Sichere Ablage von Zugangsdaten (nicht nur) zu Datenbanken in PHP-Projekten

Eine immer wieder gestellte Frage im Zusammenhang mit PHP-Projekten ist, die, wie Datenbankzugänge (Benutzernamen und Passwörter) „sicher“ abgelegt werden können. Hier machen längst nicht nur Anfänger Fehler.

Problembeschreibung:

Der lesende und schreibende Zugriff auf Datenbanken ist Bestandteil vieler PHP-Projekte. Üblicherweise wird der Zugriff wie folgt hergestellt:

```
<?php
$strDbHost='localhost'; # Datenbankserver (IP oder Hostname; String)
$strDbUser='Benutzer';  # Dein Benutzername (Name; String)
$strDbPass='Passwort';  # Dein Passwort (String)
$strDbBase='Datenbank'; # Deine Datenbank, (Name; String)
$resDb=mysql_pconnect($strDbHost,$strDbUser,$strDbPass)
    or die ("Keine Verbindung zu MySQL.");
$result=mysql_query('USE `' . $strDbBase, $resDb . '`')
    or die ('Datenbank ist nicht wählbar.');
?>
```

Auf diese Datei wird sich im folgenden auch bezogen

Dieses ist in jedem Skript erforderlich, in welchem der Datenbankzugriff benötigt wird. Das ist nicht ganz ungefährlich. Zum Beispiel legen viele Editoren Backups der geöffneten Datei ab und schnell gelangt eine solche Kopie auf den Server. Wenn z.B. „kate“ als Editor verwendet wird, dann haben diese Backups eine zusätzliche Tilde am Ende des Dateinamens. **Hierdurch besteht Gefahr!**

Mit dem Aufruf

```
http://host/index.php~
```

kann sich ein Angreifer den PHP-Quelltext anzeigen lassen und schon ist er im Besitz Deines Benutzernamens und Deines Passwortes!

Natürlich kann man hier Abhilfe schaffen, indem man dafür sorgt, dass diese Backups andere Namen bekommen, hier geht es aber um eine grundsätzliche Lösung.

Inkludieren des Zugriffs und Dateiname

Obigen Quelltext können und sollten Sie einfach in einer externen Datei bereithalten. Schon mal um bei künftigen Änderungen viel Zeit und Nerven zu sparen. Hierbei sind einige wenige Dinge zu beachten:

1. Der Dateiname

Der am häufigsten verwendete Apache Webserver weigert sich in der Standardkonfiguration Dateien auszuliefern, die mit „.ht“ beginnen. Zum Beispiel die Datei „.htaccess“. Das sollte genutzt werden:

Speichern Sie den obigen Inhalt in einer Datei mit dem Name „.ht_db_open.php“ (wir bleiben bei diesem Name), so wird der Server (so lange niemand entgegen jedem Grundsatz der Vernunft diese Einstellung nicht

©Jörg Reinholz, fastix WebDesign & Consult, Kassel, http://www.fastix.de
Dieser Aufsatz darf ganz oder teilweise im Rahmen des „fair-use“ in jeder Form (gedruckt, digital, eingebunden in andere Werke) weitergegeben werden, so lange hierbei der Autor wie oben angegeben genannt wird.

ändert) diese nicht ausliefern, sondern den versuchten Abruf mit dem Statuscode 404 und einer Fehlermeldung quittieren. Der führende Punkt vor dem Rest des Dateinamens ist wichtig!

2. Vermeide „short tags“!

Vermeide unbedingt „short-tags“. Das bedeutet:

```
<?
# So nicht!
# wenn in der (verwendeten!) php.ini „short_open_tag“ auf „0“ gesetzt
# ist oder wird, dann liefert der Webserver Deinen PHP-Code als Text
# aus! Damit natürlich auch die Zugangsdaten.
?>
```

Besser:

```
<?php
# So kann dies jedenfalls so lange nicht passieren, wie PHP erkannt
# und als solches interpretiert (ausgeführt) wird.
?>
```

3. Das Verzeichnis

Der sicherste Weg ist zweifelsfrei, diese Datei in einem Verzeichnis abzulegen, die außerhalb des Serverroots liegt. Eine solche Standardkonfiguration sähe so aus:

```
/
|
...
- srv
   |
   - www
      |
      - cgi-bin       (spez. Verzeichnis für Ausführbares)
      - htdocs        (Serverroot / $_SERVER['Document_ROOT'])
      - conf          (Konfigurationverzeichnis außerhalb)
          |
          - .ht_db_open.php  (Datei mit dem Zugangsdaten)
```

Von außerhalb des Document-Roots liefert zumindest der Webserver keine Daten aus. Das genügt normalerweise, dennoch ist es eine gute Idee, weiter zu lesen.

