Albrecht Weinert

Memo

Apache on Windows — mySQL user authentication

http://a-weinert.de/weinert/pub/apache_auth_mysql.pdf

Stand: 06.08.2010

MEVA Lab
Medien und verteilte Anwendungen

Albrecht Weinert
Labor für Medien und verteilte Anwendungen (MEVA-Lab)
Fachbereich Informatik der Hochschule Bochum

Apache on Windows — mySQL user authentication

V01.01, 20.07.2010: neu
V01.03, 23.07.2010: erste abgeschlossene Version (ohne Kap.3)
V01.04, 06.08.2010: erste abgeschlossene Version

Version: V1.04
Zuletzt geändert von A. Weinert am 06.08.2010

Hinweis: Listen, Tabellen, Listings, Bilder etc. sind gemeinsam durchnummeriert.

Hinweis: Die URL dieses Dokuments ist
http://www.a-weinert.de/weinert/pub/apache_auth_mysql.pdf .
Die dort zu findende Version könnte neuer sein, als das Vorliegende.

Inhalt

1. Motiv, Hintergrund 3
1.1 Vor- und Nachteile der textbasierten Nutzerverwaltung 3
1.2 Alternativen 4
2. Vorgehen 4
2.1 Modul-Installation und -Test 4
2.2 Vorbereiten der Datenbank 5
2.3 Aktivieren der Authentifizierung gegen mySQL 7
3. Nutzerhandhabung 8
3.1 Datenbankhandhabung mySQL 8
2.3 Zur Handhabung mit PHP 8
4. Resümee 10
Anhang 11
A1 Konfigurationshinweise (Zitat, Englisch) 11
Configuration Parameters.....11
Additional Information.....14
Multiple Tables.....14
Formats 14
A2 Abkürzungen 16
A3 Literatur 19

1. Motiv, Hintergrund

Ein Apache-Web-Server dient als Grundlage für eine ganze Reihe von Web-Diensten und liefert auch geschützte Inhalte. Im hier berichteten Beispiel läuft er auf einem gemieteten (virtual) Server mit Intel-Architektur unter Windows Server 2003 enterprise, 64 bit. Natürlich lässt sich das Verfahren allgemein anwenden.

Der Apache-Web-Server macht für seine Diente selbst und einheitlich die Nutzer- und Gruppenverwaltung. Eine solche stützt sich standardmäßig auf zwei Textdateien, deren Aufbau die Ausschnitte Listing 1 und 2 zeigen. Diese zwei Dateien findet man (bei einer Collabnet-Apache-Installation) üblicherweise unter

- `C:\programme\subversion\httpd\conf\auth_file`
- `C:\programme\subversion\httpd\conf\auth_group`

Es kann aber auch mehr als zwei solche Dateien geben, falls man "virtual hosts", "locations" und "directories" diesbezüglich (unter Verlust der Einheitlichkeit der Nutzerbasis und des Logins) ganz unterschiedlich handhaben will.

```
albrecht:$apr1$LpD/6bhH$cafegWl4K7v/p65ScNU0I/
ralf:$apr1$eB1.....$RHmxhRcafe8Fu0lr2RHTk1
christoph:$apr1$C9MvRrMg$j7RZcafeyMVwh7.W1bcpb/
rolf:$apr1$xZG4bl9j$zXYnZQPAgOUcafee3kfl0
thorsten:$apr1$1zEQm89x$ExhFdGCRR.zcafeDD.q0V.
.......
```

Listing 1: Die Nutzerdatei in Apache-Textformat (Beispielauszug).

```
Apache authorisation
#  Server ai2t.de
#  nur für Apache-spezifische Gruppen
#  V.$Revision: 178 $ ($Date: 2010-07-15 12:17:19 +0200 (Do, 15 Jul 2010)$,
       $Author: albrecht $)

#  AI²T-Gruppen (teils nur Spiegel von EGroupware-Gruppem)
ai2t_member:  albrecht rolf ralf christop....
ai2t_admin:   albrecht egw_ad......
ai2t_web_adm: albrecht ralf egw_ad....
ai2t_svn_adm: albrecht ralf egw_ad....
ai2t_partner: andreas joergF

public: guest gast .......
```

Listing 2: Die Gruppendatei in Apache-Textformat (Beispielauszug).

1.1 Vor- und Nachteile der textbasierten Nutzerverwaltung

In Listing 1 und 2 vorgestellte Textdateien lassen sich mit jedem Editor und auch in einem (Serververwaltungs-) Eclipse-Projekt gut bearbeiten. Das gehashte Passwort lässt sich mit `htpasswd.exe` (zu finden in C:\programme\subversion\httpd\bin\) erzeugen.

Die Texte sind selbstdokumentierend und sie lassen sich (mit z.B. Subversion) unter Versionsverwaltung stellen. Im letzteren Ansatz kann man sogar nach einem commit auf dem Server beliebig komplizierte Aktionen bis hin zu einem abschließenden restart von Apache automatisiert anstoßen (Zauberwort: post commit hook).

