

Ausarbeitung zum

Proseminar "*Website-Management-Systeme*"
im Wintersemester 2003/2004

Thema „*Web-Server: Apache/Roxen*“

von Oliver Roeschke

Inhaltsverzeichnis:

1. Motivation
2. Was ist ein Web-Server?
3. Das HTTP-Protokoll
 - 3.1 Die Geschichte des HTTP-Protokolls
 - 3.2 Die Anfrage per HTTP
 - 3.3 HTTPS-Verbindungen
4. Die Verarbeitung im Web-Server Apache
5. Was ist HTML?
6. Fachbegriffe
7. Quellenangaben

Kapitel 1: Motivation

Für das Verständnis was ein Content-Management-System leistet, welche Techniken es einsetzt, wo seine Grenzen liegen ist es unerlässlich einen kurzen Einblick in die wichtigsten Techniken zu liefern, auf denen ein Content-Management-System aufbaut.

Um den Kreis der Techniken einzugrenzen beginne ich hier mit einer Erklärung was ein Web-Server ist, also welche Aufgaben er übernimmt.

Weiterhin wird die Übertragung einer Web-Site behandelt, da das Verständnis der zugrunde liegenden Funktionsweise elementar für das Verständnis des Grundlagenvortrages „Session-Management“ ist.

Auch nötig ist eine kurze Einsicht in HTML. So wird schnell deutlich welche Möglichkeiten zur Gestaltung einer Web-Seite HTML bietet, und welche nicht.

Kapitel 2: Was ist ein Web-Server?

Ein Web-Server (hier am Beispiel von Apache) ist eine Software die Dokumente (z.B.: Internet-Seiten, Downloads, Internetshops, etc.) in Computernetzwerken bereitstellen kann, und diese auf Anfrage, mit Hilfe des HTTP-Protokolls, überträgt.

Zur einfachen Anfragestellung an den Web-Server besitzt dieser eine eindeutige Adresse, die sich folgendermaßen zusammensetzt:

<Maschine>.<Domain>.<TLD>

z.B. „www.google.de“

Standard gemäß nimmt ein Web-Server Anfragen am Port 80 entgegen. Eingegangene Anfragen werden mit dem Versand des angeforderten Dokuments oder einer entsprechenden Fehlermeldung beantwortet.

Die angefragten Dokumente müssen dem Server nicht statisch im HTML-Format vorliegen, durch die Einbindung von Scriptsprachen (z.B.: PHP, Perl, etc.) ist es möglich diese dynamisch auf Anfrage zu erzeugen. Das Ergebnis einer dynamischen erzeugten Web-Seite ist eine HTML formatierte Seite, die nun zum anfragenden Client/ Browser übertragen wird.

Kapitel 3: Das HTTP-Protokoll

Das Kürzel HTTP steht für "**H**yper**T**ext **T**ransport **P**rotocol" und dient als Protokoll zur Übertragung von Daten. Hauptsächlich wird es genutzt zur Kommunikation zwischen Web-Servern und Browser, um ganze Web-Seiten mit den dazugehörigen Bildern, Animationen, Sounds, etc. zu übermitteln.

Kapitel 3.1: Geschichte des HTTP-Protokolls:

1989	<u><i>HTTP 0.9 :</i></u> Hierbei handelt es sich um eine HTTP-Version, die nur für den Transport von HTML-Dokumenten entwickelt und fast ausschließlich im WWW-Projekt am CERN eingesetzt wurde.
1990	<u><i>HTTP 1.0 :</i></u> Diese Version konnte bereits beliebige Datenformate übertragen und wurde außer beim CERN auch noch bei einigen anderen Institutionen eingesetzt.
Juli 1993	<u><i>1. HTTP-Internet-Entwurf („Basic HTTP“):</i></u> In dem Jahr wird das WWW durch den frei verfügbaren Browser Mosaic der breiten Öffentlichkeit bekannt gemacht.
Sept. 1995	<u><i>HTTP/1.0-Internet-Entwurf, 3. Fassung :</i></u> Dabei handelt es sich um die erste praxisorientierte Fassung, welche alle HTTP-Features beschreibt, die von einer WWW-Anwendung zu diesem Zeitpunkt erwartet werden konnte.
Jan. 1996	<u><i>1. HTTP/1.1-Internet-Entwurf:</i></u> Dieser Entwurf führte grundlegende Änderungen zu HTTP/1.0 ein, wie z.B. unbedingte Angabe des angefragten Hosts und Mehrfachnutzung einer bestehenden Verbindung als Standard-einstellung.
Jan. 1997	<u><i>HTTP/1.1 als RFC2616:</i></u> Die neueste praxisorientierte Version des HTTP-Protokolls. Es implementiert alle Neuerungen die bereits im "1. HTTP/1.1 Internet-Draft" beschlossen wurden. Es wird durch die RFC2616 (http://ftp.ics.uci.edu/pub/ietf/http/rfc2616.txt) beschrieben.

