Apache

Webseiten und andere Daten aus dem WWW (»World Wide Web«) wandern in der Regel per HTTP (»Hypertext Transfer Protocol«, meistens Port 80) bzw. HTTPS (»HTTP Secure«, Port 443, für verschlüsselte HTTP-Verbindungen) durchs Netz.

Ruft ein Benutzer mit einem Client, also z. B. einem Web-Browser, eine Seite auf kontaktiert die Anwendung den verantwortlichen Webserver (»Request«) und erhält daraufhin eine entsprechende Antwort (»Response«).

Die Webserver-Software regelt dabei die Datenablage, wie diese übertragen werden und wer Zugriff auf das Angebot hat. Neben statischen Elementen, wie z. B. HTML-Seiten, Stylesheets oder Bilder, werden Web-Inhalte auch immer häufiger dynamisch generiert. Dabei greift der Webserver auf externe Programme (z. B. PHP-, Perl- oder J SP-Skripte) zurück. Diese sind kein Bestandteil des Servers, sondern werden entweder als Module oder eigenständige CGI-Prozesse aufgerufen.

Der Apache Webserver

Die am weitesten verbreitete Server-Software ist der von der Apache Software Foundation entwickelte open source Webserver Apache. Apache gibt es für zahlreiche Betriebssysteme, darunter Mac OS X, Windows, OS/2 und viele Unix-Varianten.

Konfigurationsdateien

Im Ordner `/etc/apache2` und seinen Unterverzeichnissen finden Sie alle Konfigurationsdateien des Webservers. Die Haupteinrichtungsdatei ist auf Debian-Systemen die Datei `apache2.conf`: Diese enthält sinnvolle Voreinstellungen (so dass Sie hier in der Regel nichts verändern müssen) und viele ausführliche Erklärungen und Beispiele, die durch ein Rautezeichen am Zeilenanfang auskommentiert sind.

Datei/Verzeichnis	Funktion
`/etc/apache2/apache2.conf`	Zentrale Datei: hier werden Arbeitsumgebung, das Multiprozessing—Modul und die Logfiles eingerichtet; weitere Konfigurationsabschnitte werden durch `Include` eingebunden.
`/etc/apache2/ports.conf`	Teilt dem Webserver den Port mit, auf dem er lauschen soll (Voreinstellung 80); automatisch durch `include` verfügbar.
`/etc/apache2/sites-available/`	Hier finden Sie Konfigurationsdateien für die virtuellen Hosts (Virtual Hosts); das Verzeichnis wird nicht durch `include` eingebunden.
`/etc/apache2/sites-enabled/`	Hier liegen Symlinks auf die gleichnamigen Dateien in sites -available. Standardmässig ist hier der Link für den Virtual Host `default`. welcher für die Anzeige von `http://localhost/` verantwortlich ist. Ordner wird durch `include` automatisch eingebunden.
`/etc/apache2/ssl/`	Platz für SSL-Zertifikate, wird nicht automatisch eingebunden.

Logfiles

In verschiedenen Konfigurationsdateien spezifizieren Sie, welche Ereignisse der Webserver protokolliert und wo diese Logfiles landen. Standardmässig enthält die Hauptkonfigurationsdatei apache2.conf den Eintrag, der definiert, wo sämtliche Fehlermeldungen landen:

```
ErrorLog /var/log/apache2/error,log
```

Apache – Konfiguration

Namensbasierte Virtuelle Hosts sind die gängigste Methode zur Konfiguration virtueller Hosts und werden für eine gemeinsame IP-Adresse eingerichtet. Apache entscheidet anhand des HTTP-Headers, welche Seite ausgeliefert wird. Hierfür ist es aber notwendig, dass der Client das HTTP 1.1 Protokoll unterstützt (was bei allen gängigen Webbrowsern der Fall ist)!

Im Verzeichnis **/etc/apache2/sites-available/** befindet sich die Datei **default**. Diese kann man als Vorlage für die weiteren virtuellen Hosts verwenden. Für jeden virtuellen Host wird dann eine eigene Datei angelegt.

Auch sollte die **default**-Site nicht deaktiviert werden, da die darin enthaltene "NameVirtualHost"-Direktive erst den Betrieb von Virtual Hosts möglich macht.

Der <virtualhostname> Dateiname **muss** zwingend auf **.conf** enden! Ansonsten wird bei Aktivierung des VirtualHost mit **a2ensite** ein Fehler generiert.