Andererseits führt der text-basierte Ansatz nicht sehr weit (bzw. in beliebige zusätzliche Mühen), falls man die Nutzer- und Gruppenverwaltung mit Scripting und/oder Webdiensten automatisieren will. Man denke nur an das notorische "Passwort vergessen".

Zudem werden dem text--basierten Ansatz (von einer gewissen Größe der Nutzerbasis an sicher zurecht) Performance-Probleme nachgesagt.

1.2 Alternativen

Sowohl bei der Automatisierbarkeit als auch bei der Performance kommt man mit Datenbank- (DB) basierten Ansätzen sicher weiter.

Apache bietet standardmäßig ein DB-Autorisierungs-Modul (mod_authn_db...so). Dieses arbeitet aber mit einem integrierten DB-Programm auf dann zwei privaten Binär-Dateien. Mit denen kann man dann noch weniger anfangen als mit den o.a. Textdateien bzw. man ist dazu auf einen mehr oder weniger "glücklichen" Werkzeugsatz (teilweise nur in Script-Sprachen vorliegend) angewiesen. mod_..._db... ist also eher eine Sackgasse, und die Frage "Wenn schon Datenbank, warum dann keine übliche?" ist mehr als berechtigt.

Wer mySQL einsetzen kann oder dies eh (für einige der gehosteten oben erwähnten Web-Anwendungen) schon tut, findet mit dem Modul `mod_auth_mysql` möglicherweise die Lösung.

Anmerkung: Allerdings ist das hierzu "Ergoogelte" im Zusammenhang mit Windows (gar 64 Bit), so voller Probleme, dass man an den Erfolg zunächst nicht glaubte. Auch mag man eine professionelle und funktionierende Installation nicht mit scheinbar fragwürdigen Ansätzen gefährden. Neben unzähligen Problemberichten gab es glücklicherweise auch ein paar fundierte Hinweise. Dank diesen und dem im folgenden festgehaltenen systematischen Vorgehen klappte es doch ziemlich schnell und absolut problemlos.

2. Vorgehen

Die im hier zugrunde liegenden Beispielfall verwendete Apache-Distribution von Collabnet — "gebundlet" mit einem Subversion-Server — ist wohl eine der stabilsten und professionellsten für Windows.

2.1 Modul-Installation und -Test

Sie bringt aber, wie erwähnt, das mySQL-Modul nicht mit. Dies findet man in einer anderen Windows-Distribution funktionierend vor, nämlich in XAMPP, das man sich nur hierfür runter lädt:

```
21.07.2010  09:31     85.319.590   xampp-win32-1.6.2.zip
```

Hieraus extrahiert man lediglich die Datei

```
20.01.2006  07:33      1.511.513   mod_auth_mysql.so
```

und tut diese in das Verzeichnis

```
C:\programme\subversion\httpd\modules
```

In der Konfigurationsdatei `httpd.conf` fügt man an einschlägiger Stelle

```
LoadModule mysql_auth_module modules/mod_auth_mysql.so
```

hinzu.

Dies ist im ersten Schritt Alles. Als Test muss man nun ein re-start von Apache machen. Wenn dieser genau problemlos wie vorher läuft, hat man schon gewonnen. Das neu hinzugekommene Modul passt zum Apache und auch zum 64-Bit-Windows, auch wenn der o.a. Name `xampp-win32-1.6.2.zip` das Gegenteil vermuten ließ.

Anmerkung: Es mag einem widerstreben 85MB .zip zu laden, um daraus eine einzige 1,5MB große Datei zu verwenden. Aber was soll's. Diese Distribution ist für Windows gemacht — und das offenbar gut.

2.2 Vorbereiten der Datenbank

Nun das Modul erfolgreich installiert ist, muss man zunächst

- eine passende Datenbank bereitstellen und
- alle existierenden Nutzer und Gruppen dort eintragen.

Für die Datenbankstruktur soll gelten:

- Sie soll für künftige Aufgaben notwendige Informationen mit enthalten.
- Nutzer sollen Mitglied in mehr als einer Gruppe sein können.

Der letzte Punkt führt zu zwei Tabellen, einer für die (erweiterte) Nutzer-Information plus eine nur für die Gruppenzuordnung. (Nur falls man sich die Möglichkeit mehrerer Gruppen verbauen mag, genügte eine Tabelle.)

