iranphp articles

عنوان مقاله : طراحي يك برنامه لاگين امن و حرفه اي
نگارنده : .....
آدرس پست الکترونیک : .....
تاریخ نگارش : .....

چگونه یک برنامه لاگین امن و حرفه ای طراحی کنید:

سرفصل‌ها :

۱ـ این اسکریپت چگونه کار می کند | پیشنیازها | الگوی بانک اطلاعاتی

۲ - اتصال به بانک اطلاعاتی | متغیرهای جلسات

۳ـ هسته اسکریپت | ورود کاربران

۴ـ کنترل مداوم اعتبار ورود | اطمینان از معتبر بودن اطلاعات جلسات

این آموزش در سطح متوسط طراحی گردیده و به شما خواهد آموخت که چگونه یک برنامه لاگین امن و حرفه ای طراحی کنید . بزودی یاد می گیریم که چگونه با استفاده از توابع کوکی ها دسترسی به جلسات را قانونی کنیم و از سرقت جلسات جلوگیری کنیم .

۱ـ این اسکریپت چگونه کار می کند ؟

در این قسمت می خواهم برایتان دلیل انتخاب این متد برای لاگین امن را برایتان توضیح دهم فراموش نکنید که اصولا امنیت قانون ندارد و شما نیز با استفاده از تجربیات و توانایی های هوشی خود باید به فکر توسعه و طراحی موارد مشابه و امن تر نمائید و به هیچ وجه به این حد بسنده نکنید :

کاربرانی که قادرند تا به سرور دست یابی داشته باشند می تواند جلسات معتبر لاگینها را از شاخه پیش فرض /tmp

که به منظور ذخیره سازی اطلاعات جلسات استفاده میشود مشاهده نمایند روش جلوگیری از این نوع حمله کنترل آی پی می باشد .

کسانی که بر روی میزبان شما سایتی دارند قادرند جلسات معتبر برای سایت شما تولید کنند توجه بفرمائید که برخی سایتها سرورهای اختصاصی دارند که به لحاظ امنیتی بسیار مطلوبترند مثل سایت پرشین بلاگ

برخی از اون آدمایی که به قول خودمون آخرشن و اند حکن می تونن شبکه رو بو بکشن و کوکی‌ها رو بقاپن

کنترل آی پی این مشکل رو هم حل می کنه

۱-۲ـ اما پیش نیازها

شما اول از همه باید بدونین که چه اطلاعاتی از کاربران قرار که در سایت ذخیره بشه در این مثال برای سهولت آموزش ساده ترین شکل ممکن رو فرض قرار دادم

ضمنن چون الان همه دیگه پی اچ پی ۴,۱ به بالا دارن من هم از استفاده کردم اگر می خواین که این اسکریپت رو روی نسخه های قدیمی تر اجرا کنین شما مجبورید که از `$GLOBALS['HTTP_SESSION_VARS']` استفاده کنید . `super global arrays`

۱-۳ الگوی بانک اطلاعاتی

این فقط یک مثال ساده با ساختاری مناسب برای مدیریت اگر که مایلید این مثال را برای کاربران ثبت نام شده استفاده کنید می توانید ستونهای را به دلخواه اضافه کنید

من الگوی بانک اطلاعاتی را که از مای اس کیو ال استفاده می کنه اینجا گذاشتم شما می تونین از دیگر بانکهای اطلاعاتی نیز استفاده کنید :

```
CREATE TABLE member (
id int NOT NULL auto_increment,
username varchar(20) NOT NULL default '',
password char(32) binary NOT NULL default '',
cookie char(32) binary NOT NULL default '',
session char(32) binary NOT NULL default '',
ip varchar(15) binary NOT NULL default '',
PRIMARY KEY (id),
UNIQUE KEY username (username)
);
```

فیلدهای پسورد و کوکی برای استفاده از ام دی فایو طراحی شدن کوکی مقدارش برای زمانیست که کاربر بخواهد اطلاعات برایش ذخیره شود و فیلدهای جلسه و آی پی برای جلسه آی دی و آی پی کاربر استفاده می شوند .

