Leander Kirstein-Heine

mit virtuellen Servern und SSL

Jena, 06.03.2002

1. Virtuelle Server

a. Einleitung

HTML als plattformunabhängiges Dokumentenformat ist ideal dazu geeignet Informationen sowohl firmenintern als auch weltweit über das Internet zu verbreiten. Somit haben auch Webserver immer mehr an Bedeutung gewonnen.

Für den Besucher der Webseiten z.B. auf einem Firmenwebserver ist es oft einfacher, sich eine Adresse in der Form *www.unternehmensbereich.de* zu merken, als mit langen Pfadangaben operieren zu müssen. Weiterhin macht es auch keinen Sinn für die kleinste Internetpräsenz jeweils einen eigenen, alleinstehenden Server einrichten zu müssen.

Aus diesem Grund untersützt der Apache seit der Version 1.1 die Einrichtung von virtuellen Servern, die nach aussen wie ein selbstständiger Webserver auftreten.

b. Was sind virtuelle Server?

Virtuelle Server sind sozusagen Nebenserver, die vom selben Webserver verwaltet werden, aber eine andere Dokumentenwurzel benutzen. Dazu kommt, daß diese Server auch noch entweder über andere IP-Adressen oder andere DNS-Namen ansprechbar sind.

Die Definition eines virtuellen Servers ermöglicht es also dem Webserver zu entscheiden, an welchen DNS-Namen bzw. welche IP-Adresse die Anfrage ging. Diese Entscheidung führt dann zu unterschiedlichen Verzeichnissen (*document roots*); es also werden unterschiedliche Seiten angezeigt. Fällt der Webserver die Entscheidung nach dem DNS-Namen, spricht man von *name-based*, bei der Entscheidung nach der IP-Adresse von *ip-based virtual hosts*. Diese Server werden vom selben Eltern-Prozeß gestartet.

c. Kompilieren und Installieren des Apache

Bei allen gängigen Distributionen ist der Apache standardmäßig dabei und nach der Installation in der Regel sofort einsetzbar. Sollte dies nicht der Fall sein, steht die aktuelle Version des Apache unter der Adresse http://www.apache.org/dist/httpd/ oder einem aktuellen Mirror als tar-ball zum Download bereit. Nach dem Entpacken in ein Verzeichnis kann die Installation am einfachsten mit folgenden Anweisungen gestartet werden:

```
$ ./configure --prefix=/usr/local/apache
$ make
$ make install
$ /usr/local/bin/apachectl start
```

Der Prefix gibt das Verzeichnis an, in das der Apache installiert werden soll. Nach dem Kompilieren müsste sich der Apache, falls keine Fehler aufgetreten sind, mit */usr/loacal/bin/apache start* starten lassen und auf dem Rechner selbst mit *http://localhost/* im Browser die Begrüssungsseite erscheinen.

Weitere Einzelheiten zur Installation finden sich in den Hilfetexten des Apache.

d. Einrichten der virtuellen Server

i. IP-based Hosts

Für jeden virtuellen Server muss, wie der Name schon vermuten läßt, eine eigene IP-Adresse vorhanden sein. Weiterhin muss der Server unter dieser Adresse auch ansprechbar sein. Dies kann über mehrere Netzwerkkarten realisiert werden, oder dadurch, dass eine einzige Ethernetkarte mehrere IP-Adressen bekommen kann. Dazu wird der Bezeichnung der Ethernetkarte (z.B. eth0) einfach ein Doppelpunkt und eine Nummer zugefügt (eth0:1, eth0:2, ...). Jeder dieser "virtuellen Netzwerkkarten" kann jetzt eine eigene IP-Adresse vergeben werden. Entweder mit dem entsprechenden Konfigurationsprogramm (yast, linuxconf,...) oder mit dem Befehl:

```
ifconfig eth0:1 Adresse netmask Maske
```

Ab einer gewissen Anzahl (ca. ab 4 virtuellen Hosts) bietet es sich aber an, namensbasierte Hosts zu generieren, statt mit x virtuellen Netzwerkkarten zu arbeiten.