Die Listings 3 und 4 zeigen den Aufbau der beiden Datenbanktabellen für die Daten der Nutzer bzw. ihre Gruppenzugehörigkeiten.

```
delimiter $$

CREATE TABLE `user_info` (
  `user_name` char(64) NOT NULL,
  `user_passwd` char(64) DEFAULT 'md5',
  `passwd_plain` char(64) DEFAULT NULL,
  `u_mail` char(64) DEFAULT NULL,
  `may_req_pwc` int(11) DEFAULT '1',
  `given_name` varchar(45) DEFAULT '',
  `surname` varchar(45) DEFAULT NULL,
 PRIMARY KEY (`user_name`),
  UNIQUE KEY `user_name_UNIQUE` (`user_name`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
  COMMENT='ai2t.de  Apache mySQL Authent., Nutzer'$$
```

Listing 3: Erzeugung der Nutzertabelle für MySQL-Auth.

```
delimiter $$

CREATE TABLE `user_group` (
  `user_name` char(64) NOT NULL DEFAULT '',
  `user_group` char(24) NOT NULL DEFAULT '',
  PRIMARY KEY (`user_name`,`user_group`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
  COMMENT='ai2t.de  Apache mySQL Authent., Gruppen'$$
```

Listing 4: Erzeugung der Gruppentabelle für MySQL-Auth..

Für die reine Authentifizierungsfunktion genügen die Spalten `user_name` und `user_passwd`, die die Kontenbezeichnung und den (MD5-) hash des Passworts halten. Bis auf den Unterschied MD5 / SHA1 steht in der vorherigen Textdatei nichts anderes drin. Beide Einträge dürfen nicht leer sein und die Kontenbezeichnung muss eh eindeutig sein.

`passwd_plain` ist für das Klartextpasswort. Dieser Eintrag sollte i.A. leer sein. Er dient nur administrativen Anwendungen und ermöglicht einen experimentellen Rückfall auf unverschlüsselt.

`may_rec_pwc` (may request password change) ist bei den meisten Konten (default) true (1) und signalisiert einer entsprechenden Anwendung, dass diesem Nutzer ein Passwortänderung bzw. -erneuerung gestattet ist. Wird nur bei "Verdächtigen" und bei von mehreren Personen zu nutzenden Logins auf false (0) gesetzt.

Die übrigen Spalten enthalten Namen, Vornamen und eine mail-Adresse, also Informationen, die man schon in der ursprünglichen Textdatei gerne dabei gehabt hätte.

Die Gruppentabelle ist selbsterklärend; jede Zeile stellt genau eine Gruppenmitgliedschaft dar. In ihr muss die Kombination aus `user_name` und `user_group` eindeutig sein. Weitere (hier noch nicht vorgesehene) Spalten dieser Tabelle könnten die Mitgliedschaft des betreffenden Nutzers in der betreffenden Gruppe begründen (Warum ist Meier in admin?) oder auch zeitlich einschränken. (Das Modul ermöglicht so etwas auf SQL-Basis.)

```
INSERT INTO `apa_auth_db`.`user_info`
(`user_name`, `user_passwd`, `passwd_plain`,
  `u_mail`, `may_req_pwc`, `given_name`, `surname`)
VALUES
(
 {user_name: %newUser%}, {user_passwd: CHAR}, {passwd_plain: CHAR},
 {u_mail: CHAR}, {may_req_pwc: 0},
 {given_name: VARCHAR}, {surname: VARCHAR}
);


INSERT INTO `apa_auth_db`.`user_group`
(`user_name`, `user_group`)
VALUES
({user_name: %newUser%}, {user_group: %group1%}), ......
({user_name: %newUser%}, {user_group: %groupn%});
```

Listing 5: Einsetzung eines Nutzers für MySQL-Auth. (%x% und CHAR sind Platzhalter)

Nun muss man diese Datenbank "nur" noch mit allen bzw. mit den für erste Tests wesentlichen Nutzer füllen. Listing 5 fasst die Anweisungen zusammen, die einen Nutzer mit mehreren Gruppenzugehörigkeiten neu eintragen.

Hinweis: Die ursprünglichen Textdateien kann man nebenläufig stehen lassen und durch entsprechende Konfiguration bereichsweise parallel verwenden. Wenn erste Tests das Funktionieren von Authentifizierung und gegen Gruppen und Nutzer zeigen, sollte man aber alsbald alles portieren und umstellen.

Hinweis 2: Zur Herstellung des MD5-Digest des Passworts gibt es einige Werkzeuge und Bibliotheksfunktionen auch in PHP und Java. Mit dem Java-Framework Frame4J hat man das passend Werkzeug. Der Aufrufe

```
java de.frame4j.MakeDigest password
java de.frame4j.MakeDigest password -append toFile
```

. liefern beide den MD5-Digest zu password. Die zweite Variante hängt nur diesen (schweigend) an die bezeichnete Datei an, sodass sich diese Information so gut in Scripten (batch) verwenden lässt.