۲ـ اتصال به بانک اطلاعاتی

```
function &db_connect() {

require_once 'DB.php';

PEAR::setErrorHandling(PEAR_ERROR_DIE);

$db_host = 'localhost';
$db_user = 'root';
$db_pass = '';
$db_name = 'shaggy';

$dsn = "mysql://$db_user:$db_pass@unix+$db_host/$db_name";

$db = DB::connect($dsn);

$db->setFetchMode(DB_FETCHMODE_OBJECT);
return $db;

}
```

این تابع شما را به بانک اطلاعاتی متصل می کند و یک اشاره گر به شی بانک اطلاعاتی پیر باز می گرداند

۲-۲ متغیرهای جلسات :

برای سهولت در امر دستیابی به اطلاعات کاربران من اون رو یک متغیر جلسه ثبت می کنم ولی برای جلوگیری از پیغام خطا و همچنین ست کردن برخی پیش فرض ها از تابع زیر استفاده می کنم :

```
function session_defaults() {

$_SESSION['logged'] = false;
$_SESSION['uid'] = 0;
$_SESSION['username'] = '';
$_SESSION['cookie'] = 0;
$_SESSION['remember'] = false;
}
```

برای ست کردن یه مقادیر پیش فرض از تابع بالا و برای چک کردن از تابع زیر

```
if (!isset($_SESSION['uid']) ) {
session_defaults();
}
```

رو فرا خوانی کنیم`session_start`البته فراموش نمی کنیم که قبل از اینها باید تابع

۳ـ هسته اسکریپت :

برای ایجاد یکپارچگی ساده تر با دیگر اسکریپتها و ساخت مدوله شده تر هسته اسکریپت رو یک آبجکت با ظاهری خیلی ساده می سازم

```
class User {

var $db = null; // PEAR::DB pointer
var $failed = false; // failed login attempt
var $date; // current date GMT
var $id = 0; // the current user's id
```

```
function User(&$db) {

$this->db = $db;
$this->date = $GLOBALS['date'];

if ($_SESSION['logged']) {
$this->_checkSession();
} elseif ( isset($_COOKIE['mtwebLogin']) ) {
$this->_checkRemembered($_COOKIE['mtwebLogin']);

}

}
```

این کلاس که تعریف میشه آبجکت ما رو میسازه البته این کاملا مدوله شده نیست اما یک تاریخ مشکل بزرگی نیست و شما می تونین اونو با اسکریپتهایی که بقیه دوستان نوشتن به صورت شمسی تولید کنید در اینجا ما چنین چیزی رو می سازیم :

```
$date = gmdate("'Y-m-d'");
$db = db_connect();
$user = new User($db);
```

حالا برای روشن شدن هدف کد یعنی لاگین کردن تلاش می کنیم ما ابتدا کنترل می کنیم که آیا کاربر لاگین کرده یا نه اگر این کار رو کرده بود ما جلسات رو چک می کنیم(فراموش نکنین که این یک کد امنیتی) وگرنه یک کوکی رو نام گذاری می کنیم برای کنترل کردن این به ما اجازه می ده که بینندگان سایت رو شناسایی کنیم

۳-۱-لاگین کردن کاربران :

برای اجازه دادن به کاربران برای لاگین کردن شما باید یک فرم وب بسازید پس از اعتبار سنجی فرم شما می تونید اعتبار کاربر رو برای تائید اطلاعات وارد شده کنترل کنید که برای اینکار از

```
$user->_checkLogin('username', 'password', remember)
```

استفاده می کنیم خاطر نشان می کنیم که یوزر نیم و پسورد البته نباید ثابت باشند و ریممبر یک مقدار بولین است که به کاربر اجازه می دهد تا با درست قرار دادن مقدار آن لاگین خودکار را فعال بسازد

```
function _checkLogin($username, $password, $remember) {

$username = $this->db->quote($username);
$password = $this->db->quote(md5($password));

$sql = "SELECT * FROM member WHERE " .
"username = $username AND " .
"password = $password";

$result = $this->db->getRow($sql);

if ( is_object($result) ) {
$this->_setSession($result, $remember);
return true;
} else {
$this->failed = true;
$this->_logout();
return false;
}
```

```
}
```