Danach kann mit der Konfiguration der virtuellen Server begonnen werden. Dazu ist nur die zentrale Konfigurationsdatei *httpd.conf* zu editieren (seit Version 1.3.4 werden alle Einträge nur noch in dieser Datei vorgenommen, die srm.conf und access.conf sind leer bzw. enthalten den Hinweis auf die httpd.conf). Diese Datei finden Sie standardmäßig unterhalb von */usr/local/apache/conf* oder /*etc/httpd*.

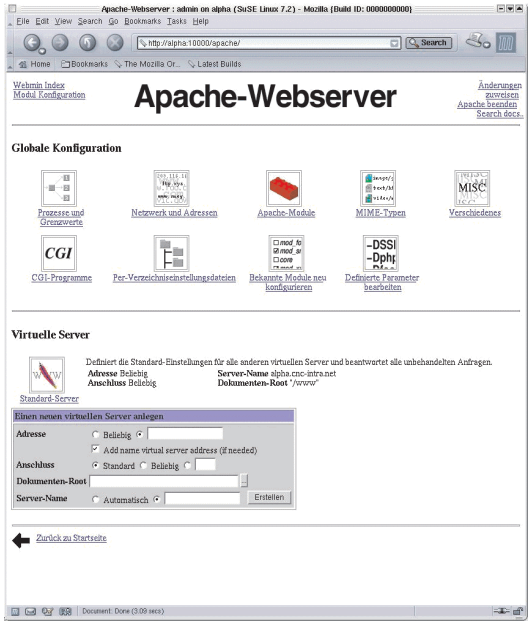
In dieser Datei finden Sie standardmäßig schon ein Beispiel für die Einträge der virtuellen Server (Section 3). Dies müssen Sie nun Ihren Andorderungen entsprechend anpassen. Dazu soll folgendes Beispiel dienen:

```
<VirtualHost 192.168.11.101>
   ServerAdmin root@mydomain.de
   DocumentRoot /www1/htdocs
   ScriptAlias /cgi-bin/ "www1/cgi-bin/"
   ServerName www1.mydomain.de
</VirtualHost>

<VirtualHost 192.168.11.102>
   ServerAdmin hans@marvin.mydomain.de
   DocumentRoot /www2/htdocs
   ScriptAlias /cgi-bin/ "www2/cgi-bin/"
   ServerName www2.mydomain.de
</VirtualHost>
```

Nach den Änderungen können die Webseiten für www1.mydomain.de in das Verzeichnis /www1/htdocs und für www2 in das Verzeichnis /www2/htdocs eingespielt werden. Jeder der hier eingerichteten virtuellen Server verfügt über ein eigenes cgi-bin Verzeichnis zum Ausführen von Skripten. Die Webseiten müssen weltweit lesbar sein, die Skripte für alle ausführbar sein.

Wenn dies alles geschehen ist, kann die neue Konfiguration über
/usr/local/bin/apache restart | gracefull
neu eingelesen werden.

Alternativ können virtuelle Server auch über das Administrationstool "Webmin" eingerichtet werden:



Dazu sind nur die oben genannten Einträge in die entsprechenden Formularfelder einzutragen. Die Änderungen sind hier durch Anklicken des Links "Änderungen zuweisen" zu übernehmen.

In jedem Fall sollten die Webseiten im Browser nach der Eingabe der IP-Adressen escheinen.

ii. Named-based Hosts

Die Konfiguration wird in vergleichbarer Weise vorgenommen, nur dass ein Eintrag "*NameVirtualHost*" mit der IP-Adresse den Einträgen für die virtuellen Server voranzustellen ist:

```
NameVirtualHost 192.168.11.100

<VirtualHost 192.168.11.100>
    ServerAdmin root@mydomain.de
   DocumentRoot /www1/htdocs
   ScriptAlias /cgi-bin/ "www1/cgi-bin/"
    ServerName www1.mydomain.de
</VirtualHost>

<VirtualHost 192.168.11.100>
   ServerAdmin hans@marvin.mydomain.de
   DocumentRoot /www2/htdocs
   ScriptAlias /cgi-bin/ "www2/cgi-bin/"
    ServerName www2.mydomain.de
</VirtualHost>
```

Auch diese Einstellung kann selbstverständlich über Webmin vorgenommen werden.

Damit diese Server auch über ihren Namen erreichbar sind, sind sie im Nameserver einzutragen bzw. in den hosts-Dateien der Clients.