2.3 Aktivieren der Authentifizierung gegen mySQL

Alle Bereiche, also "virtual hosts", "locations" und "directories" haben in der betreffenden Konfigurationsdatei — das ist i.A.

```
C:\programme\subversion\httpd\conf\extra\httpd-ssl.conf
```

— für die vorherige Textdatei-basierte Authentifizierung Einträge der etwa in Listing 6 gezeigten Form.

```
<Directory "c:\www\ai2t-de\egroupware" >
  Require group ai2t_web_adm egroupware bachi fam_we
  AuthType Basic
  AuthName "Doku.. Wiki Subversion SVN .."

  AuthUserFile "c:\programme\subversion/httpd/conf/auth_file"
  AuthGroupFile "c:\programme\subversion/httpd/conf/auth_group"

  AllowOverride FileInfo Options
  ErrorDocument 404 /error404.html
</Directory>
```

Listing 6: Konfigurationseintrag für Textdatei-Authentifizierung (Bereichsbeispiel)

Anmerkung: `AuthType Basic` sollte nur verwendet werden, wenn für alle so geschützten Bereiche https erzwungen wird. Dann ist und bleibt `AuthType Basic` aber auch vollkommen OK.

Nach dem oben beschriebenen erfolgreichen Vorbereiten des Moduls und der Datenbank muss man an den Listing 6 entsprechenden Stellen nun lediglich die beiden mit `AuthUserFile` und `AuthGroupFile` beginnenden Zeilen durch Listing 7 ersetzen.

```
#only4fallbck2txt#  AuthUserFile  "c:\programme\subversion/httpd/conf/auth_file"
#only4fallbck2txt#  AuthGroupFile  "c:\programme\subversion/httpd/conf/auth_group"

AuthMySQLHost localhost
AuthMySQLDB apa_auth_db
AuthMySQLUserTable  user_info
AuthMySQLGroupTable user_group
AuthMySQLEnable On
AuthMySQLUser     apaAutUser
AuthMySQLPassword leDBTmotDp
AuthMySQLAuthoritative On

AuthMySQLNameField user_name
#only4Test#  AuthMySQLPasswordField passwd_plain
#only4Test#  AuthMySQLPwEncryption none
AuthMySQLPasswordField user_passwd
AuthMySQLPwEncryption  md5
AuthMySQLGroupField user_group
```

Listing 7: Konfigurationseintrag für Authentifizierung mit mySQL (Auszug)

Restart — und das war's.

3. Nutzerhandhabung

3.1 Datenbankhandhabung mySQL

Die Nutzer, ihre Eigenschaften und Gruppenzugehörigkeiten lassen sich nun durch Zugriff auf die beiden genannten mySQL-DB-Tabellen betrachten und verwalten. Für die Wandlung eines Klartextpassworts in MD5 nimmt man `de.frame4j.MakeDigest` aus dem Framework Frame4J.

Leider lassen sich die Zugriffe auf die DB selbst nicht wirklich leicht in robuste Kommandos, die in Scripten (batch files) automatisiert einsetzbar wären fassen. Man hat etwas den Eindruck, dass Windows command line tools von den mySQL-Entwicklern eher "auf Lücke" gesetzt sind.

Insofern ist eine gute graphische Oberfläche für mySQL schon zwingend. Unter Windows ist zur Zeit des Schreibens (nur) mySQLworkbench das Mittel der Wahl. Das ausgezeichnete Werkzeug erfordert allerdings leider dotNet. Das findet man gemieteten (virtual) servern oft nicht oder nicht in der neuesten Vestion vor, und schließlich kann man auf dotNet in einem vorwiegend mit Java und open source gefahrenen Windows Server ja auch meist völlig verzichten.

In den sauren Apfel muss man für mySQLworkbench beißen, aber der Komfort und die gewonnene Übersicht ist es allemal wert.

2.3 Zur Handhabung mit PHP

Im Gegensatz zur textdateibasierten Authentifizierung lässt sich die mySQL-basierte gut mit PHP nutzen. So lasssen sich viele Vorgänge, wie unter anderem das notorische "Passwort vergessen", mit Webdiensten automatisieren.

```
<?php // get user acount & declare variables
  $debug = false; // set only true in protected development environment
  $user_account=$_SERVER['REMOTE_USER']; // the user logged in (at browser)
  $username = "apCafeUser";   // user for DB $database
  $password = "meCafePass";  // password for $database
  $database = "apa_auth_db";   // mod_auth_mysql
  $usertable= "user_info";    // user table
  $grouptable= "user_group"; // group table
  $dbError  = "";       // = false = no er5ror (yet)
  $mayReqPwc = false;  // field in user table may request password etc. change
  $dbSelected = false;      // no DB selected (yet)
  $numRowsUser = 0; // no user row found yet,  0 or 1 as accounts are unique
  $surnUser = "";   // field from DB: name
  $givnUser = "";  // field from DB: christian name
  $mailUser = ""; // field from DB: main personal mail account
  $userPasswd = "";   // its the MD5 from DB
  $numRowsGroup = 0; // number of group rows; >= 0 as any number of groups
  // $resultGroups will contain the group memberships if any
?>
```

Listing 8: PHP-Variablen zur o.a. Konfiguration der Authentifizierung mit mySQL

Eine PHP-Seite von der hier die Rede ist, sollte nur mit https und nur für einen authentifizierten Anwender erreichbar sein. Listing 8 zeigt eine für solche Seiten schon fast standardmäßig verwendbare Variablenvereinbarung für nachfolgende DB-Zugriffe auf die Nutzerdaten.

