ZSDGMDZFGW...

Zehn Sicherheitsprobleme, die gerne mit dem ZendFramework gebaut werden

Ben Fuhrmannek • #phpug-köln • 2.10.2009

SektionEins

Über mich...

- Informatiker

- Entwickler

- IT Security

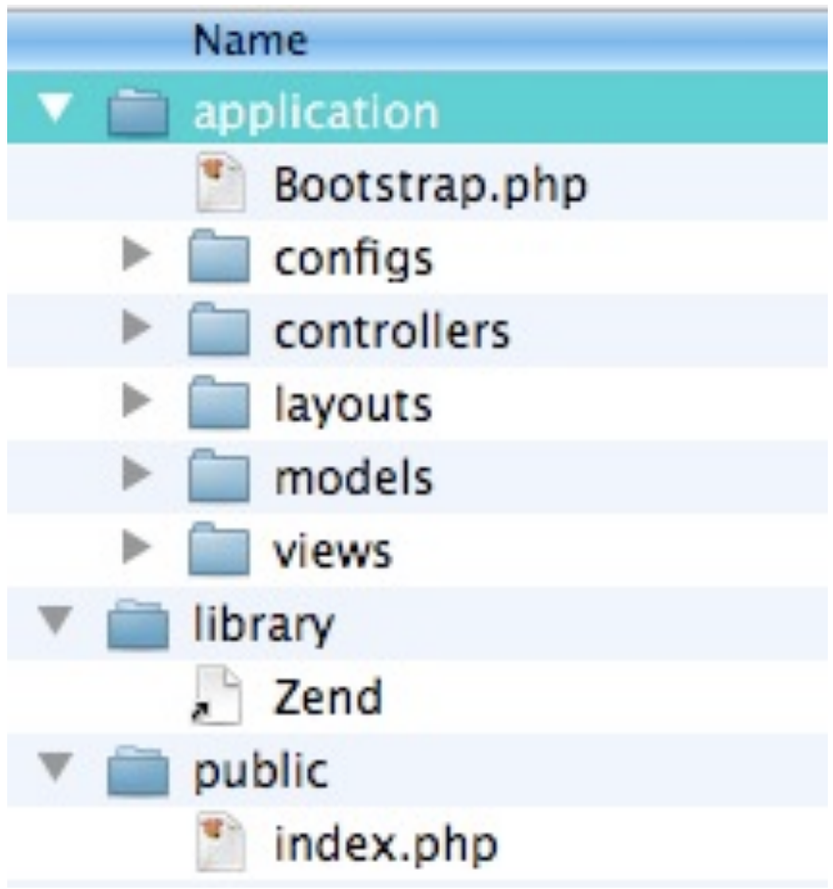
SektionEins

TOC

- Aufbau ZF
- Problem 1 bis 10

Aufbau eines ZF-Projekts

- Public
- Bootstrap
- Config
- Models
- Views
- Controllers



Aufbau - public/index.php

```
<?php
// ... set_include_path usw.

/** Zend_Application */
require_once 'Zend/Application.php';

// Create application, bootstrap, and run
$application = new Zend_Application(
    APPLICATION_ENV,
    APPLICATION_PATH . '/configs/application.ini'
);

$application->bootstrap();
$application->run();
```

Aufbau - application/Bootstrap.php

```php
<?php

class Bootstrap extends
Zend_Application_Bootstrap_Bootstrap
{
  protected function _initFoo()
  {
    // ...
  }
}
```

Problem 1 - SQL-Injections

Problematischer Code:

```
...

$sql =
   "SELECT name FROM users
    WHERE username = '$username'";

$db->query($sql);

...
```

