1 Hyper Text Transfer Protocol

- Aktuelle Version http/1.1
- Übermittlung der Daten erfolgt nach Request-Reponse-Schema
- http-Client sendet seine Anfrage an den http-Server, der diese bearbeitet und eine Antwort zurücksendet
- http ist Abwärtskompatibel (höhere Version muss sich an ältere anpassen)

1.1 Ablauf einer http-Verbindung

- Kommunikation zwischen Client und Server durch http-Nachrichten
- Client stellt TCP-Verbindung auf Port 80 zum Webserver her
- Client schickt Anfrage (*Request*) an den Server indem er die Informationen spezifiziert, die er einsehen möchte
- Request enthält u.a. die Art der Anfrage (GET, POST, HEAD, PUT, DEL, TRACE; je nach Protokollversion), den URL (*Universal Ressource Locater*) und die Protokollversion (http/1.0 oder http/1.1)
- Webserver registriert den Request und durchsucht sein Dateisystem nach der geforderten Datei
- Server schickt Antwort (*Response*), welche die gewünschten Informationen und den MIME-Type der angeforderten Datei enthält, sofern die Aktion erfolgreich war
- War die Aktion nicht erfolgreich endet der Server eine Fehlermeldung
- MIME-Type sagt dem Client was er mit den empfangenen Daten anfangen soll (z.B. Anzeige im Browser)
- Verbindung wird direkt nach dem senden der *Response* beendet ➔ http = zustandsloses Protokoll (Gegenteil = zustandsorientiertes Protokoll = FTP = unterhält Verbindung bis Client sie ausdrücklich beendet)

1.2 http-Requests

- bestimmt durch Angabe von:
  - Methode
  - URL
  - Request-Header-Felder
- METHOD URL HTTP/version
  General Header
  Request Headers
  Entity Header (optional)
  _____Leerzeile_____
  Request Entity (falls vorhanden)

1.3 http-Methoden

- GET:
  - Wichtigste Methode
  - Dient zur Anforderung eines Dokuments oder einer anderen Quelle
  - Quelle wird durch Request-URL identifiziert
  - Zwei Typen:
    1. conditional GET ➔ Anforderung von Daten ist an Bedingungen (If-Modified-Since, If Unmodified-Since, If-Match) geknüpft, welche im Header-Feld „Conditional" hinterlegt sind; Netzbelastung lässt sich so deutlich verringern (nur die wirklich benötigten Daten werden übertragen); Praxis: Proxyserver nutzen diese Funktion um zu verhindern dass Daten die bereits im Cache liegen mehrfach übertragen werden
    2. partial GET ➔ verwendet das Range-Header-Feld, welches nur Teile der Daten

überträgt, die der Client aber noch verarbeiten kann; wird für Wiederaufnahme eines unterbrochenen Datentransfer verwendet
- POST
- DELETE

1.4 http-Response

- Aufbau ähnlich zum Request:
  HTTP/version Status-Code Reason-Zeile
  General Header
  Response Header
  Entity Header (optional)
  _____Leerzeile_____
  Resource Entity (falls vorhanden)
- Server übermittelt zunächst die http-Version der Nachricht
- Zweiter Eintrag = Statusmeldung
- Content Type und Content Length wichtig für weitere Bearbeitung durch den Client
- Content Type beschreibt den MIME-Typ der im Datenbereich übermittelten Datei
- Content Length = Länge der Daten in Byte (Einsatz des Feldes nicht zwingend vorgeschrieben

!! MIME-Typ wird vom Server in Response mitgeschickt damit Client weiß was er machen soll!!

1.5 Response Codes

- Dreistelliger Integerwert
- Liefert dem Client wichtige Infos zu Verfügbarkeit, erfolgreiche Bearbeitung und Fehlermeldungen
- Fünf Kategorien:
  - 1XX:
    Informelle Meldungen wie Request erhalten, Bearbeitung wird durchgeführt
  - 2XX:
    Erfolg: Request wurde erfolgreich erhalten, verstanden und angenommen
    (Beispiel: 200 = OK)
  - 3XX:
    Weiterleiten: Weitere Aktionen müssen eingeleitet werden, damit eine Request vollständig bearbeitet werden kann
  - 4XX:
    Clientfehler: Request enthält ungültige Syntax oder kann nicht bearbeitet werden
    (Beispiel: 404 = File not Found oder 403 = Forbidden (Zugriff verweigert))
  - 5XX:
    Serverfehler: Der Server kann eine gültige Request nicht bearbeiten