2. Apache mit SSL

a. Einleitung

Wenn es darum geht empfindliche Daten wie z.B. Bankverbindungen, Kreditkartendaten etc. über einen Webserver zu transportieren, so sollte die Verbindung verschlüsselt werden. Im Gegensatz zu einer normalen Netzwerkverbindung ist es bei einer SSL-Verschlüsselung fast unmöglich, die Daten einfach auszulesen.

Im folgenden werden die dazu notwendigen Schlüssel selbst generiert. Für einen produktiven Webserver im Internet sollten die Schlüssel von einer registrierten Stelle (CA) signiert werden, da sonst der Browser eine entsprechende Meldung ausgibt und der Besucher der Seite wohlmöglich mißtraut.

a. Konfiguration des Apache

i. Benötigte Software

Ausser dem Apache selbst benötigt man noch folgende Pakete:

mod_ssl:
zum Patchen des Apache für die Schnittstellen zum OpenSSL-Paket (die Apache-Quellen müssen daher zuerst enpackt werden). Nach dem Entpacken der Quellen wird das Modul mit folgendem Befehl konfiguriert und die Apache-Quellen verändert:

```
sh ./configure --with-apache=../apache
```

Bei der Option --with-apache= muß der Pfad angegeben werden, der auf die zuvor entpackten Apache-Quellen weist. Verlief die Ausführung des vorstehenden Kommandos erfolgreich, wird eine abschließende Meldung ausgegeben:

```
Done: source extension and patches successfully applied.
```

OpenSSL:
Das Paket stehe unter www.openssl.org zum Download bereit. Nach dem Entpacken kann es mit

```
./config --prefix=/usr/local
```

konfiguriert und mit

```
make
make install
```

übersetzt und installiert werden.

ii. Erstellen der Sicherheitszertifikate

Vor der eigentlichen Konfiguration des Apache sind die Zertifikate zu erstellen. Dies geschieht mithilfe von OpenSSL. Dazu sind folgende Schritte notwendig:

- ☆ Erstellen eines Verzeichnisses unterhalb des Verzeichnisses für die Konfiguration, (z.B. *schluessel)*

- ☆ Das Zertifikat wird mit folgenden Anweisungen erstellt:

  Zuerst muss ein Request zu erzeugt werden, der anschließend signiert wird:

  ```
  openssl req -new > server.crt
  ```

  Nach Eingabe des Befehls müssen folgende Angaben gemacht werden (die Eingaben sind nur beispielhaft):

  ```
  Generating a 1024 bit RSA private key
  .............++++++
  ..........................++++++
  writing new private key to 'privkey.pem'
  Enter PEM pass phrase:
  Verifying password - Enter PEM pass phrase:
  -----
  You are about to be asked to enter information that will be
  incorporated
  into your certificate request.
  What you are about to enter is what is called a Distinguished
  Name or a DN.
  There are quite a few fields but you can leave some blank
  For some fields there will be a default value,
  If you enter '.', the field will be left blank.
  -----
  Country Name (2 letter code) [AU]:DE
  State or Province Name (full name) [Some-State]:Thueringen
  Locality Name (eg, city) []:Jena
  Organization Name (eg, company) [Internet Widgits Pty
  Ltd]:Computer & Network Consulting
  Organizational Unit Name (eg, section) []:Webhosting
  Common Name (eg, YOUR name) []:gamma.cnc-intra.net
  Email Address []:webmaster@gamma.cnc-intra.net

  Please enter the following 'extra' attributes
  to be sent with your certificate request
  A challenge password []:
  An optional company name []:
  ```

  Das Passwort sollten Sie sich umbedingt merken; weiterhin sollte bei Common Name unbedingt der Server-Name angegeben werden.

  Anschließend wird mit dem erzeugten Request (privkey.pem) der RSA-Schlüssel geschrieben:

  ```
  openssl rsa -in privkey.pem -out server.key
  ```

  Hier müssen Sie nun wieder das oben eingegebene Passwort eingeben:

  ```
  read RSA key
  Enter PEM pass phrase:
  writing RSA key
  ```

Mit der Anweisung

```
openssl x509 - in server.crt -out server.cert
           -req -signkey server.key -days 365
```

wird dieser Schlüssel für 365 Tage signiert. Dies wird mit der Meldung

```
subject=/C=DE/ST=Thueringen/L=Jena/O=Computer &
Network Consulting/OU=Webhosting/CN=gamma.cnc-
intra.net/Email=webmaster@gamma.cnc-intra.net
Getting Private key
```

bestätigt.

