Auth

ApacheCon

Rich Bowen - <Rich@CooperMcGregor.com>

- mod_auth_mysql
- Access Control
- Allow and Deny
- Satisfy
- Allow/Deny from other criteria
- Writing an AC module with mod_perl
- PerlAccessHandler
- PerlAccessHandler configuration
- PerhAuthenHandler
- PerlAuthenHandler example
- PerlAuthzHandler

Authentication, Authorization, and Access Control

- ApacheCon 2002
- Las Vegas
- Rich Bowen

Index
Forward to About me

About me

- Rich Bowen
- CTO, Cooper McGregor, Inc
- Member, Apache Software Foundation
- Contributor, Apache Server Documentation Project

http://httpd.apache.org/docs-project/

Index
Back to Authentication, Authorization, and Access Control
Forward to Definitions

Auth - ApacheCon

Definitions

- Authentication
- Authorization
- Access Control
- Security

Index
Back to About me
Forward to Authentication

Auth - ApacheCon

Authentication

- Make sure someone is who they say they are
- Usually involves a username and password
- Can involve other "secret"
- Photo ID

Index
Back to Definitions
Forward to Authorization
ApacheCon : Auth - Slide #4 of 33

Authorization

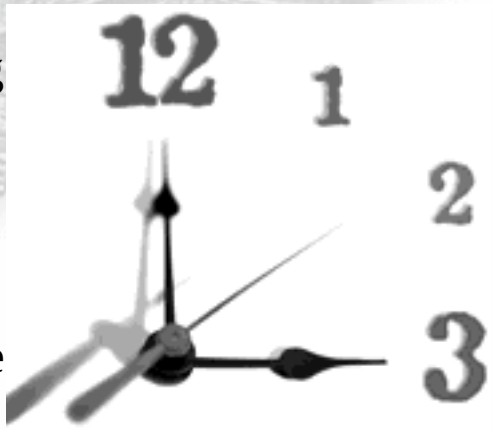
- Once we know who you are ... (usually)
- Are you allowed to be here?
- Often a group of users who fall into a particular category
- Plane ticket

Index
Back to Authentication
Forward to Access Control

Auth - ApacheCon

Access Control

- Other methods of keeping you out
- Usually unrelated to the other two terms
- You can't get on the plane until boarding starts.



Index
Back to Authorization
Forward to Security

Auth - ApacheCon

Security

- Don't confuse auth with security
- Auth can be faked or circumvented.
- People lie.
- Internal attacks are much more deadly than external ones
- An insecure system cannot be helped by a password dialog

Index
Back to Access Control
Forward to Standard authentication types

Standard authentication types

- HTTP spec mentions Basic and Digest auth
- Basic universally supported
- Digest less supported, so less used, so less supported (repeat as necessary)

Index
Back to Security
Forward to Basic

Auth - ApacheCon

Basic

- Server sends 401 Unauthorized header to indicate that authentication is required.
- Credentials (username and password) passed with request
- If credentials are valid (for some definition of valid) then the request is passed on to the next phase of the request handling (usually authorization).

Index
Back to Standard authentication types
Forward to Modules

Auth - ApacheCon

Modules

- mod_auth
- mod_auth_db(m)
- mod_auth_anon

Index
Back to Basic
Forward to Password file

Password file

- htpasswd

```
htpasswd -c password_file username
```

- Prompted for password
- Omit -c to add another user to the same file

Index
Back to Modules
Forward to Configuration

Configuration

- Tell Apache to protect something
- Tell it where your password file is
- Tell it what methods to use

Index
Back to Password file
Forward to Configuration, cont'd

Configuration, cont'd

```
AuthType Basic
AuthName "Go Away"
AuthUserFile /usr/local/apache/passwords/password_file
Require valid-user
```

- "Go Away" appears in the password dialog
- AuthName is also called the "Realm"
- User is asked for username and password.

Index
Back to Configuration
Forward to Configuration, cont'd

Configuration, cont'd

- Can require a particular user ...

```
require user bob
```

- or users.

```
require user billy joe bob
```

Index
Back to Configuration, cont'd
Forward to Groups

Groups

- Create group file

```
admin: tom dick harry
others: larry moe curly
```

- Modify configuration:

```
AuthType Basic
AuthName "Go Away"
AuthUserFile /usr/local/apache/passwords/password_file
AuthGroupFile /usr/local/apache/passwords/group_file
Require group admin
```

Index
Back to Configuration, cont'd
Forward to Caveats

Auth - ApacheCon

Caveats

- Slow - unindexed text files must be read line by line
- Insecure - username and password passed in the clear every time, as is the content.
- Has been known to fail for large user lists (where "large" will depend on many variables.)
- Keep your password file outside of the document root. Preferably owned by and readable only by 'nobody'

Auth - ApacheCon

Digest

- Username/Password never sent to the server
- MD5 hash of name, password, realm
- MD5 is a one-way hashing algorithm
- Password never stored anywhere in a readable format, and never passed across the network at all.