تعریف تابع باید در مکانی کنار کلاس تعریف شده یوزر باشه مانند تمام کدهای پائین در تابع از متد PEAR::DB
استفاده کردم تا اطلاعات با امنیت کامل به بانک اطلاعاتی انتقال پیدا کنند و به صورت بی خطری نیز از آن رهای یابند و باز گردنند من از تابع ام دی فایو ترجیحا به جای توابع مای اسکیو ال استفاده کردم تا شما اگر مایل بودید بتوانید از بانکهای اطلاعاتی دیگر نیز استفاده کنید .
حلقه ور بهینه شده زیراکه یوزرنیم به صورت منفرد تعریف شده است
نیازی به کنترل خطاهای بانک اطلاعاتی نیست زیراکه خطاهای پیشفرض قبلا در بالا ست شدند
چنانچه آبجکت با رزالت بانک اطلاعاتی متچ شود لذا متغیر جلسات ست می شوند و مقدار ترو باز میگردد وگرنه
مقدار فلد با ترو برابر می گردد شما می تونین اینجا یک دستور کنترلی قرار دهید تا پیغام سقوط عملیات لاگین رو اعلام کنه و برای انجام لاگ اوت برای این بیننده کافیست تا session_defaults() را اجرا کنیم

۳-۳-وضع کردن جلسه :

```
function _setSession(&$values, $remember, $init = true) {

$this->id = $values->id;
$_SESSION['uid'] = $this->id;
$_SESSION['username'] = htmlspecialchars($values->username);
$_SESSION['cookie'] = $values->cookie;
$_SESSION['logged'] = true;

if ($remember) {

$this->updateCookie($values->cookie, true);

}

if ($init) {

$session = $this->db->quote(session_id());
$ip = $this->db->quote($_SERVER['REMOTE_ADDR']);

$sql = "UPDATE member SET session = $session, ip = $ip WHERE " .
"id = $this->id";
$this->db->query($sql);

}

}
```

این متد متغیر جلسه را ست می کند و همچنین اگر در خواست کوکی برای داشتن لاگین مسمتر (خودکار) ارسال شده باشد همچنین این متد یک پارامتر دارد که معین می کند که این بار اول لاگین کردن است یا نه (از طریق فرم یا کوکی) یا کنترل جلسه برای اولین بار نیست .

۴ـ لاگین خود کار :

اگر بینندگان در خواست کنند که کوکی ارسال بشه تا دفعات بعدی از لاگین کردن در هر مشاهده از سایت بپریداین دو متد به شما برای رسیدن به این مهم کمک خواهد کرد

```
function updateCookie($cookie, $save) {

$_SESSION['cookie'] = $cookie;
if ($save) {

$cookie = serialize(array($_SESSION['username'], $cookie) );
```

```
set_cookie('mtwebLogin', $cookie, time() + 31104000, '/directory/');
}

}
```

۴-۱-کنترل لاگین خود کار :

اگر کاربران لاگین خودکار را انتاخاب کرده باشند که به اسکریپت اجازه ذخیره کوکی را می دهد که کنترل می شه از طریق متد زیر

```
function _checkRemembered($cookie) {

list($username, $cookie) = @unserialize($cookie);
if (!$username or !$cookie) return;

$username = $this->db->quote($username);
$cookie = $this->db->quote($cookie);

$sql = "SELECT * FROM member WHERE " .
"(username = $username) AND (cookie = $cookie)";

$result = $this->db->getRow($sql);

if (is_object($result) ) {
$this->_setSession($result, true);
}

}
```

این تابع هرگز نباید توسط پیغام خطایی متوقف شود برای ساختن چیزهای امن تر با کوکی ها مقدار کوکی در کوکی ذخیر می شود نه پسورد کاربر یکی از این راه ها می تونه درخواست یک لغت عبور باشه برای ناحیه ای که به امنیت بیشتری نیاز دارد

۵-۵- مطمئن شدن از اعتبار جلسه :

```
function _checkSession() {

$username = $this->db->quote($_SESSION['username']);
$cookie = $this->db->quote($_SESSION['cookie']);
$session = $this->db->quote(session_id());
$ip = $this->db->quote($_SERVER['REMOTE_ADDR']);

$sql = "SELECT * FROM member WHERE " .
"(username = $username) AND (cookie = $cookie) AND " .
"(session = $session) AND (ip = $ip)";

$result = $this->db->getRow($sql);

if (is_object($result) ) {
$this->_setSession($result, false, false);
} else {
$this->_logout();
}

}
```

پوف بالاخره آخرین قسمت کار ما کنترل می کنیم که آیا کوکی ذخیره شده در جلسه درست هست یا نه جلسه آی دی و آی پی کاربر ا یک پارامتر که اجازه می ده که بفهمیم که این اولین بار لاگین کردن در سیستم هست بنابراین `setSession` فراخوانی مقدار آی پی و آی دی در جلسه بروز رسانی نشود که در بقیه موارد بطور معمول انجام می شود.