Anmerkung: Ja, in PHP braucht man so was gar nicht, aber man kann auch bei chaotischen Programmiersprachen versuchen, gewisse Standards und Stile durchzuhalten.

Mit Listing 9 an gleicher oder (nach Prüfen anderer Bedingungen) späterer Stelle kann man nun die Nutzerdaten zur weiteren Verwertung aus der Datenbank lesen. Es zeigt auch beispielhaft den Zugriff auf einzelne DB-Felder mit den Funktionen `mysql_num_rows()`, welche die Anzahl der zutreffend gefundenen Zeilen oder 0 liefert. und `mysql_result()`. Letztere handhabt das Ergebnis einfach als Zeile*Spalte-Matrix.

Nach diesem Schema kann man sinngemäß "alles" machen. Änderungen in der DB gehen mit entsprechenden SQL-Queries (UPDATE statt SELECT z.B.), wenn der in . Listing 8 vereinbarte Nutzer die Rechte hat, auch.

```
<?php // read mySQL data (should be as user is authenticated)
 $dbLink = mysql_connect('localhost',$username,$password);
 if ($dbLink) {
  $dbSelected= mysql_select_db($database);
 }
 if ($dbSelected) {
  //Index           0        1        2             3          4        5 (MD5)
  $query  = "SELECT user_name, u_mail, may_req_pwc, given_name, surname,
        user_passwd FROM ".$usertable." WHERE user_name = '".$user_account."'";
  $result = mysql_query($query);
 }
 if (! $result) {
  echo("Server error: unable to select database<br />");
  if ($debug) {
    echo('query :  '.$query.'<br />' );
    echo("error :  ".mysql_error().'<br />' );
  }
 } else { // there is a result
  $numRowsUser  = mysql_num_rows($result);
  if ($numRowsUser == 1) {
    $mayReqPwc  = mysql_result($result,0,2);
    $mailUser   = mysql_result($result,0,1);
    $surnUser   = mysql_result($result,0,4);
    $userPasswd = mysql_result($result,0,5);
    $givnUser   = mysql_result($result,0,3);
    // get group memberships
    $query  = "SELECT user_name, user_group FROM ".$grouptable."
                   WHERE user_name = '".$user_account."' ORDER BY user_group";
    $resultGroups = mysql_query($query);
    if ($resultGroups) {     // meberships in mysql_result($resultGroups,$i,1)
      $numRowsGroup = mysql_num_rows($resultGroups);  // may still be 0 ! (OK)
    } else  if ($debug) {  // failed to get group membership
      echo('query :  '.$query.'<br />' );
      echo("error :  ".mysql_error().'<br />' );;
    }
 }// there is a user result
}?>
```

Listing 9: Lesen der Nutzerdaten aus der mySQL-DB

4. Resümee

Dieses Memo hält fest, wie man die Authentifizierung eines (gut) funktionierenden Apache-Webservers von textfile-basierter Authentifizierung auf mySQL-basiert umstellt.

So wie geschildert geht dies erstaunlich problemlos und bis auf zwei restarts ohne Unterbrechung der Dienste.

Was anschließend recht gut gelingt ist die Teilautomatisrierung der Nutzerverwaltung mit PHP-Selbstbedienungsfunktionen, wie Datenabfrage, gewisse Änderung, Passwort vergessen (mit Ticket-mail) etc.

Anhang

A1 Konfigurationshinweise (Zitat, Englisch)

Das Folgende ist der Vollständigkeit halber ein umfangreicher Auszug aus dem Text "Configure" von mod_auth_mysql.

Authentication options may be placed in the appropriate <Directory> entry in your httpd.conf file, or in a .htaccess file in the directory you wish to protect. Using the httpd.conf file is preferred, but you must restart Apache after making any changes. If you change the .htaccess file, you do not need to restart Apache, but there is additional run-time overhead. (Apache must read every .htaccess file from the file's directory back up to the DocumentRoot on EVERY request.)

