

Apache HTTP-Server Teil 2

Linux-Kurs der Unix-AG

Zinching Dang

04. Juli 2014



UNIX
AG

TU Kaiserslautern

RH Regionales
Hochschul-
Rechenzentrum **RK**
Kaiserslautern

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

Konfigurationsbeispiel

► Beispiel-Konfiguration:

```
1 <IfModule mod_ssl.c>
2 <VirtualHost *:443>
3     ServerName www.example.com
4     DocumentRoot /var/www
5     <Location /login>
6         AuthType Basic
7         AuthName "Login-Bereich"
8         AuthUserFile /etc/apache2/login-users
9         Require valid-user
10    </Location>
11    SSLEngine on
12    SSLCertificateFile /etc/ssl/certs/www.example.
        com-cert.pem
13    SSLCertificateKeyFile /etc/ssl/private/www.
        example.com-key.pem
14 </VirtualHost>
```

.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

Lab: Authentifizierung einrichten

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

- ▶ 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

- ▶ 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

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>
```


Lab: CGI einrichten

- ▶ CGI für einen VHost aktivieren und Test-Script einrichten

- ▶ als Modul für den Apache HTTP-Server verfügbar
- ▶ wurde 1995 als Framework für dynamische Webanwendungen 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: PHP einrichten

- ▶ PHP installieren und Test-Programm einrichten