©Jörg Reinholz, fastix WebDesign & Consult, Kassel, http://www.fastix.de
Dieser Aufsatz darf ganz oder teilweise im Rahmen des „fair-use“ in jeder Form (gedruckt, digital, eingebunden in andere Werke) weitergegeben werden, so lange hierbei der Autor wie oben angegeben genannt wird.

4. Das Verzeichnis bei Massenhostern

Leider ist dies nur möglich, wenn Sie einen eigenen (virtuellen) Server verwalten. Oft ist es so, dass der „Webspace“, der bei Massenhostern gemietet werden kann diese schöne Möglichkeit, eines Verzeichnisses außerhalb des Document-Roots (hier /srv/www/htdocs) nicht unterstützt.

Bei einem solchen Massenhoster legen Sie in Ihrem Document-Root ein Unterverzeichnis an:

```
- ~        (User-Dir, zugleich Serverroot / $_SERVER['Document_Root'])
  |
  - conf          (Konfigurationverzeichnis innerhalb)
     |
     - .ht_db_open.php  (Datei mit dem Zugangsdaten)
     - .htacess         (Beschreibung folgt)
     - index.html       (Beschreibung folgt)
```

In dieses Verzeichnis legen Sie Ihre Datei „.ht_db_open.php“ ab. Bei einem Aufruf würde diese, wie bereits beschrieben weder ausgeführt noch angezeigt. Selbst wenn der Schutz durch den mit „.ht“ beginnenden Dateiname nicht bestände, so würde diese ausgeführt und so nichts relevantes anzeigen, aber das brächte nur ein geringes Maß an Sicherheit.

5. Die Datei .htaccess

Wenn Sie gezwungen sind auf ein solches Verzeichnis zurückzugreifen, dann sollten Sie eine Datei mit dem Name „.htaccess“ in Ihrem „conf“- Verzeichnis ablegen. Als Inhalt genügt im Regelfall eine Zeile:

```
deny from all
```

Hierdurch wird verhindert, dass der Apache-Webserver irgendeine Datei aus dem Verzeichnis ausliefert. Der letzte Schritt dient als allerletzte Versicherung und hilft bestenfalls zu verdunkeln: Die Datei „index.html“:

```
<html>
<header>
<title>Error 404 (Forbidden)</title>
</header>
     <body>
                <h1>Error: 404</h1>
                <p>Hier gibt es nichts zu sehen.</p>
          </body>
</html>
```

Das genügt jedenfalls so lange die Datei 'index.html' in der Serverkonfiguration als jene Datei bestimmt ist, die beim Aufruf eines Verzeichnisses ausgeliefert wird. Der Rest mag Deiner Phantasie überlassen bleiben, aber diese Datei bekommt (hoffentlich) ohnehin nie jemand zu Gesicht.

©Jörg Reinholz, fastix WebDesign & Consult, Kassel, http://www.fastix.de
Dieser Aufsatz darf ganz oder teilweise im Rahmen des „fair-use“ in jeder Form (gedruckt, digital, eingebunden in andere Werke) weitergegeben werden, so lange hierbei der Autor wie oben angegeben genannt wird.

Natürlich sollte der Zugriff getestet werden:

```
http://DeinHost/conf/     oder: http://DeinHost/conf/.ht_db_open.php
```

sollten zur Anzeige der Standard-Fehlerseite ihres Servers für den Fehler '404 (Forbidden)' führen.

Dateirechte:

Vergeben die Dateirechte restriktiv. Verzeichnisse müssen von Dir als dem Eigentümer gelesen, geändert und betreten werden dürfen, der Rest der Welt braucht für Verzeichnisse das Leserecht und das Recht es zu betreten:

```
chmod 755 /srv/www/htdocs/conf
```

oder:

```
chmod 755 /srv/www/conf
```

ist zielführend.