2 Konfigurationsdateien

- httpd ➔ d = dämon
- Serverkonfigurationen im Verzeichnis */etc/httpd/conf* in der Datei *httpd.conf*

3 Modulkonfiguration

3.1 Dynamic Shared Objects

- Apache Webserver kann monolithisch oder modular aufgebaut sein

- Mittlerweile ist er so umfangreich dass er nicht mehr monolithisch sondern modular aufgebaut ist
- Der Kern-Server kennt nur die grundlegenden Eigenschaften eines Webservers
- Zusätzliche Fähigkeiten (z.B. Authentifizierung, CGI-Scripts, PHP-Scripts, Session Management,…) werden mit Hilfe von Modulen eingebaut
- Durch die DSO-Funktionalität ist es möglich Module dynamisch in den Apache einzubinden oder zu entfernen
- Apache muss nicht neu übersetzt werden aber das Modul *mod_so* (stellt DSO zur Verfügung) muss statisch im Apache eingebunden sein
- Module müssen als sog. DSO-Dateien vorliegen
- Module können zur Laufzeit des Apache verändert werden ohne dass aktive Verbindungen unterbrochen werden müssen

4 Konfigurationsanweisungen

- Konfigurationsdatei *httpd.conf* in drei wichtige Bereiche aufgeteilt:
  - Globale Einstellungen
  - Einstellungen der Hauptserver
  - Einstellungen der virtuellen Server

4.1 Globale Einstellungen (Section1: Global Environment)

- Bedingungen unter denen der Apache Server arbeiten soll
- Dazu gehören:
  - Angaben zum Servertyp:
    httpd Dämon kann durch *inetd* gestartet werden oder als *standalone* (üblicherweise) laufen (bei Windows nur standalone)
    Nachteil inetd: Apache muss nach jeder Anfrage neu gestartet werden
  - Definition des Hauptverzeichnisses:
    Hier befinden sich die Verzeichnisse für die Logfiles, Konfigurations- und Protokolldateien
  - Festlegungen für Zugriffszeiten und Zahlen
  - Liste der zur Laufzeit einzubindenden Module:
    im Standardfall erst einmal deaktiviert, Einbindung individuell erstellter Module möglich

4.2 Einstellungen des Hauptservers (Section2: Main Server Configuration)

- Anweisungen zur Arbeitsweise
  - Festlegung des vom Server beanspruchten Ports (für http-Protokoll i.d.R. Port 80)
  - Administratoradresse:
    Adresse an die eventuelle Probleme mit dem Server gemeldet werden können; Sie erscheint auf allen Dokumenten die vom Server generiert werden (z.B. Fehlermeldungen)
  - Servername:
    um Hostnamen festzulegen über den der Webserver angesprochen werden soll;
    es muss ein gültiger DNS-Name sein; verfügt der Host nicht über einen DNS-Namen kann auch eine IP-Adresse angegeben werden
  - Identität des Webusers:
    Benutzer und Gruppe unter dem der Apache seine Requests bearbeiten soll
  - Pfadangaben zu Dokument-, Benutzer- und Scriptverzeichnissen:
    - Dokumentverzeichnis:
      enthält HTML-Dokumente und sonstige Dateien die über den Webserver abrufbar sein sollen

- Benutzerverzeichnis:
  Beispiel: http://www.irgendwas.com/~user/
  Name nach der Tilde ist der Accountname des Users; Unterverzeichnis im Home-Verzeichnis eines Benutzers
- Verzeichnisindex:
  Name der Datei die in einem Verzeichnis aufgerufen werden soll wenn nur der Verzeichnisname angegeben wurde (mehrere Namen durch Leerzeichen getrennt)
- Formatierung der Protokolldateien
- Aufgabenzuweisung an die geladenen Module

4.3 Virtuelle Server (Section3: Virtual Hosts)

- Apache = erster Webserver der die Möglichkeit von virtuellen Webservern gewährleistete
- Server kann gleichzeitig unter mehr als einem Hostnamen agieren und dabei jeweils ein eigenes Dokumentverzeichnis, LogFiles,… verwalten

