:  echo 'File only partially uploaded';  break;
      case 4:  echo 'No file uploaded';  break;
    }
    exit;
  }
```


File uploader / 2.

```
if ($userfile_type != 'text/plain') {
    echo 'Problem: file is not plain text';
    exit;
}

$upfile = 'datoteke/' . $userfile_name;

if (is_uploaded_file($userfile))  {
   if (!move_uploaded_file($userfile, $upfile))   {
        echo 'Problem: Could not move file to destination directory';
        exit;
   }
}
 else {
  echo 'Problem: Possible file upload attack. Filename:
'.$userfile_name;
   exit;
}

echo 'File uploaded successfully<br /><br />';
?>
```


Programiranje na strani poslužitelja

File Upload - preuzimanje lokalne datoteke

Funkcije datotečnog sustava

Virtualno vrijeme sustava

PHP sigurnost

Funkcije datotečnog sustava

U radu aplikacija može se javiti potreba za preuzimanjem datoteke/a s izabranog direktorija na poslužitelju ili dinamički prikaz slikovnih datoteka kao galerije s izabranog direktorija na poslužitelju.

Za namjenu postoje PHP funkcije kao što su:

- **is_file**
- **is_dir**
- **is_executable**
- **is_writable**
- **is_readable**
- **opendir**
- **readdir**
- **fileperms**
- **mkdir**

Ispis direktorija

```
<!DOCTYPE html>
<html>
    <head>
        <title>Izbor direktorija</title>
        <meta charset="utf-8">
    </head>
    <body>
        <?php
        $sname = $_SERVER["SCRIPT_FILENAME"];
        $pos = strrpos($sname, "/");
        $dir = substr($sname, 0, $pos + 1);

        if ($dh = opendir($dir)) {
            while (($file = readdir($dh)) !== false) {
                if (is_dir($dir . $file) && ($file != '.' &&
                        $file != '..')) {
                    $dat[$file] = fileperms($dir . $file);
                }
            }
            closedir($dh);
            ksort($dat);
            ?>
```


Ispis direktorija / 2.

```
            <table style="border: 1px #000000 solid; background:
                    bisque">
                <tr><td>Direktorij</td><td>Pregled</td></tr>
                <?php
                foreach ($dat as $file => $perms) {
                    echo "<tr><td>$file</td><td>" .
                        "<a href='pregled.php?dir=$file'>Otvori</a>".
                        "</td></tr>";
                }
            }
            ?>
        </table>
    </body>
</html>"
```


Pregled galerije

```
<!DOCTYPE html>
<html>
    <head>
        <title>Pregled direktorija</title>
        <meta charset="utf-8">
    </head>
    <body>
        <?php
        if (!isset($_GET["dir"]) || $_GET["dir"] == "" ||
                    strpos($_GET["dir"], '..')) {
            header("Location: izbor.php");
            exit();
        }
        $sname = $_SERVER["SCRIPT_FILENAME"];
        $pos = strrpos($sname, "/");
        $dir = substr($sname, 0, $pos + 1) . $_GET["dir"] . "/";
        echo "Pregled galerije: " . $_GET["dir"];
```


Pregled galerije / 2.

```
        if ($dh = opendir($dir)) {
            while (($file = readdir($dh)) !== false) {
                if (is_file($dir . $file) && ($file != '.' &&
                        $file != '..')) {
                    $dat[$file] = fileperms($dir . $file);
                }
            }
            closedir($dh);

            ksort($dat);
            if (isset($dat)) {
               foreach ($dat as $file => $perms) {
                        $put = $_GET["dir"] . "/" . $file;
                        echo "<img src='$put'><br>";
                }
            }
        }
        ?>
    </table>
</body>
</html>
```


Programiranje na strani poslužitelja

File Upload - preuzimanje lokalne datoteke

Funkcije datotečnog sustava

Virtualno vrijeme sustava

PHP sigurnost

Virtualno vrijeme sustava

Određene radnje imaju svoje trajanje (npr. aktivacija korisničkog računa) a postoje pravila koja se odnose na prekoračenje roka.

Vrijeme poslužitelja ne može se mijenjati po potrebi, a kod testiranja nema vremena za čekanje da prođe zadano trajanje.

Jedino rješenje je uvođenje virtualnog vremena sustava koje se temelji na zadanom pomaku vremena (npr. u broju sati). Virtualno vrijeme se koristi u svakoj akciji u kojoj je potrebno vrijeme poslužitelja, a dobije se