Kapitel 3.2: Die Anfrage per HTTP:

Beim Aufruf einer Web-Seite (z.B.: „<http://www.google.de>“) im Browser stellt dieser per HTTP eine Anfrage an den entsprechenden Web-Server. Hierzu wird zunächst die Adresse des Web-Servers per DNS in eine IP-Adresse aufgelöst (z.B.: '216.239.39.147').

Danach baut der Browser mit Hilfe des Betriebssystems eine TCP/IP-Verbindung zu der ermittelten IP-Adresse an Port 80 auf. Gelingt dies, so besteht eine bi-direktionale TCP/IP-Verbindung zwischen Server und Browser, welche als Kommunikationsmedium für das HTTP-Protokoll dient. HTTP realisiert die Übertragung der Daten als unverschlüsselten Klartext.

Die Anfrage an den Server muss nach folgenden Konventionen aufgebaut sein:

GET <STRING> HTTP/<VERSION>

HOST: <ADRESSE>

Hierbei steht <STRING> für eine Zeichenkette die vom Web-Server ausgewertet wird (z.B.: '/', <VERSION> für die bei der Kommunikation via HTTP einzusetzende Version des HTTP-Protokolls und <ADRESSE> für die Adresse des Web-Servers (z.B.: „www.google.de“).

Ist die Auswertung durch den Web-Server erfolgt (hierauf wird in Kapitel 4 näher eingegangen) wird das Ergebnis der Auswertung an den Browser übertragen. Die Auswertung kann die angeforderte Web-Site oder aufgetretene Fehlermeldungen enthalten.

Seit HTTP Version 1.1 ist es möglich eine Verbindung zur Übertragung mehrerer Dokumente zu nutzen. Hierzu wird nach der Übertragung der angeforderten Datei vom Browser die nächste Anfrage gestartet, und über die noch bestehende Verbindung geschickt. Dies kann beliebig oft wiederholt werden.

Wird die Verbindung nicht mehr weiter benötigt, wird sie automatisch durch das Betriebssystem wieder geschlossen.

Kapitel 3.3: HTTPS-Verbindungen:

Da HTTP Daten in Klartext überträgt ist aus Sicherheitsgründen von der Übertragung sensibler Daten, wie z.B. Passwörter, dringend abzuraten.

Um dennoch sensible Daten per HTTP übertragen zu können, wurde das HTTP-Protokoll um

die Möglichkeit erweitert, die zu versendenden Daten zu verschlüsseln. Diese Erweiterung wird auch mit dem HTTPS-Protokoll bezeichnet.

Hierzu handeln der Web-Server und der Browser nach Aufbau der TCP/IP-Verbindung eine gemeinsame Verschlüsselungsmethode aus. Die Daten werden im gleichen Format wie es bei HTTP standardisiert ist übertragen, jedoch findet vor dem Versand der Daten die Verschlüsselung mit der jeweils ausgehandelten Methode statt. Nun können die sensiblen Daten gefahrlos transferiert werden.

Um HTTPS- und HTTP-Verbindungen besser voneinander trennen zu können, nimmt ein Web-Server HTTPS-Verbindungen standard gemäß nur am Port 443 entgegen, HTTP-Verbindungen werden dagegen standard gemäß nur am Port 80 akzeptiert.

Kapitel 4: Die Verarbeitung im Web-Server Apache

Hat Apache die Anfrage empfangen, so wird `<STRING>` genutzt um dem Browser die richtige Web-Seite zurück zu liefern.

Hierzu wird zunächst bestimmt ob es sich bei `<STRING>` um eine statische Datei handelt, oder ob die Ausführung eines Skriptes nötig ist, um die angeforderte Datei dynamisch zu erzeugen, z.B. PHP- oder CGI-Skripte.