Problem 1 - SQL-Injections

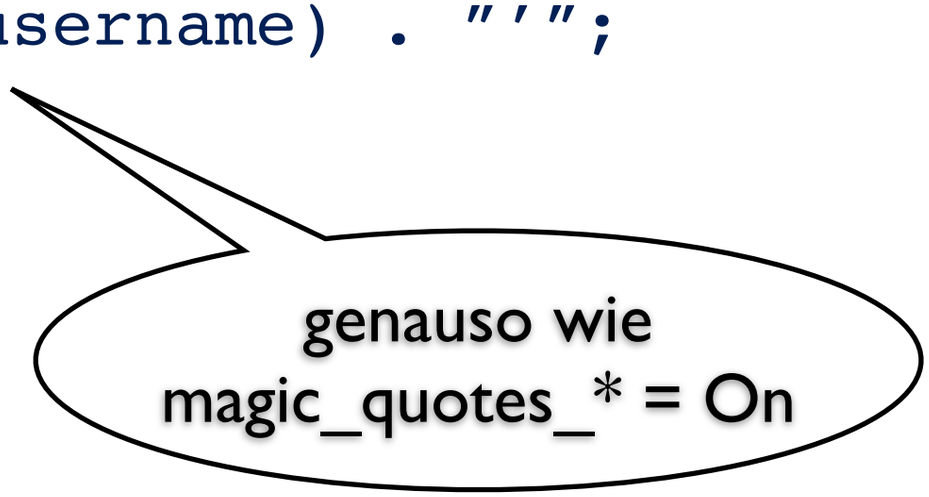
Genauso problematischer Code:

```
...

$sql =
  “SELECT name FROM users WHERE
  username = ‘“ . addslashes($username) . “’”;

$db->query($sql);

...
```



Problem 1 - SQL-Injections

OK

```
...

$sql =
  “SELECT name FROM users WHERE
  username = “ . $db->quote($username);

$db->query($sql);

...

// $db ist ein Zend_Db_Adapter
```

siehe auch quoteInto und quoteIdentifier

Problem 1 - SQL-Injections

Auch OK, aber nicht die ZF

```
...

$escaped_username =
    mysql_real_escape_string($username);

$sql =
  “SELECT name FROM users WHERE
  username = ‘$escaped_username’“;

$db->query($sql);

...
```

Problem 1 - SQL-Injections

Schon wieder problematisch:

```
...

$sql =
   “SELECT foo FROM bar WHERE
    foobar LIKE “ . $db->quote($foobar);

$db->query($sql);

...
```

ZF Forms (Überblick)

Controller

```
$form = new Zend_Form();
$form->setAction($this->view->url())
     ->setMethod('post');

$form->addElement('submit', 'login', array(
                'label' => 'Login'));

// ... noch mehr addElement ...

$this->view->form = $form;
```

View

```
<?php echo $this->form->render(); ?>
```

Problem 2: Forms

Problematisches Formular:

```
$form = new Zend_Form();
$form->setAction($this->view->url())
     ->setMethod('get');

$form->addElement('submit', 'action', array(
        'label' => 'Passwort-Reset'));
```

Problem 2: Forms / CSRF

OK

```
$form = new Zend_Form();
$form->setAction($this->view->url())
     ->setMethod('get');

$form->addElement('hash', 'csrftoken', array(
        'required' => true,
        'salt' => '12345',
));

$form->addElement('submit', 'action', array(
        'label' => 'Passwort-Reset'));
```

Problem 3: Forms

Vielleicht problematisches Formular:

```
$form->addElement('text', 'username',
  array('label' => 'Username'));

$form->addElement('password', 'password',
  array('label' => 'Password'));

$form->addElement('submit', 'login', array(
        'label' => 'Login'));
```

Problem 3: Forms / Validators

Problematisches Formular:

```
$form->addElement('text', 'username', array(
            'label' => 'Username',
            'validators' => array(
array('Regex', true, array('/^[a-zA-Z0-9_.@-]+$/D')),
array('StringLength', true, array(1, 50)) ) ));

$form->addElement('password', 'password', array(
            'label' => 'Password',
            'validators' => array(
    array('StringLength', true, array(1, 255))) ));

$form->addElement('hash', 'csrftoken', array(
            'required' => true,
            'salt' => '12345'));

$form->addElement('submit', 'login', array(
            'label' => 'Login'));
```

Vorsicht REGEX!!