Configuration Parameters

Configuration paramaters generally contain a single value:

On | Off : Whether this value is active or not

Number: A valid integer value for the paramater

String: A string containing the value. The string may include spaces if the entire string

Following are the options for configuring mod_auth_mysql.

```
AuthMySQLEnable On
AuthMySQLHost localhost
AuthMySQLPort <default port in MySQL>
AuthMySQLSocket <default socket in MySQL>
AuthMySQLUser <no default -- NULL>
AuthMySQLPassword <no default -- NULL>
AuthMySQLDB test
AuthMySQLUserTable user_info
AuthMySQLUserCondition <no default>
AuthMySQLNameField user_name
AuthMySQLPasswordField user_passwd
AuthMySQLNoPasswd Off
AuthMySQLPwEncryption crypt
AuthMySQLSaltField <>
AuthMySQLGroupTable <defaults to value of AuthMySQLUserTable>
AuthMySQLGroupCondition <no default>
AuthMySQLGroupField <no default>
AuthMySQLKeepAlive Off
AuthMySQLAuthoritative On
AuthMySQLCharacterSet <no default>
```

AuthMySQLEnable On | Off

Whether or not mod_auth_mysql should attempt to authorize the user.
Off: No authorization will be done by this module
On: Attempt to authorize the user

AuthMySQLHost localhost | host_name_or_ip_address

Identifies the MySQL host.

AuthMySQLPort tcp/ip_port_number

The tcp/ip port which should be used to access MySQL. MySQL normally uses port 3306, but this can be changed in the MySQL configuration. See the MySQL documentation for more details.

AuthMySQLSocket full_path_to_socket_file

The UNIX socket which should be used to access MySQL host "localhost" on a UNIX system. The default is /tmp/mysql.sock, but this can be changed in the MySQL configuration. See the mySQL documentation for more details.

AuthMySQLUser userid

The userid to be used to access MySQL. This user must have SELECT access to the appropriate tables. As the password must be in plain text (see AuthMySQLPassword below), it is recommended you use a userid with limited privileges (do NOT use "root"!).

AuthMySQLPassword password

The password for the userid specified in AuthMySQLUser. An, as the password must be in plain text, it is recommended you use a userid with limited privileges (do NOT use "root"!).

AuthMySQLDB database_name

The name of the MySQL database containing the authorization information. On systems with case sensitive file systems (i.e. Unix), this field is case sensitive.

AuthMySQLUserTable mysql_table_name

The name of the MySQL table in AuthMySQLDB which contains the userids and passwords. (If this field contains two or more table names, you will need to join the tables in the AuthMySQLUserCondition; see below).

AuthMySQLUserCondition

Additional conditions to be placed in the WHERE clause when retrieving user information. Whatever is in this string is appended after an AND condition in the SQL statement.
If two or more tables have been specified in the AuthMySQLUserTable option above, this option must contain the information required to join the tables.

AuthMySQLNameField mysql_column_name

The name of the column in AuthMySQLUserTable which contains the userids to be authenticated. The column must contain unique, non-empty field values. Its length is however long you want it to be. This value is case sensitive.

AuthMySQLPasswordField mysql_column_name

The name of the column in AuthMySQLUserTable which contains the passwords. This value is case sensitive. It's length may be as long as you want it to be for plaintext passwords. If the password is encrypted, the field must be long enough to contain the encrypted data. Passwords values are case sensitive.

AuthMySQLNoPasswd Off

No password is required for this resource.

AuthMySQLPwEncryption none | crypt | scrambled | md5 | aes | sha1

The encryption type used for the passwords in AuthMySQLPasswordField:
none: not encrypted (plain text)
crypt: UNIX crypt() encryption
scrambled: MySQL PASSWORD encryption
md5: MD5 hashing (recomended)
aes: Advanced Encryption Standard (AES) encryption
sha1: Secure Hash Algorihm (SHA1)
WARNING: When using aes encryption, the password field MUST be a BLOB type.

AuthMySQLSaltField <> | <string> | mysql_column_name

Contains information on the salt field to be used for crypt and aes encryption methods. It can contain one of the following:
<>: password itself is the salt field (use with crypt() only)
<string>: "string" as the salt field
mysql_column_name: the salt is take from the mysql_column_name field in the same row as the password
This field is required for aes encryption, optional for crypt encryption. It is ignored for all other encryption types.

AuthMySQLGroupTable

Contains the name of the table with the group information when authorizing by groups (Apache option require group). As with the AuthMySQLUserTable, you can specify two or more tables in this option, in which case you will need to join the tables in the AuthMySQLGroupCondition below.

AuthMySQLGroupCondition

Additional conditions to be placed in the WHERE clause when retrieving group information. Whatever is in this string is appended after an AND condition in the SQL statement. If two or more tables have been specified in the AuthMySQLGroupTable option above, this option must contain the information required to join the tables.

AuthMySQLGroupField

This option contains the name of the column containing the group information when Apache group authorization is required. Values in the Apache require group option will be matched against the retrieved rows.

AuthMySQLKeepAlive

Indicates whether to keep the connection to MySQL open or close it after each request. Keeping the connection open can improve performance at the cost of the resources necessary to maintain the connection. If this is Off, the connection will be closed after each request.
Currently, only one connection to the server can have AuthMySQLKeepAlive on.
Note: This parameter currently does not work with Apache 2.x and is ignored.