4.3.1 IP-basierte virtuelle Hosts

- Ursprüngliche Form virtueller Server
- Setzt voraus dass jeder virtuelle Server eine eigene IP-Adresse bekommt
- Beruht auf einer Restriktion von http/1.0, da hier ein Client nicht kenntlich machen kann auf welche Website er zugreifen will
- Verwendet man jedoch für jeden virtuellen Server eine eigene IP-Adresse erkennt der Webserver anhand dieser welches Dokument er zurückliefern soll

4.3.2 Namensbasierte virtuelle Hosts

- Eingeführt mit http/1.1
- Virtuelle Server ohne jeweils eine eigene IP-Adresse vergeben zu müssen
- Lediglich ein Hostname-Alias im Nameserver eintragen

4.3.2.1 <VirtualHost>

- nur in Serverkonfiguration erlaubt
- Einrichtung oder Deaktivierung eines virtuellen Servers

4.3.2.2 NameVirtualHost

- Muss verwendet werden um überhaupt namensbasierte Server verwenden zu können
- Eingeführt mit Apache 1.3
- Einfachere / eindeutigere Konfiguration der virtuellen Server

5 .htaccess – Server-Reaktionen kontrollieren

5.1 Allgemeines

- Außer Konfigurationsdateien auch die Möglichkeit von Verzeichniskonfigurationsdateien
- .htaccess = Konigurationsdateien für Verzeichnisse die zum Webangebot gehören
- der übliche Weg um nur bestimmten Benutzern Zugriff auf bestimmte Daten zu erteilen
- „richtiger" Passwortschutz und einiges mehr
- Möglichkeit ganze Benutzerkreise automatisch auszusperren oder alle bis auf bestimmte auszusperren
- Einstellen von Optionen zum Verzeichnisbrowsing
- Einstellen von automatischen Weiterleitungen

- Schaffung eigener Regelungen für den Fall von http-Fehlermeldungen
- Angebot alternativer Inhalte abhängig von bestimmten Bedingungen
- Einstellung dass Daten komprimiert an den aufrufenden Browser übertragen werden

5.2 Verzeichnisse und Dateien mit Passwort schützen

- .htaccess-Dateien sind verzeichnis-spezifisch, d.h. die .htaccess-Datei wird in dem Verzeichnis gespeichert in dem die geschützten Daten liegen
- geschützt werden kann das ganze Verzeichnis mit all seinen Dateien oder nur bestimmte Dateien oder Dateitypen
- Passwortschutz wahlweise für Benutzer oder ganze Benutzergruppen oder kombiniert
- .htaccess alleine genügt jedoch nicht, zusätzlich braucht man eine Datei in der Benutzer und Passwörter stehen; wird mit Benutzergruppen gearbeitet braucht man zusätzlich noch eine Datei in der die Benutzergruppen definiert sind
- Schlüsselwörter zum Passwortschutz:
  - AuthType:
    Art der Authentifizierung; Standard ist Basic ➔ Benutzer und Passwörter stehen in einer noch anzugebenden Datei
  - AuthName:
    Mitteilung an den Benutzer für welchen Bereich er sich mit UserName und Passwort authentifizieren soll; beliebe Zeichenkette, wenn Leerzeichen muss sie in Anführungszeichen stehen
  - AuthUserFile:
    Angabe der Datei in der die autorisierten Benutzer und ihre Passwörter stehen; der vollständige absolute Pfadname muss angegeben werden (ab dem Wurzelverzeichnis des Rechners)
  - AuthGroupFile:
    wenn Benutzergruppen verwendet werden sollen
  - Require:
    zweites Schlüsselwort entweder *user* oder *group* für Benutzer bzw. Benutzergruppe; valid-user wenn alle in der Datei aufgeführten User Zugriff erhalten sollen

5.2.1 Htpasswd

- Passwortdatei kann mit Hilfe des Apache Utility htpasswd erzeugt werden

5.2.2 Die Gruppendatei

- Einträge bei denen zunächst ein Gruppenname notiert wird und dahinter nach einem Doppelpunkt die Namen der Benutzer die in die Gruppe gehören (für die Benutzernamen muss es einen Eintrag in der Benutzerdatei geben)