Die Konfigurationsdatei „.ht_db_open“, die Datei „.htaccess“ soll von Dir geändert und von jedermann gelesen werden können:

```
chmod 644 /srv/www/htdocs/conf/.ht_db_open
chmod 644 /srv/www/htdocs/conf/.htaccess
chmod 644 /srv/www/htdocs/conf/index.html
```

oder:

```
chmod 644 /srv/www/conf/.ht_db_open
chmod 644 /srv/www/conf/.htaccess
chmod 644 /srv/www/conf/index.html
```

ist hier zielführend.

6. Einbinden der Datei '.ht_db_open.php'

In Ihren Skripten binden Sie die Datei wie folgt ein:

```
<?php
$strConfDir=$_SERVER['DOCUMENT_ROOT'].'/../conf/';
/*
wenn '.ht_db_open.php' innerhalb von Document_Root abgelegt ist, die
folgende Zeile durch Entfernen des Raute-Zeichens aktivieren:
*/
# $strConfDir=$_SERVER['DOCUMENT_ROOT'].'/conf/';

if (! isfile ($strConfDir . '.ht_db_open.php')) {
     die ('Fatal: Die Datenbankkonfigurationsdatei ist nicht an der
     angegebenen Stelle.');
}
```

©Jörg Reinholz, fastix WebDesign & Consult, Kassel, http://www.fastix.de
Dieser Aufsatz darf ganz oder teilweise im Rahmen des „fair-use“ in jeder Form (gedruckt, digital, eingebunden in andere Werke) weitergegeben werden, so lange hierbei der Autor wie oben angegeben genannt wird.

```
if (! isreable($strConfDir . '.ht_db_open.php')) {
     die ('Fatal: Die Datenbankkonfigurationsdatei ist nicht lesbar.
     Bitte überprüfen Sie die Rechte');
}
require_once $strConfDir.'.ht_db_open.php';
# ... weiter mit dem Programm
?>
```

7. Betrachtung der Sicherheit

Bitte sei Dir im Klaren, dass eine absolute Sicherheit nicht gibt und nicht geben wird.

8. Letzte Ratschläge:

Einige Editoren, z.B. Microsofts Notepad haben die unangenehme Eigenschaft, Dateien „zwangsweise“ mit der Endung „.txt“ zu versehen. Schlimm ist, dass dies in den Standardeinstellungen des Windows-Explorer nicht sichtbar wird. Abhilfe verschafft bei der Eingabe der Dateinamen diesen in einfache Anführungsstriche einzufügen:

Menü: 'Datei' - Menüpunkt 'Speichern unter', Eingabe des Dateinamens mit Anführungsstrichen:

```
'.htaccess'
```

ist hier also zielführend.

Der Autor Jörg Reinholz

arbeitet seit dem Jahr 1999 als freier EDV-Dozent und Webentwickler (häufig mit „Webdesigner“ verwechselt). Seine Themenbereiche sind ganz allgemein Linux, Shell-Programmierung, Perl, PHP, das Web und natürlich diverse Dienste wie DNS, FTP, HTTP(Apache), SSH, Mail, MySQL. Tätig war er bereits im Auftrag namhafter Unternehmen unter anderem für die Deutsche Telekom AG, die Post AG, K&S AG, Daimler-Chrysler, die Bundeswehr und sogar das österreichische Bundesheer. Er programmierte unter anderem einen Webshop der für über 200 mittelständische Unternehmen im Einsatz ist.

Sie erreichen Herrn Reinholz über seine Webseite: http://www.fastix.de/kontakt.php

Originale URL dieses Dokumentes:

http://www.fastix.de/r/Sichere_Ablage_von_Zugangsdaten_zu_Datenbanken_in_PHP-Projekten.pdf

©Jörg Reinholz, fastix WebDesign & Consult, Kassel, http://www.fastix.de
Dieser Aufsatz darf ganz oder teilweise im Rahmen des „fair-use“ in jeder Form (gedruckt, digital, eingebunden in andere Werke) weitergegeben werden, so lange hierbei der Autor wie oben angegeben genannt wird.