Ist die Ausführung eines Skriptes nicht nötig, so wird versucht, die Datei auf dem Plattenspeicher des Web-Servers zu lokalisieren. Schlägt dieser Vorgang fehl, so sendet der Web-Server als Antwort die Fehlermeldung „**404 Not found**“. Andernfalls untersucht Apache ob als `<STRING>` nur der Name eines Verzeichnisses auf dem Plattenspeicher übergeben wurde, falls ja wird in dem Verzeichnis standardgemäß nach der Datei „**index.html**“ gesucht. Vor dem Versand des Dokumentes werden die Sicherheitseinstellungen für dieses überprüft. Somit wird gewährleistet, daß kein unautorisierter Zugriff auf Dateien stattfinden kann. Hat der Browser keine Zugriffsrechte auf die angeforderte Datei, sendet der Web-Server die Fehlermeldung „**401 Unauthorized**“ an den Browser zurück, ansonsten wird die angeforderte Datei an den Browser übertragen.

Die Abfrage von Authentifizierungsinformationen, die einem Benutzer den Zugriff auf ein Dokument gewähren, obliegt dem Browser. Die vom Benutzer eingegebenen Informationen

werden bei der Anfrage des Dokuments mit an den Server übertragen.

Hat der Web-Server ermittelt dass ein Skript angefordert wurde, so wird dessen Ausführung angestoßen. Die Ausgaben des Skriptes werden an den Browser zurückgeschickt.

Kapitel 5: Was ist HTML?

Das Akronym HTML steht für **H**yper**T**ext **M**arkup **L**anguage. Die Sprache HTML dient zur Gestaltung und zum Aufbau von Web-Seiten.

Der Unterschied von HTML zu normalem Text ist die Möglichkeit, einzelne Texte durch so genannte „Hyperlinks“ miteinander zu verknüpfen und zu strukturieren (daher die Bezeichnung „HyperText“). Zur Gestaltung und Strukturierung einer Seite werden TAGs („Marken“) verwendet, welche eine vordefinierte Funktion besitzen. Daraus ergibt sich eine Begrenzung der Formatierungsmöglichkeiten, da HTML die Definition eigener TAGs nicht vorsieht.

Eine weitere Beschränkung von HTML liegt in der nicht vorhandenen Trennung zwischen Inhalt eines Dokumentes und dessen Layout. Daher eignet sich HTML lediglich zum Transfer von Dokumenten zwischen dem Betrachter (Browser) und dem Web-Server, nicht aber zur getrennten Verwaltung von Content und Design/ Layout.

Die wichtigsten HTML-TAGs:

<code><HTML></code>	<code></HTML></code>	Markieren Anfang und Ende eines HTML Dokuments
<code><HEAD></code>	<code></HEAD></code>	Markieren Anfang und Ende des Kopfes, welcher Meta-Daten über das darzustellende Dokument enthalten kann
<code><BODY></code>	<code></BODY></code>	Markieren Anfang und Ende des sichtbaren Dokumentbereichs
<code></code>	<code>TEXT </code>	Erzeugt Hyperlink zu <i>LINK</i> und stellt diesen als <i>TEXT</i> dar
<code></code>		Fügt <i>BILD</i> in das Dokument ein

Kapitel 6: Fachbegriffe

DNS	D omain N ame S ystem – System das Domain- und Hostnamen in IP-Adressen auflöst
Domain	Eindeutiger Name, innerhalb einer DNS-Hierachiestufe (z.B. „google.de“)
Host	Name unter dem ein Server verfügbar ist
IP-Adresse	Eindeutige numerische Adresse, mit deren Hilfe TCP/IP die Kommunikation zwischen Endknoten koordiniert
TCP/IP	T ransmission C ontrol P rotocol/ I nternet P rotocol – Protokolle die Verbindungen zwischen Knoten des Internets erlauben
Port	Numerischer Wert, mit dessen Hilfe verschiedene Protokolle Verbindungen zu IP-Adressen aufbauen

Kapitel 7: Quellenangaben

- HTML 4.01 Spezifikation
<http://www.w3c.org/TR/html4/>

- RFC 1945 - HTTP/1.0
<http://www.w3c.org/Protocols/rfc1945/rfc1945>

- RFC 2616 - HTTP/1.1
<http://www.w3c.org/Protocols/rfc2616/rfc2616.html>

- Apache Dokumentation
<http://httpd.apache.org/docs-2.0>