5.3 Schutz von Dateien, Dateitypen oder Zugriffsmethoden

- Zugangsschutz gilt für das Verzeichnis in dem die .htaccess-Datei liegt und für alle Verzeichnisse unterhalb davon
- Schutz kann aber auch eingeschränkt werden auf Dateien, Dateitypen oder Zugriffsmethoden
- Um Schutz einzuschränken benutzt man Tags (wie in HTML oder XML)

5.4 IPs, IP-Bereiche oder Namensadressen zulassen / ausschließen

- IP-Adressen bzw. IP-Bereiche können davon ausgeschlossen werden , auf Webseiten zuzugreifen
- Es können alle IP-Adressen ausgeschlossen werden und nur ganz bestimmten Zugriff erlaubt werden
- Greift ein nichtauthorisierter Anwender auf so eine Seite zu bekommt er die Fehlermeldung 403 und der Zugriff wird verweigert

6 Secure Webserver

- Notwendig um eine sichere Kommunikation zu ermöglichen

6.1 Das eigentliche Problem

- A und B sitzen in einer geschützten Umgebung
- Kommunikationskanal von A nach B ist öffentlich und kann von jedem abgehört werden
- A und B müssen damit rechnen dass C sie belauscht
- C kann auch beliebige Nachrichten aus diesem Kanal nehmen und diese verändert weitersenden ohne dass der Empfänger dies bemerkt

6.1.1 Die Lösung

- Beispiel:
  - B besitzt einen Ausweis mit dem er sich A gegenüber eindeutig ausweisen kann
  - Ausweis muss so gestaltet sein dass sich kein anderer als B mit diesem ausweisen kann
  - Theoretisch kann C jeden Ausweis fälschen
  - Man kann es ihm so erschweren dass die Kosten dafür höher werden würden als der Gewinn

6.2 Verschlüsselungsverfahren allgemein

6.2.1 Symmetrische Verschlüsselung

- Gleicher Schlüssel zum ver- und entschlüsseln der Nachricht
- Schlüssel muss allen Beteiligten bekannt sein
- Deshalb muss er zuerst den Kommunikationspartnern bekannt gemacht werden
- Begegnen sich Kommunikationspartner nicht persönlich ➔ Verfahren ungeeignet, weil der Schlüssel vor Einsatz der geschützten Kommunikation ausgetauscht werden muss und so ungeschützt über das Internet übertragen werden müsste
- Vorteil: Geschwindigkeit der Datenver- und Entschlüsselung
- Symmetrische Schlüssel können nur dann verwendet werden wenn der Schlüssel nicht übermittelt werden muss
- Möglichkeit: Übermittlung des symmetrischen Schlüssel über eine Asymetrische Verschlüsselung

6.2.2 Asymmetrische Verschlüsselung

- Sender und Empfänger verwenden unterschiedliche Schlüssel zur Ver- bzw. Entschlüsselung, sog. Schlüsselpaare

- Schlüsselpaare bestehen aus einem privaten Schlüssel (private Key) und einem öffentlichen Schlüssel (public key)
- Private key ist nur dem Sender bekannt
- Public key ist allgemein zugänglich
- Daten die mit einem privaten Schlüssel verschlüsselt wurden können nur mit einem öffentlichen Schlüssel entschlüsselt werden und umgekehrt
- Empfänger der Daten erzeugt einen privaten und einen öffentlichen Schlüssel, welcher an die Gegenstelle übermittelt wird
- Diese verwendet den öffentlichen Schlüssel des Empfängers um die Daten zu verschlüsseln
- Entschlüsselt können die Daten nur vom Besitzer des dazu passenden privaten Schlüssel werden

6.3 Secure Socket Layer

- SSL-Protokoll verwendet sowohl symmetrische als auch asymmetrische Verschlüsselung
- Asymmetrische Verschlüsselung nur zu Beginn zum Verbindungsaufbau und zur Vereinbahrung des symmetrischen Schlüssels verwendet
- SSL = Protokoll für die sichere verschlüsselte Übertragung von Nachrichten im Internet
- Es lassen sich beliebige Dienste wie SMTP, POP3, FTP,… verschlüsseln