AuthMySQLAuthoritative

Used to indicate if other modules should be called when mod_auth_mysql is not able to authorize the user. If this is On, no other modules will be called and the request will fail. If this is off, Apache will attempt to use mod_auth and/or any other active modules to authorize the user.

AuthMySQLCharacterSet

Used to override the default characterset for the connection. This parameter must specify a valid character set in MySQL. It is generally required only in MySQL 4.1 and above, where the characterset encoding for the tables being used is different that the default specified in the MySQL configuration.

Additional Information

AuthMySQLUserCondition and AuthMySQLGroupCondition

The optional directives AuthMySQLUserCondition and AuthMySQLGroupCondition can be used to restrict queries made against the User and Group tables.The value for each of these should be a string that you want added to the end of the where-clause when querying each table. For example, if your user table has an "active" field and you only want users to be able to login if that field is 1, you could use a directive like this:
AuthMySQLUserCondition active=1
You can specify parameters for system options. These parameters will be replaced by the appropriate values in the query. See Formats below.

Multiple Tables

If you have user information stored in two (or more) different tables, you can join the tables like this:

```
AuthName My Authorization

AuthType Basic

AuthGroupFile /dev/null  # do NOT include this directive if using Apache2!!!

AuthMySQLHost localhost

AuthMySQLDB test

AuthMySQLUserTable "user_info, user_status"

AuthMySQLUserCondition = "user_info.user_name = user_status.user_name and
        user_status.status = 'OK'"

require valid-user
```

Formats

You can specify the following paramaters in the AuthMySQLUserCondition and AuthMySQLGroupCondition clauses. They will be replaced by the appropriate values in the query:

%h	DNS name of the remote host
%a	IP address of the remote host
%f	The filename being requested
%V	Hostname of the Apache server
%v	Virtual hostname
%H	Protocol sent with the request (i.e. HTTP/0.9)
%m	Request method (i.e. GET, HEAD, POST, etc.)
%q	Arguments following the ? in the request
%r	Request line
%U	Path portion of the URI

These parameters can be used to further limit access. For instance, if you wish to limit users to a single ip address, you could add the following column to your user_info table:

```
ip_address VARCHAR (15)
```

You could then do something similar to this:

```
AuthName My Authorization
AuthType Basic
AuthMySQLHost localhost
AuthMySQLDB test
AuthMySQLUserTable user_info
AuthMySQLUserCondition = "ip_address = '%a'"
require valid-user
```

[Ende des Zutats aus "configure"]

A2 Abkürzungen

ACL	access control list (Liste mit Zugriffsrechten auf ein Objekt)
AD	Active Directory (Microsofts Interpretation von LDAP)
AJAX	Asynchronous JavaScript + XML
API	Application Programme Interface
BuB	Bedienen und Beobachten (von Prozessen)
C/S	Client-Server
CA	Certification authority
CVS	Concurrent Versioning System
DB	Datenbank
FAQ	Frequently Asked Questions (Hilfetexte in Frage-Antwort-Form)
FSFS	fast secure file system (neues Datenbanksystem von SVN)
GSS	Generic Security Service
GUID	Globally Unique Identifier
GWT	Google Webtoolkit, AJAX mit nur Java
HTML	Hypertext Markup Language [RFC 1866]
HTTP	Hypertext Transfer Protokoll. Internet-Protokoll zur Übertragung von Seiten.
HTTPS	HTTP über SSL. Abgesicherte Übertragung.
HW	Hardware
IIOP	Internet Inter-ORB Protocol
IP	Internet Protocol
J2EE	Java 2 Enterprise Edition
J2ME	Java 2 Micro Edition
J2SE	Java 2 Standard Edition
JAAS	Java Authentication and Authorization Service
JAF	JavaBeans Activation Framework
JAR	Java Archive. (.zip + Semantik)
JAXP	Java API for XML Parsing
JCA	Java Cryptography Architecture (der Java Security API)

JCE	Java Cryptography Extensions (zur JCA, Exportrestriktion wegen DAS, DES)
JDBC	Java Database Connectivity (Java Datenbankanschluss)
JDC	Java Developer Connection (Ein WWW-Service)
JDK	Java Development Kit; der Werkzeugsatz für die Entwicklung mit Java
JEB	Enterprise JavaBeans (ungleich JavaBeans)
JMX	Java Management Extensions
JNDI	Java Naming and Directory services Interface
JNI	Java Native Interface
JRE	Java Runtime Environment; JDK-Subset ohne Entwicklungswerkzeuge.
JSF	Java Server Faces
JSP	Java Server Pages
JSSE	Java Secure Socket Extension (seit JDK1.4.x integriert)
JSTL	JavaServer Pages Standard Tag Library
JVM	Java virtual machine; der eigens für Java erfundene Prozessor. Er wird im Allgemeinen auf dem jeweiligen Zielsystem emuliert.
LAN	Local area network; Datennetz für mittlere Entfernungen
LDAP	Lightweight Directory Access Protocol
LGPL	Lesser GNU Public License
MBean	Managed Bean (JMX)
MEVA	Labor für Medien und verteilte Anwendungen
MS	Microsoft
NT	Betriebssystem Windows NT (MS)
OMG	Object Management Group
OS	Operating System
PAM	Pluggable Authentication Module
PC	Personal Computer
R&D	Research and Development
RAID	Redundant Array of inexpensive Disks
RDF	Resource Description Framework (W3C)
RMI	Remote Method Invocation
RPC	Remote Procedure Call

