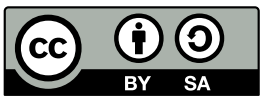
Apache HTTP-Server Teil 2

Zinching Dang

01. Dezember 2014



Benutzer-Authentifizierung

Benutzer-Authentifizierung

- ermöglicht es, den Zugriff auf die Webseite zu schützen
- Authentifizierung mit Benutzer und Passwort gegenüber verschiedenen Systemen:
  - einfache Passwortdatei
  - LDAP
  - etc.
- sollte mit SSL/TLS verwendet werden, um das Mitschneiden der Zugangsdaten zu verhindern
- mit dem Befehl `htpasswd` wird die Passwortdatei verwaltet
  - Passwörter liegen in gehashter Form vor
  - die Option `-c` legt neue Passwortdateien an und überschreibt vorhandene

Der Apache HTTP-Server erlaubt es, Verzeichnisse der Webseite gegen unbefugten Zugriff zu schützen. Die einfachste und hier vorgestellte Möglichkeit ist es, die „Basic Authentication" mit Passwortdateien zu nutzen. Diese fragt Benutzernamen und Passwort ab, welche mit den Datensätzen einer Passwortdatei, die auf dem Server hinterlegt ist, verglichen werden

Da bei der Authentifizierung sensible Daten übertragen werden, ist es ratsam, die Authentifizierung nur in Kombination mit SSL/TLS zu verwenden.

Mit dem Programm `htpasswd` werden diese Passwortdateien verwaltet. Zum Anlegen einer solchen Datei muss die Option `-c` angegeben werden. Zum Hinzufügen weiterer Benutzer oder Ändern von Passwörtern darf diese Option nicht angegeben werden, da die Passwortdatei sonst überschrieben wird. Weitere Optionen finden sich in der Man-Page.

Es ist ratsam, die Passwortdatei nicht in das über den Webserver zugreifbare Verzeichnis zu legen, da sie sonst unter Umständen über den Webserver abrufbar ist.

Benutzer-Authentifizierung

Konfigurationsbeispiel

- Beispiel-Konfiguration:

```
<IfModule mod_ssl.c>
<VirtualHost *:443>
  ServerName www.example.com
  DocumentRoot /var/www
  <Location /login>
    AuthType Basic
    AuthName "Login-Bereich"
    AuthUserFile /etc/apache2/login-users
    Require valid-user
  </Location>
  SSLEngine on
  SSLCertificateFile /etc/ssl/certs/www.example.
    com-cert.pem
  SSLCertificateKeyFile /etc/ssl/private/www.
    example.com-key.pem
</VirtualHost>
```


Die Konfiguration des VHosts mit aktiviertem SSL/TLS ist um die Benutzer-Authentifizierung für das Unterverzeichnis `login` der Webseite erweitert worden. Der `<Location /login>` Tag verhält sich relativ zum `DocumentRoot` Pfad, d. h. der komplette Pfad ist `/var/www/login`. Im Gegensatz dazu ist die Pfad-Angabe des `<Directory>` Tags absolut.

In diesem Beispiel ist das Unterverzeichnis `login` über HTTP weiterhin **ohne** Authentifizierung erreichbar. Dies muss in der VHost-Konfiguration durch die Option `Deny From All` für das Verzeichnis verhindert werden.

`.htaccess` Konfiguration

`.htaccess`

- ermöglicht das Setzen von Optionen für ein Verzeichnis über eine spezielle Verzeichnis-gebundene Datei
- entspricht den `<Directory>`-Optionen in der VHost Konfiguration
- sollte nach Möglichkeit vermieden werden, jedoch hilfreich bei Web-Hosting
- die Option `Allow Override All` muss für das Verzeichnis gesetzt sein
- wird häufig für Authentifizierung benutzt

Mit Hilfe der `.htaccess` Dateien kann außerhalb der Konfigurationsdateien des Apache HTTP-Servers das Verhalten von Verzeichnissen einer Webseite geändert werden. Die Anweisungen in dieser Datei entsprechen den Optionen im `<Directory>` Abschnitt der Konfiguration, setzt jedoch voraus, dass die Option `Allow Override All` für das Verzeichnis gesetzt ist, da diese Dateien ansonsten ignoriert werden.

Bei Webhostern wird diese Art der Konfiguration häufig angewandt, da die Benutzer keine Schreibrechte auf die globale Konfigurationsdatei haben, u. U. jedoch einzelne Verzeichnisse mit einem Passwort schützen oder andere Einstellungen ändern wollen.

Lab 4.3: Authentifizierung einrichten

Lab: Authentifizierung einrichten

- Authentifizierung mit SSL/TLS für einen Unterbereich konfigurieren
- Passwortdatei erstellen und Benutzer hinzufügen

CGI

CGI

- stellt eine standardisierte Schnittstelle zwischen dem Webserver und einer anderen Anwendung bereit
- entspricht heutzutage für viele Programmiersprachen nicht mehr dem Stand der Technik
- Alternativen: z. B. `mod_wsgi` für Python oder „Ruby on Rails" für Ruby

Common **G**ateway **I**nterface (CGI) wurde 1993 als standardisierte Schnittstelle zwischen Web-Server und Dritt-Anwendungen entwickelt. Heutzutage ist dieser Standard in vielerlei Hinsicht überholt und es existieren zahlreiche Alternativen. Zum Ausführen einfacher Programme oder Scripte stellt CGI aber weiterhin eine einfach einzurichtende Lösung dar.

CGI

CGI

- ermöglicht dynamische Inhalte zu generieren oder Webanwendungen zu realisieren
- Code in beliebiger Programmiersprache kann auf dem Server ausgeführt werden
- Code befindet sich in einem besonderem Verzeichnis
- Scripte müssen ausführbar sein

Die Hauptanwendung von CGI und den Nachfolgern ist es, Server-seitig Code beliebiger Programmiersprachen (und damit beliebige Programme) auszuführen. Dieser Code muss sich in einem besonderen Verzeichnis befinden und ausführbar sein.

CGI

Konfigurationsbeispiel

- Beispiel-Konfiguration für CGI:

```
1 ScriptAlias /cgi-bin/ /srv/cgi-bin/
2 <Directory /srv/www1>
3 ...
4 </Directory>
5 <Directory /srv/cgi-bin>
6   Options ExecCGI
7   SetHandler cgi-script
8 </Directory>
```


Die Option `ScriptAlias` gibt an, wohin der Scripte-Aufruf geleitet wird. Bei dem Aufruf `http://www.example.com/cgi-bin/foo.bar` wird das Script `/srv/cgi-bin/foo.bar` ausgeführt, und nicht ein Script `/srv/www1/cgi-bin/foo.bar`.

Außerdem müssen für das Verzeichnis, in dem sich die Scripte befinden, Optionen gesetzt werden, die das Ausführen von Code erlauben.

Lab 4.4: CGI einrichten

Lab: CGI einrichten

- CGI für einen VHost aktivieren und Test-Script einrichten

PHP

PHP

- als Modul für den Apache HTTP-Server verfügbar
- wurde 1995 als Framework für dynamische Web-anwendungen veröffentlicht
- heutzutage weit verbreitet, u. a. basieren viele Foren-Software, Content-Management-Systeme, etc. darauf
- im HTML eingebetteter PHP-Code wird Server-seitig ausgeführt und das Ergebnis an den Client geschickt

Lab 4.5: PHP einrichten

Lab: PHP einrichten

- PHP installieren und Test-Programm einrichten