/etc/apache2/ports.conf

```
NameVirtualHost *:80
```

```
NameVirtualHost *:80
```

Anpassen der Datei:

/etc/apache2/sites-available/000-default.conf → Apache 2.4

```
<VirtualHost *:80>
      ServerAdmin root@localhost     // E-Mail-Adresse des Server-Administrators
      DocumentRoot /www              // das Basisverzeichnis des Virtual Hosts.

      ErrorLog ${APACHE_LOG_DIR}/error.log
      CustomLog ${APACHE_LOG_DIR}/access.log combined

      <Directory "/www">             // Für den Ordner Einstellungen und Berechtigungen vergeben
            Require all granted      // Alle IP-Adressen sind berechtigt
      </Directory>
</VirtualHost>
```

Löschen der ssl-Konfiguration

```
rm /etc/apache2/sites-available/default-ssl.conf     // wen ssl nicht verwendet wird
```

Anpassen der Datei für den virtuellen Host tsbe.dmz

/etc/apache2/sites-available/www.tsbe.dmz.conf → Apache 2.4

```
<VirtualHost *:80>
      ServerName www.tsbe.dmz          // primärer Server-Name unter welchem die Seite erreichbar ist
      ServerAdmin root@localhost       // E-Mail-Adresse des Server-Administrators
      DocumentRoot /www/tsbe.dmz       // das Basisverzeichnis des Virtual Hosts. Es sollte nun mit
                                       // http://www.tsbe.dmz ein Zugriff versucht werden

      <Directory "/www/tsbe.dmz">      // Für den Ordner Einstellungen und Berechtigungen vergeben
            Require all granted        // Alle IP-Adressen sind berechtigt (löschen bei Authentifizierung)
            AllowOverride None         // Aktiviert .htaccess
            AuthType Basic             // Basis Authentifizierung
                  AuthName "Kunde-Downloadbereich"     // Fenstertitel
                  AuthUserFile /etc/apache2/passwords/.htpasswd    // Passwortdatei
                  AuthGroupFile /etc/apache2/passwords/.htgroup    // Gruppendatei
                  Require group Gruppe2          // Berechtigung für Gruppe2 setzen
      </Directory>
</VirtualHost>
```

Aktivieren der Seite

```
a2ensite www.tsbe.dmz
```

Passwörter für ein Verzeichnis erstellen

Schlüssel auf DNS-Server berechtigen

```
VmLS3:~# htpasswd -c /etc/apache2/passwords/.htpasswd User1
                                                       // Benutzername
                  // Passwortdatei mit "." verstecken
            // Datei erstellen, nur beim ersten User verwenden, sonst wird die Datei wieder überschrieben
      // Programm htpasswd zur Erstellung von Passwörter


New password:****
Re-type new password:****
Adding password for user User1
vmLS3:~#
```

Erstellen einer Gruppendatei (mit Editor)

/etc/apache2/passwords/.htgroup

```
Gruppe2: User1 user2 user3
test: user3 user4
  |       |
  |       // Users
  // Gruppe
```

Allgemeines

Hilfreiche Befehle

```
root@vmLS3:/# apt-get install apache2 apache2-utils   // apache2 installieren
root@vmLS3:/# /etc/init.d/apache2 restart             // Restart apache
root@vmLS3:/# a2ensite www.tsbe.dmz                   // Virtuellen Host www.tsbe.dmz aktivieren
root@vmLS3:/# a2enmod rewrite                         // Modul für passwörter aktivieren
root@vmLS3:/# a2enmod authz_groupfile                 // Modul aktivieren für gruppendatei
```

HTTPS konfigurieren

Um Webseiten verschlüsselt auszuliefern, benötigt Ihr Webserver zunächst ein Serverzertifikat. Zu Testzwecken oder für den Eigenbedarf können Sie sich ein solches Zertifikat schnell und einfach selbst erstellen. Sie müssen lediglich damit leben, dass Ihr Browser beim Ansurfen der HTTPS-geschützten Webseite eine Warnung ausgibt. weil das selbst erstellte Zertifikat nicht von einer vertrauenswürdigen Instanz signiert wurde. Da unbedarfte Nutzer von dieser Warnung leicht abgeschreckt werden. sollten Sie für Ihre Produktivsysteme auf signierte Zertifikate kommerzieller Anbieter zurückgreifen. Für die Beispiele in diesem Kapitel genügt natürlich ein selbst erstelltes Zertifikat.