Problem 3: Forms / ‘required’

OK

```
$form->addElement('text', 'username', array(
            'label' => 'Username',
            'required' => true,
            'validators' => array(
array('Regex', true, array('/^[a-zA-Z0-9_.@-]+$/')),
array('StringLength', true, array(1, 50)) ) ));
```

```
$form->addElement('password', 'password', array(
            'label' => 'Password',
            'required' => true,
            'validators' => array(
  array('StringLength', true, array(1, 255))) ));

$form->addElement('hash', 'csrftoken', array(
            'required' => true,
            'salt' => '12345'));
```

Problem 4: Views / XSS (trivial)

Controller

```
$request = $this->getRequest();

$this->view->lang =
  $request->getParam('lang');
```

View

```
<?php echo $this->lang; ?>

<?php echo $this->escape($this->lang); ?>
```

Problem 4: Views / XSS (URL)

Controller

```
$request = $this->getRequest();

$this->view->lang =
  $request->getParam('lang');
```

View

```
<a href="...?lang=<?php echo $this->lang; ?
>">boo</a>

<a href="...?lang=<?php
    echo urlencode($this->lang); ?>">boo</a>
```

siehe auch http_build_query

Problem 4: Views / XSS (falsches Escaping)

Controller

```
$request = $this->getRequest();

$this->view->lang =
  $request->getParam('lang');
```

View

```
<script language="javascript">
var lang = '<?php
  echo htmlspecialchars($this->lang); ?>';
</script>
```

Problem 4: Views / Whitelisting!

Controller

```
$request = $this->getRequest();

$lang = $request->getParam('lang');

if (!in_array($lang, array('en', 'de'))
  $lang = 'en';

$this->view->lang = $lang;
```

View

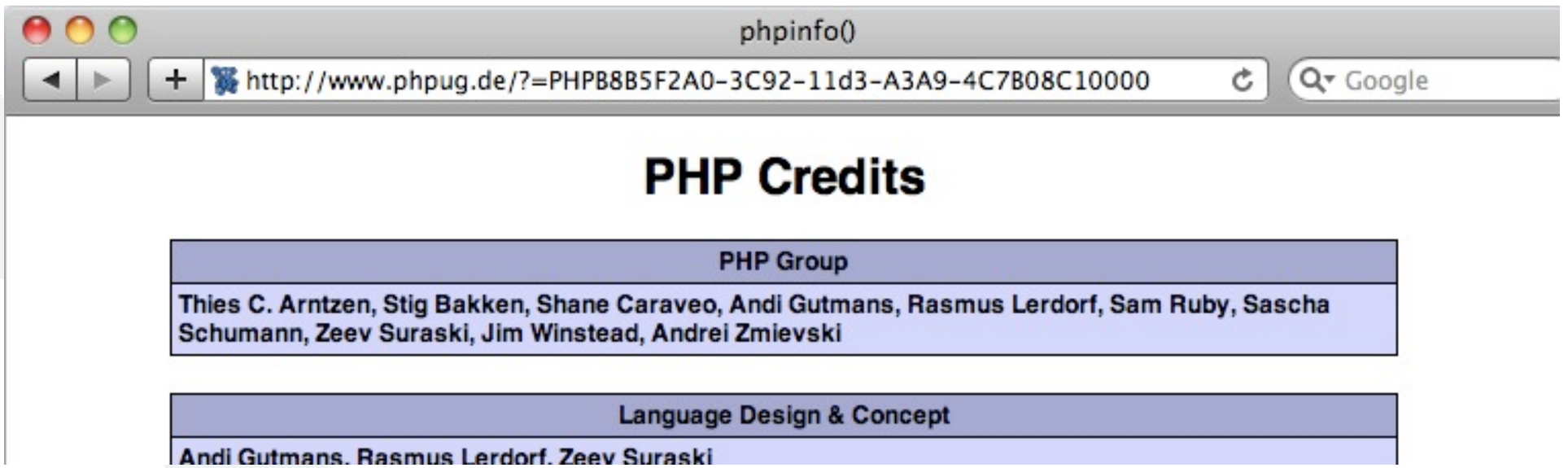
```
<?php echo $this->lang; ?>
```

Problem 5: Config / Empfehlungen

- register_globals = Off (php.ini)
- expose_php = Off (php.ini)
- ServerTokens Prod (apache config)
- ServerSignature Off (apache config)

Problem 5: Config / eastereggs

expose_php = On



Problem 5: Config / Dateisystem

- chown htdocs .../public
  (www / htdocs privilege separation)
- ZendFramework (oder andere Frameworks) nie im public Verzeichnis