```
$vrijeme_servera = time();
$vrijeme_sustava = $vrijeme_servera + ($sati * 60 * 60);
```


Virtualno vrijeme sustava

Postavljanje virtualnog vremena sustava smije obaviti samo korisnik s određenom ulogom (npr. administrator).

Postupak pripreme i korištenja virtualnog vremena sastoji se od sljedećih koraka:

- **-unos pomaka vremena**
- **-preuzimanje zadanog pomaka vremena te se upis u privatni dio aplikacije (npr. datoteka, tablica baze podataka i sl.)**
- **-postavljanje virtualnog vremena sustava**

Postavljanje pomaka vremena

Na WebDiP-u unos pomaka vremena obavlje se:

```
<?php
  header("Location: " .
        "http://barka.foi.hr/WebDiP/pomak_vremena/vrijeme.html");
?>
```

Upisani pomak vremena nalazi se u datoteci:

http://barka.foi.hr/WebDiP/pomak_vremena/pomak.php?format=xml

Sadržaj datoteke pomaka vremena ima strukturu:

```
<?xml version="1.0"?>
<WebDiP><vrijeme><pomak brojSati="3"/></vrijeme></WebDiP>
```


Korištenje virtualnog vremena sustava

```
<?php
$url = "http://barka.foi.hr/WebDiP/pomak_vremena/pomak.php?format=xml";

if (!($fp = fopen($url, 'r'))) {
    echo "Problem: nije moguće otvoriti url: " . $url;
    exit;
}

// XML data
$xml_string = fread($fp, 10000);
fclose($fp);

// create a DOM object from the XML data
$domdoc = new DOMDocument;
$domdoc->loadXML($xml_string);
```


Korištenje virtualnog vremena sustava

```
$params = $domdoc->getElementsByTagName('brojSati');
$sati = 0;

if ($params != NULL) {
    $sati = $params->item(0)->nodeValue;
}

$vrijeme_servera = time();
$vrijeme_sustava = $vrijeme_servera + ($sati * 60 * 60);
?>
```


Korištenje virtualnog vremena sustava

