

Aufgabe 1

Apache 2 Server (Version 2.2.3) installiert. Und an Port 18080 gebunden mittels Listen im httpd.conf Konfigurationsdatei.

Adresse: 134.60.209.103:18080

Im httpd.conf Konfigurationsdatei hab ich festgelegt, wer auf dem Server zugreifen darf, so dass man nur innerhalb der Universität zugreifen kann.

```
<Directory /apache/htdocs >  
AllowOverride None  
Order deny,allow  
Deny from all  
Allow from 134.60.0.0/16  
</Directory>
```

Eine HTML-Datei mit Bild erstellt.

Aufgabe 2

HTTP/1.1 sieht eine Methode zur Benutzerauthentisierung vor.

Dazu werden Ressourcen auf der Seite des Webservers zu sogenannten Realms gruppiert.

Alle Ressourcen eines Realms sind mit den gleichen Zugriffsrechten versehen und erfordern die gleiche Art der Authentisierung. In der Konfiguration des Webservers wird festgelegt, welche Ressourcen bzw. URLs oder Dateien zu welchem Realm gehören und welche Benutzer darauf zugreifen können.

Um Zugriff auf die Ressourcen eines Realms zu erhalten, die einer Authentisierung bedürfen, sendet die Clientsoftware des Benutzers beim Zugriff auf eine solche Ressource im HTTP-Request einen sogenannten Authorization Header mit, der die für den Zugriff nötigen Authentisierungsdaten (z. B. Benutzername und Passwort) enthält.

HTTP/1.1 sieht zwei verschiedene Methoden zur Benutzerauthentisierung vor:

1. Die sogenannte Basic-Access-Authentisierung.

Dabeisendet der Client den Benutzernamen und das Passwort kodiert im Authorization Header. Das Passwort ist somit zwar auf den ersten Blick nicht entzifferbar, aber ohne Probleme auswertbar, da unverschlüsselt.

2. Die Digest-Authentisierung.

Dabei erhält der Client vom Server einen Zufallsstring, die sogenannte Challenge. Aus dieser Challenge und dem Passwort des Benutzers errechnet der Client nach einem standardisierten Verfahren einen sogenannten Digest, der dann zur Authentisierung an den Server gesandt wird. Da der Server sowohl über den von ihm generierten Zufallsstring, als auch über das Passwort des Benutzers verfügt, kann er den Digest ebenfalls berechnen und so die Authentisierung durchführen. Wird Digest-Authentisierung verwendet, so

enthält die Antwort des Servers an einen Client, der einen Request ohne Verwendung eines Authorization Headers sendet, u. a. folgende Daten:

- das Realm der Authentisierung
- die Domain der Authentisierung: eine Liste von URLs, an die der Client die gleichen Authentisierungsdaten senden kann
- ein vom Server generierter String Nonce, der für jede Autorisierungsanfrage eindeutig neu erzeugt werden sollte.

Aufgabe 3:

URI:

Uniform Resource Identifier (URI) (engl. „einheitlicher Bezeichner für Ressourcen“) ist eine Zeichenfolge, die zur Identifizierung einer abstrakten oder physikalischen Ressource dient. *URIs* werden zur Bezeichnung von Ressourcen (wie Webseiten, sonstigen Dateien, Aufruf von Webservices, aber auch z. B. E-Mail-Empfängern) im Internet und dort vor allem im WWW eingesetzt.

URL:

Uniform Resource Locator identifiziert eine Ressource über ihren primären Zugriffsmechanismus, gibt also den *Ort* (engl. *location*) der Ressource im Netz an. Beispiele hierfür sind `http` oder `ftp`. URLs waren ursprünglich die einzige Art von URIs, weshalb der Begriff URL oft als gleichbedeutend zu URI verwendet wird.

URN:

Uniform Resource Names mit dem *URI*-Schema `urn` identifizieren eine Ressource mittels eines vorhandenen oder frei zu vergebenden Namens, z. B. `urn:isbn`

Bsp:

`urn:isbn:3-446-22149,2`
`urn:fipa:language:ac1`
`http://www.uni-ulm.de`