Index
Back to Caveats
Forward to Configuration

Configuration

```
AuthType Digest
AuthName "Go Away"
AuthDigestFile /usr/local/apache/passwords/digest
AuthDigestGroupFile /usr/local/apache/passwords/groups
Require group admin
```

Index
Back to Digest
Forward to Password file

Password file

- htdigest utility to create

```
htdigest [ -c ] passwdfile realm username
```

- And the file looks like:

```
rbowen:foo:b000697b277cfaec3c4476596c52197d
```

- Mathematically impossible to retrieve password from this
- Groups look just like those used by mod_auth

Index
Back to Configuration
Forward to Caveats

Caveats

- Not supported by all browsers
- Content is still passed in the clear
- It is still a text file
- The hash itself could be used as a key

Index
Back to Password file
Forward to Additional auth modules

Additional auth modules

- mod_auth_mysql
- mod_auth_*

http://modules.apache.org/

Index
Back to Caveats
Forward to mod_auth_db

mod_auth_db

- Creating a password file

```
dbmmanage passwords.dat adduser montressor
dbmmanage add groups.dat rbowen one,two,three
```

- dbmmanage --help for full details
- This is still Basic authentication, with all the concerns pertaining thereto. It's just using a different file for its information.

```
AuthName "Members Only"
AuthType Basic
AuthDBUserFile /usr/local/apache/passwd/passwords.dat
AuthDBGroupFile /usr/local/apache/passwd/groups.dat
require group three
```

Index
Back to Additional auth modules
Forward to mod_auth_mysql

mod_auth_mysql

- Username, password, group info, in mysql database tables.

```
create table mysql_auth (
   username char(25),
   passwd char(25),
   groups char(25),
   primary key (username)
   );
```

- Then ...

```
Auth_MYSQL_DB yourdb
Auth_MYSQL_Password_Table mysql_auth
Auth_MySQL_Username_Field username
Auth_MYSQL_Password_Field passwrd
Auth_MYSQL_Empty_Passwords Off
AuthName "Members-Only Area"
AuthType Basic
```

```
require valid-user
```

http://www.cgi101.com/class/password/mod_auth_mysql.html

Access Control

- Allow, or deny, from particular addresses

```
<Location /server-status>
 Order deny,allow
 Deny from all
 Allow from apacheadmin.com
</Location>
```

- These are applied as a series of filters. Everyone is excluded, then apacheadmin.com is let in. The other way around, it would be ineffectual.
- Alternately,

```
Order allow,deny
Allow from all
Deny from something.com
```

Index
Back to mod_auth_mysql
Forward to Allow and Deny

Allow and Deny

- Allow and Deny take a variety of arguments

```
Allow from 192.168.1
Allow from .org
Allow from 192.168.1.0/16
Allow from fully.qualified.hostname.com
```

Index
Back to Access Control
Forward to Satisfy

Satisfy

- Specify more than one rule, and obey one of them

```
Require group customers
Allow from internal.subnet.com
Satisfy any
```

- Or obey all of them

```
Satisfy all
```

Index
Back to Allow and Deny
Forward to Allow/Deny from other criteria

Auth - ApacheCon

Allow/Deny from other criteria

- mod_setenvif and Deny from env=

```
SetEnvIf User-Agent EmailSiphon Spammers
Order Allow,Deny
Allow from all
Deny from env=Spammers
```

Index
Back to Satisfy
Forward to Writing an AC module with mod_perl

Writing an AC module with mod_perl

- PerlAccessHandler for access control
- PerlAuthenHandler for authentication
- PerlAuthzHandler for authorization

Index
Back to Allow/Deny from other criteria
Forward to PerlAccessHandler

PerlAccessHandler

- Provide access control based on arbitrary criteria

```
package Example::AccessHandler;
use Apache::Constants qw(:common FORBIDDEN);
```

```
sub handler {
   my ($class, $r) = @_;
   if (goodconditions($r)) {
       return OK;
   } else {
       return FORBIDDEN;
   }
}
```

Index
Back to Writing an AC module with mod_perl
Forward to PerlAccessHandler configuration

PerlAccessHandler configuration

```
<Location /admin>
   PerlAccessHandler Example::AccessHandler

   PerlSetVar AdditionalArgument1 value1
   PerlSetVar AdditionalArgument2 value2
</Location>
```

- Variables set by `PerlSetVar` are available via the `dir_config` method

```
   $value = $r->dir_config{AdditionalArgument1}
```

Index
Back to PerlAccessHandler
Forward to PerhAuthenHandler

Auth - ApacheCon

PerhAuthenHandler

- Provide authentication via your own custom mechanism
- Eliminate need for pop-up auth dialog
- Store auth information in a cookie
- Retrieve password information from a database

Index
Back to PerlAccessHandler configuration
Forward to PerlAuthenHandler example

PerlAuthenHandler example

- Returns `AUTH_REQUIRED` or `OK`

  http://perl.apache.org/docs/1.0/guide/security.html

- Note that username, password, can also be supplied via other means, like an HTML form, or from a cookie.
- Or, if from a standard auth dialog:

```
my($status, $sent_pw) = $r->get_basic_auth_pw;
my $user = $r->connection->user;
```

- Your auth handler is called each time there is an access
- If getting auth from elsewhere, you should set `$r->`connection->user> to the username value

Index
Back to PerhAuthenHandler
Forward to PerlAuthzHandler

PerlAuthzHandler

- Similarly, returns `OK` or `AUTH_REQUIRED`
- For authorization, not authentication
- Splitting it into these two separate steps improves retention of sanity

Index
Back to PerlAuthenHandler example