```
<!DOCTYPE html>
<html>
    <head>
        <title>Virtualno vrijeme sustava</title>
        <meta charset="utf-8">
    </head>
<?php
    echo "Stvarno vrijeme servera: " . date('d.m.Y H:i:s',
        $vrijeme_servera) . "<br>";
    echo "Virtualno vrijeme sustava: " . date('d.m.Y H:i:s',
        $vrijeme_sustava) . "<br>";
    ?>
</body>
</html>
```


Programiranje na strani poslužitelja

File Upload - preuzimanje lokalne datoteke
Funkcije datotečnog sustava
Virtualno vrijeme sustava
PHP sigurnost

Sigurnost web aplikacija

Sigurnosne točke:

- **-web poslužitelj (Apache)**
- **-aplikacijski poslužitelj (PHP)**
- **-poslužitelj baze podataka (MySQL)**
- **-ostali poslužitelji (LDAP, ...)**
- **-komunikacijski kanal**

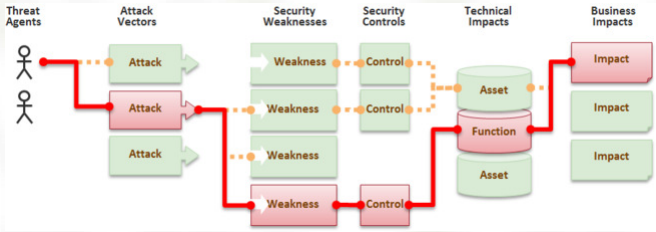
OWASP

Open Web Application Security Project (OWASP) - svjetska neprofitna organizacija koja je usmjerena na poboljšanje sigurnosti programske podrške.

Misija OWASP je da se učini programsku podršku vidljivijom tako da pojedinci i organizacije u svijetu mogu donositi odluke na bazi informacija o stvarnim sigurnosnim rizicima programske podrške.

OWASP ne odobrava ili preporučuje komercijalne proizvode ili usluge ostavljajući da zajednica bude neovisna o dobavljaču uz zajedničku mudrost najboljih umova u svjetskoj sigurnosti programske podrške.

Sigurnosni rizici aplikacija



OWASP Metodologija procjene rizika

Koraci:

1. **identificiranje rizika**
2. **faktori za procjenu vjerojatnosti**
3. **faktori za procjenu napada**
4. **utvrđivanje ozbiljnosti rizika**
5. **odlučivanje što popraviti**
6. **prilagođavanje svog modela procjene rizika**

Sigurnosni rizici aplikacija

Threat Agents	Attack Vectors	Weakness Prevalence	Weakness Detectability	Technical Impacts	Business Impacts
App Specific	EASY	WIDESPREAD	EASY	SEVERE	App / Business Specific
	AVERAGE	COMMON	AVERAGE	MODERATE	
	DIFFICULT	UNCOMMON	DIFFICULT	MINOR	

OWASP Top 10 vrsta napada - 2013

OWASP Top 10 vrsta napada - 2013 u usporedbi s 2010:

- A1 Injektiranje (SQL, LDAP, OS)
- A2 Prekinuta autentikacija i upravljanje sesijom
- A3 Izvršavanje skripte s drugog računala (XSS) (2010-A2)
- A4 Nesigurno direktno referenciranje objekta
- A5 Neispravna sigurnosna konfiguracija (2010-A6)
- A6 Otkrivanje osjetljivih podataka (2010-A7 Nesigurno kriptografsko spremište i 2010-A9 Nedovoljna zaštita transportnog sloja, spojeni su u 2013-A6)
- A7 Nedostatak funkcijske razine kontrole pristupa (preimenovano/prošireno od 2010-A8 Pogreška u ograničenju URL pristupa)
- A8 Krivotvorenja zahtjeva s drugog računala (CSRF) (2010-A5)
- A9 Korištenje komponenata s poznatim ranjivostima (novi ali je bio dio 2010-A6 – Neispravna sigurnosna konfiguracija)
- A10 Neprovjerena preusmjeravanja i prosljeđivanja

https://www.owasp.org/index.php/Top_10_2013-Top_10

OWASP Top 10 vrsta napada - 2017

OWASP Top 10 vrsta napada - 2017 u usporedbi s 2013:

- A1 Injektiranje (SQL, LDAP, OS)
- A2 Prekinuta autentikacija i upravljanje sesijom
- A3 Izvršavanje skripte s drugog računala (XSS)
- A4 Pekinuta konstrola pristupa (spojeno Nesigurno direktno referenciranje objekta A4 iz 2013 i Nedostatak funkcijske razine kontrole pristupa A7 iz 2013)
- A5 Neispravna sigurnosna konfiguracija
- A6 Otkrivanje osjetljivih podataka
- A7 Nedovoljna zaštita od napada (NOVO)
- A8 Krivotvorenja zahtjeva s drugog računala (CSRF)
- A9 Korištenje komponenata s poznatim ranjivostima
- A10 Slabo zaštićen API (NOVO)

https://github.com/OWASP/Top10/raw/master/2017/OWASP%20Top%2010%20-%202017%20RC1-English.pdf

Korištena i dodatna literatura

- http://php.net/manual/en/security.php
- https://www.owasp.org/index.php/PHP_Security_Cheat_Sheet
- http://phpmaster.com/top-10-php-security-vulnerabilities/