SMTP Simple Mail Transfer Protocol

SOAP Simple Object Access Protocol

SQL Structured query language, Datenbankbearbeitungssprache

SSL Secure Socket Layer. Protokollschicht zu Absicherung.

SSO Single Sign on; Authentifizierung vieler (n) Anwendungen gegen eine (1) "security realm".

SSPI Security Support Provider Interface

SVN Subversion

TCP Transmission Control Protocol

TM Trade Mark (Warenzeichen)

UML Unified Modelling Language

URI Uniform Resource Locator

W2K Betriebssystem Windows 2000 (MS)

W2K3 Betriebssystem Windows Server 2003 (MS)

W3 Amerikanische Kurzform für WWW

W3C World Wide Web Consortium

WebDAV Web-based Distributed Authoring and Versioning

WS Workstation

WSDL Web Services Description Language

XML eXtensible Markup Language

A3 Literatur

[1] Ed Ort and Mark Basler, AJAX Design Strategies, SUN 2006
http://java.sun.com/developer/technicalArticles/J2EE/AJAX/.../design-strategies.pdf

[2] Brett McLaughlin, Mastering Ajax, Part 1..4, IBM, 2005
http://www-128.ibm.com/developerworks/web/library/wa-ajaxintro.html

[3] Albrecht Weinert, Zur Installation des JDK (Java Development Kit)
http://a-weinert.de/weinert/pub/java-install.txt

[4] Albrecht Weinert, Java — Tipps und Tricks
http://a-weinert.de/weinert/pub/java-tips.txt

[5] Albrecht Weinert, AJAX mit GWT — Tipps und Tricks
http://a-weinert.de/weinert/pub/gwt-tips.pdf

[6] Albrecht Weinert, Tipps zu CVS für Windows — cvsNT
http://a-weinert.de/weinert/pub/cvsnt-tipp.txt

[7] Google, Web-Toolkit, online-Dokumentation (nicht am Stück verfügbar)
http://code.google.com/webtoolkit/documentation/.

[8] Albrecht Weinert, Tipps zu JMX mit SSL
http://a-weinert.de/weinert/pub/jmx-ssl-tips.pdf

[9] Albrecht Weinert, Windows 2003 Domain
Migration von NT4 mit Fremd-DNS
http://www.a-weinert.de/weinert/pub/w2k3domain.pdf

[10] Albrecht Weinert, Windows Server 2003 — Domain FB3-MEVA
Schulungsräume und Infrastruktur — Renovierung 2007
http://www.a-weinert.de/weinert/pub/fb3-meva-domain2007.pdf

[11] Albrecht Weinert, Tipps zu Tomcat (5.x für Windows) ersetzt durch [13]
([13] stattdessen für Tomcat >= 6) http://a-weinert.de/weinert/pub/tomcat-tips.pdf

[12] Albrecht Weinert, Windows Server 2003 — Domain FB3-MEVA
Workstations und Server — Renovierung 2007
http://www.a-weinert.de/weinert/pub/fb3-meva-workst2007.pdf

[13] Albrecht Weinert, Tomcat — mit Windows und Active Directory
(ersetzt [11] als Nachfolger) http://www.a-weinert.de/weinert/pub/tomcat-win-ad.pdf

[14] Albrecht Weinert, Tipps zu MySQL (mit Java, für Windows)
(2006) http://a-weinert.de/weinert/pub/mysqjawi-tipp.txt

[15] Albrecht Weinert, Tipps zu Subversion (2006; ersetzt durch [16])
http://www.a-weinert.de/weinert/pub/subversion-install-tipp.txt

[16] Albrecht Weinert, Subversion — mit Windows und Active Directory (2008)
(ersetzt [15]) http://www.a-weinert.de/weinert/pub/subversion-win-de.pdf

[17] Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato,
Version Control with Subversion; For Subversion 1.5;
als C:\Programme\Subversion\svn-book.pdf bei CollabNet-SVN mit installiert oder im Web

<18> Albrecht Weinert, Apache on Windows — mySQL user authentication
http://www.a-weinert.de/weinert/pub/apache_auth_mysql.pdf .