Falls das noch nicht geschehen ist, installieren Sie jetzt die Pakete apache2 und openssl. Die Konfigurationsdateien für den Apache-Server finden Sie nach der Installation unter `/etc/apache2/`.

Erstellen Sie als root unterhalb dieses Pfades ein neues Verzeichnis mit dem Namen ssl, und wechseln Sie hinein. Dieses Verzeichnis soll unser Serverzertifikat beheimaten.

```
mkdir /etc/apache2/ssl
cd /etc/apache2/ssl
```

Mit dem folgenden Kommando erstellen Sie das Serverzertifikat:

```
openssl req -new -x509 -keyout myserver.pem -out myserver.pem -days 365 -nodes
```

OpenSSL stellt Ihnen eine Reihe von Fragen zu Ihrer Location und Kontaktadresse, die Sie nach Belieben beantworten können oder auch nicht -bei einem selbst signierten Testzertifikat sind diese Informationen nicht wesentlich. Passen Sie aber auf, wenn die Frage nach dem `Common`

Name erscheint! Hier müssen Sie unbedingt den exakten Namen eingeben, unter dem Ihr HITPS-Webangebot später erreichbar sein soll. Wollen Sie Ihre Webseite also unter https://www.example.com aufrufen können, so müssen Sie auf die Frage nach dem Common Name mit www.example.com antworten.
Im nächsten Schritt aktivieren Sie Apaches SSL-Modul. In der Datei /etc/apache2/ports.conf finden Sie einen Abschnitt, der so oder sehr ähnlich aussieht:

```
#(ifModule mod_ssl.c>
# Listen 443
#<ifModule>
```

Führen Sie nun die beiden folgenden Kommandos aus, um das SSL-Modul zu aktivieren und Apache neu zu starten:

```
a2enmod ssl
/etc/init.d/apache2 restart
```

Apache ist jetzt grundsätzlich in der Lage, HTTPS-Anfragen zu bedienen. Sie benötigen also nur einen virtuellen Host, der sie auch beantwortet. Legen Sie unter /etc/apache2/sites-available die Datei example.com mit folgendem Inhalt an:

```
<VirtualHost *:443>
      ServerName www.example.com
      ServerAdmin webmaster@example.com
      DocumentRoot /var/www/example.com/
      SSLEngine On
      SSLCertificateFile /etc/apache2/ssl/myserver.pem
      <Directory /var/www/exampie.com/>
            Options Indexes FoiiowSymLinks
      </Directory>

      ErrorLog /var/iog/apache2/error.log
      CustomLog /var/log/apache2/exampie,com.iog combined
</VirtuaiHost>
```

Ihr virtueller SSL-Host ist nun fertig konfiguriert, Sie müssen ihn nur noch aktivieren. Dazu genügen diese beiden Kommandos:

```
a2ensite example.com
/etc/init.d/apache2 force-reioad
```

Apache-Server mit ModSecurity schützen
ModSecurity nennt sich selbst >>Web Application Firewall«. Es ist ein Apache-Modul, das den HTTP-Datenverkehr mithilfe eines Regelwerks auf Angriffsmuster prüft. Der Unterschied zu klassischen Firewalls besteht in der OSI-Schicht, auf der der Filter arbeitet.
Die Motivation hinter dieser Vorgehensweise ist eine Verschiebung der Angriffsvektoren. Früher zielten Angreifer darauf ab, Sicherheitslücken in der Serversoftware auszunutzen.

WebDAV

Dem Protokoll HTTP mangelt es vor allem an Mechanismen für die gemeinsame Arbeit an Dokumenten (kooperative Dokumentenpflege) im Netz. Um die Nutzung des Internets als universellen Informationsspeicher zu ermöglichen, musste das HTTP-Protokoll um einige Operationen erweitert werden. Diese Erweiterung nannte man WebDAV (Web based Distributed Authoring and Versioning).
Durch WebDAV bietet sich die Möglichkeit, direkt auf die Inhalte von Dokumenten über das Internet zuzugreifen und dabei im Team den Inhalt eines Dokuments zu ändern. Insgesamt bietet das WebDAV eine ganze Palette von Funktionen, um Daten, Dokumente und sonstige

Informationsobjekte über das Internet austauschen zu können.

Uebung 1) Erstellen Sie eine einfache, passwortgeschützte Webseite in der Zone intern