6.4 Verbindung über SSL

- Angabe des Schemas https im URL
- https-Verbindungen laufen über TCP, Port = 443
- für die verschlüsselte Verbindung wird ein geheimer Schlüssel gebraucht; symmetrische Verschlüsselung, d.h. beide Seiten müssen den Schlüssel kennen ; sonst niemand
- vor Aufbau einer SSL-Verbindung wird eine Initialisierung durch das Handshakeprotokoll ausgeführt ➔ legt die Sicherheitsstufe fest auf die sich Client und Server einigen; übernimmt die notwendigen Echtheitsbestätigungen für die Verbindung und handelt einen Sitzungsschlüssel für die Verschlüsselung aus
- Ablauf:
  - Anfrage Client an Server, mit den Verschlüsselungsverfahren die er unterstützt
  - Server wählt Verfahren aus und übermittelt dem Client ein Zertifikat mit dem öffentlichen Schlüssel des Servers
  - Client generiert sog. Sitzungsschlüssel (Session Key) für Symmetrischen Verschlüsselungsverfahren. Session Key wird mit öffentlichen Schlüssel des Server verschlüsselt und zum Server übertragen, nur dieser kann ihn mit dem privaten Schlüssel wieder entschlüsseln
  - Optional: Clientauthentifizierung ähnlich der Serveridentifikation; Client muss jedoch über ein gültiges Zertifikat verfügen
- Schlüssel ist ausgetauscht und Verbindung kann aufgebaut werden
- Während Verbindung besteht übernimmt SSL lediglich die symmetrische Ver- und Entschlüsselung des Datenstroms mithilfe des Session Key

6.5 Host und Client-Authentisierung

- HostAuthentisierung:
  nur Host muss seine Identität nachweisen durch den Besitz des rechtmäßigen privaten Schlüssels
- ClientAuthentisierung:
  Benutzer muss sich ausweisen; Server muss sich auf sichere Weise den öffentlichen Schlüssel des Benutzers holen, Benutzer kann nicht einfach selbst gefragt werden ➔ nicht sicher
- Öffentlicher Schlüssel einer Person muss mit ihrer Identität offiziell und nachprüfbar verknüpft sein

6.6 X.509 Zertifikate

- Beinhaltet Namen und die digitale Signatur des Ausstellers, sowie Informationen über die Identität des Inhabers
- Zertifikat ausgestellt von einer Zertifizierungsstelle die allgemein anerkannt ist
- Sie verbürgt sich dafür dass die Zuordnung eines Public Key zu einer bestimmten Institution korrekt ist und dass die Institution die den passenden privaten Schlüssel besitzt, tatsächlich existiert

6.7 Erstellen von Zertifikaten

- Drei Schritte:
  - Schlüsselpaar (öffentlicher und privater Schlüssel) muss erstellt werden
  - Senden des öffentlichen Schlüssels zusammen mit anderen Informationen als CSR (Certificate Signing Request) an die Zertifizierungsinstanz CA (Certificate Authority)
  - Richtigkeit der Angaben wird überprüft und CSR mit Signatur der CA bestätigt und als fertiges Zertifikat zum Antragsteller zurückgeschickt

7 Common Gateway Interface

- Schnittstelle des Webservers
- Erlaubt es Anfragen eines Webbrowsers an Programme (auf dem Webserver) weiterzureichen und von diesen ausführen zu lassen
- CGI-Schnittstelle ruft nicht nur das jeweils auszuführende Programm auf und leitet dessen Antwort weiter , sondern stellt auch eine Reihe von Daten bereit die der Webserver speichert und die ein CGI-Script auslesen kann um Daten verarbeiten zu können
- Diese Daten werden vom Webserver in sog. CGI-Umgebungsvariablen gespeichert

8 Logdateien

- Logfiles zum Überblick über Funktion und Auslastung des Servers zu bekommen
- ErrorLog registrieren von Fehlern die bei der Beantwortung oder Bearbeitung auftreten
- AccessLog jeder an ihn gerichteten Request wird hier vom Webserver protokolliert

8.1 ErrorLog

- Konfigurationsanweisung in der Konfigurationsdatei in der dem Apache mitgeteilt wird in welche Datei er seine Fehlermeldungen schreiben soll