Problem 5: Config / generelle Probleme

- security = on, aber nicht nachgedacht (jede config, z.B. 수호신 geladen, aber nicht konfiguriert)
- keine oder schlechte DB-Passwörter
- development / staging / production (ZF - siehe .htaccess: SetEnv APPLICATION_ENV production)

Problem 6: Session Handling (Login)

Controller

```
...

$auth = Zend_Auth::getInstance();

$adapter = new Zend_Auth_Adapter_DbTable(
Zend_Db_Table_Abstract::getDefaultAdapter(),
  'users', 'username', 'password',
  'MD5(?) AND confirmed');

$adapter
  ->setIdentity($form->getValue('username'))
  ->setCredential($form->getValue('password'));

$result = $auth->authenticate($adapter);
```

Fortsetzung nächste Seite

Problem 6: Session Handling (Login)

Controller (problematisch)

```
...

if ($result->isValid()) {
  // successful login

  $storage = $auth->getStorage();
  $storage->write(
    $adapter->getResultRowObject(array(
      'id', 'username', 'orga', 'admin')));

} else {
  // invalid login
  ...
}
```

Problem 6: Session Handling (Login)

Controller (ok)

```
...

if ($result->isValid()) {
  // successful login
  Zend_Session::regenerateId();
  $storage = $auth->getStorage();
  $storage->write(
    $adapter->getResultRowObject(array(
      'id', 'username', 'orga', 'admin')));

} else {
  // invalid login
  ...
}
```

Problem 6: Session Handling (Logout)

Controller (problematisch)

```
public function logoutAction()
{
  $this->_redirect('/');
}
```

Problem 6: Session Handling (Logout)

Controller (ok)

```
public function logoutAction()
{
  Zend_Auth::getInstance()->clearIdentity();
  Zend_Session::destroy(true, true);

  $this->_redirect('/');
}
```

Problem 7: Verschlüsselung / Session Cookies

Bootstrap

```
protected function _initSession()
{
  $cookieParams = session_get_cookie_params();

  session_set_cookie_params(
    $cookieParams['lifetime'], // lifetime in
seconds
    "/guru2", // path
    $cookieParams['domain'], // domain
    true, // secure flag
    true // http-only flag
  );

  session_name("GURU_SID");
}
```

Problem 7: Verschlüsselung / Apache

.htaccess

```
RewriteCond %{HTTPS} !=on
RewriteRule ^.*$ - [NC,C]
RewriteCond %{REQUEST_URI} !^/index
RewriteRule ^.*$ - [NC,F,L]
```

apache config

```
SSLCipherSuite HIGH:MEDIUM:!SSLv2:!EXP:!aNULL:!eNULL
```

Problem 8: Encoding & Charset

View (problematisch)

```
<meta http-equiv="Content-Type"
      content="text/html; charset=UTF8" />
```

Problem 8: Encoding & Charset

View (ok)

```
<meta http-equiv="Content-Type"
      content="text/html; charset=UTF-8" />
```

apache config

```
AddDefaultCharset utf-8
```

Problem 8: Encoding / DB

application.ini

```
resources.db.params.charset   = utf8
```

Problem 9: Login Brute-Forcing

Controller (problematisch)

```
...

$result = $auth->authenticate($adapter);

if ($result->isValid()) {
    // successful login
    // ... (viel Zeit)
} else {
    // invalid login
    // ... (wenig Zeit)
}
```

Problem 9: Login Brute-Forcing

Controller (ok)

```
...

$result = $auth->authenticate($adapter);

if ($result->isValid()) {
    // successful login
    // ... (viel Zeit)
} else {
    // invalid login
    // wait
    usleep(1000*(750 + rand(0, 2000)));
    ...
}
```

Problem 10: Unsinn

Backdoors

```
...

if ($password == ‘secret’) {
  // backdoor login
  ...
}
```

Problem 10: Unsinn

Eigene Crypto

```
...

$warenkorb_cookie = rot23($warenkorb);

...
```

Problem 10 1/2: Alles andere

- logische Fehler
- XSS in Translations
- Format Strings - z.B. sprintf($_GET['foo']);
- MVC Mixup - z.B. SQL im View
- ...

Kommentare?