Soll der Request (hier *privkey.pem*) einer CA zum signieren vorgelegt werden, genügt es nur diesen zu erzeugen.

iii. Konfiguration des Apache

Die Konfiguration des Apache erfolgt wieder in der zentralen Konfigurationsdatei *httpd.conf*. Dazu muss dem Apache zuerst gesagt werden, auf welche Ports er hören soll. Dies wird durch folgenden Eintrag vorgenommen:

```
Listen 80

<IfDefine SSL>
     Listen 443
</IfDefine>
```

Weiterhin ist natürlich dafür zu sorgen, dass die SSL-Module geladen werden:

```
<IfDefine SSL>
  LoadModule ssl_module           libexec/libssl.so
 </IfDefine>

<IfDefine SSL>
    AddModule mod_ssl.c
</IfDefine>
```

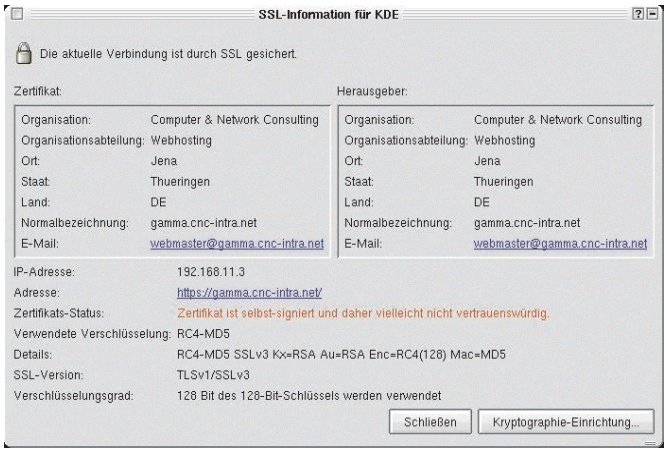
Im weiteren ist für den SSL-Zugang ein virtueller Host anzulegen:

```
##
## SSL Virtual Host Context
##

<VirtualHost 192.168.11.3:443>

#  General setup for the virtual host

DocumentRoot "/usr/local/httpd/shtdocs"
ServerName gamma.cnc-intra.net
ServerAdmin webmaster@gamma.cnc-intra.net
ErrorLog /var/log/httpd/error_log
TransferLog /var/log/httpd/access_log

#   SSL Engine Switch:
#   Enable/Disable SSL for this virtual host.

SSLEngine on

#   Server Certificate:
#   Point SSLCertificateFile at a PEM encoded certificate.  If
#   the certificate is encrypted, then you will be prompted for a
#   pass phrase.  Note that a kill -HUP will prompt again. A test
#   certificate can be generated with ‘make certificate’ under
#   built time. Keep in mind that if you’ve both a RSA and a DSA
#   certificate you can configure both in parallel (to also allow
#   the use of DSA ciphers, etc.)

SSLCertificateFile /etc/httpd/schluessel/server.cert

#   Server Private Key:
#   If the key is not combined with the certificate, use this
#   directive to point at the key file.  Keep in mind that if
#   you’ve both a RSA and a DSA private key you can configure
#   both in parallel (to also allow the use of DSA ciphers, etc.)

SSLCertificateKeyFile /etc/httpd/schluessel/server.key

</VirtualHost>
</IfDefine>
```

Nach dem Neustart des Apache mit

```
apachectl stop
apachectl startssl
```

läuft der Apache nun mit SSL. Das selbst erstellte Zertifikat wird jedoch von den meisten Browsern nicht als vertrauenwürdig eingestuft, da es selbst signiert wurde:



Literaturverzeichnis:

Apache Virtual Host documentation, The Apache Software Foundation,
http://httpd.apache.org/docs/vhosts/index.html

Das SSL-Apache Handbuch, DFN-PCA, DFN-CERT GmbH, Oberstraße 14b, D-20144 Hamburg,
http://www.dfn-pca.de/certify/ssl/handbuch/sslapache1_3/ssla13.html