

# Webdienste

Autor: Tobias Schmid

Datum: 08.06.2015

Typ: Information

Version: 1.0










## Inhaltsverzeichnis

INHALT	
1	Einleitung.....3
2	Protokolle.....3
2.1	Hypertext Transfer Protocol (http) .....3
2.1.1	Eigenschaften .....3
2.1.2	Aufbau.....4
2.1.3	Funktionsweise.....4
2.2	Hypertext Transfer Protocol Secure .....5
2.2.1	Nutzen.....5
2.2.2	Technik .....5
2.2.3	Client-Verarbeitung.....6
2.2.4	Varianten der HTTPS-Anwahl.....6
2.2.5	Vorinstallierte Zertifikate.....6
2.3	File Transfer Protocol.....7
2.3.1	Verbindungsarten.....7
2.3.2	Aktives FTP .....7
2.3.3	Passives FTP .....7
2.3.4	Öffentliche FTP-Server .....8
2.3.5	FTP-Software .....8
2.3.6	Sicherheit.....8
2.4	Simple Mail Transfer Protocol.....9
2.4.1	Geschichte.....9
2.4.2	Verfahren .....10
2.5	Secure Shell .....10
2.5.1	Geschichte.....10
2.5.2	Verwendung.....11
3	Webserver.....12
3.1	Apache HTTP Server .....12
3.2	Internet Information Services (IIS) .....13
3.2.1	Betriebssysteme.....13
3.2.2	Versionen .....13
3.3	XAMPP .....14
3.3.1	Eigenschaften und Funktionen.....14
3.3.2	Varianten .....14
3.3.3	Lizenz.....15
3.4	Microsoft WebMatrix .....15
3.4.1	Features.....16

## 1 Einleitung

## 2 Protokolle

### 2.1 Hypertext Transfer Protocol (http)

Hypertext Transfer Protocol	
	
Familie:	Internetprotokollfamilie
Einsatzfeld:	Datenübertragung (Hypertext u. a.) auf Anwendungsschicht
aufbauend auf	TCP (Transport)
Einführung:	1991
aktuelle Version:	2.0 (2015)
Standard:	<a href="#">RFC 1945</a>  (HTTP/1.0, 1996) <a href="#">RFC 2616</a>  (HTTP/1.1, 1999) <a href="#">RFC 7230</a>  : Message Syntax and Routing (1.1, 2014) <a href="#">RFC 7231</a>  : Semantics and Content (1.1, 2014) <a href="#">RFC 7232</a>  : Conditional Requests (1.1, 2014) <a href="#">RFC 7233</a>  : Range Requests (1.1, 2014) <a href="#">RFC 7234</a>  : Caching (1.1, 2014) <a href="#">RFC 7235</a>  : Authentication (1.1, 2014)

Das Hypertext Transfer Protocol (HTTP, englisch für Hypertext-Übertragungsprotokoll) ist ein zustandsloses Protokoll zur Übertragung von Daten auf der Anwendungsschicht über ein Rechnernetz. Es wird hauptsächlich eingesetzt, um Webseiten (Hypertext-Dokumente) aus dem World Wide Web (WWW) in einen Webbrowser zu laden. Es ist jedoch nicht prinzipiell darauf beschränkt und auch als allgemeines Dateiübertragungsprotokoll sehr verbreitet.

HTTP wurde von der Internet Engineering Task Force (IETF) und dem World Wide Web Consortium (W3C) standardisiert. Aktuelle Version ist HTTP/2, welche als RFC 7540 am 15. Mai 2015 veröffentlicht wurde. Die Weiterentwicklung wird von der HTTP-Arbeitsgruppe der IETF (HTTPbis) organisiert. Es gibt zu HTTP ergänzende – wie HTTPS für die Verschlüsselung übertragener Inhalte – und darauf aufbauende Standards wie das Übertragungsprotokoll WebDAV.

#### 2.1.1 Eigenschaften

Nach etablierten Schichtenmodellen zur Einordnung von Netzwerkprotokollen nach ihren grundlegenden oder abstrakteren Aufgaben wird HTTP der sogenannten Anwendungsschicht zugeordnet. Diese wird von den Anwendungsprogrammen angesprochen, im Fall von HTTP ist das meist ein Webbrowser. Im ISO/OSI-Schichtenmodell entspricht die Anwendungsschicht den Schichten 5 bis 7.

HTTP ist ein zustandsloses Protokoll. Informationen aus früheren Anforderungen gehen verloren. Ein zuverlässiges Mitführen von Sitzungsdaten kann erst auf der Anwendungsschicht durch eine Sitzung über einen Sitzungsbezeichner implementiert werden. Über Cookies in den Header-Informationen können aber Anwendungen

realisiert werden, die Statusinformationen (Benutzereinträge, Warenkörbe) zuordnen können. Dadurch werden Anwendungen möglich, die Status- beziehungsweise Sitzungseigenschaften erfordern. Auch eine Benutzerauthentifizierung ist möglich. Normalerweise kann die Information, die über HTTP übertragen wird, auf allen Rechnern und Routern gelesen werden, die im Netzwerk durchlaufen werden. Über HTTPS kann die Übertragung aber verschlüsselt erfolgen.

Durch Erweiterung seiner Anfragemethoden, Header-Informationen und Statuscodes ist HTTP nicht auf Hypertext beschränkt, sondern wird zunehmend zum Austausch beliebiger Daten verwendet, außerdem ist es Grundlage des auf Dateiübertragung spezialisierten Protokolls WebDAV. Zur Kommunikation ist HTTP auf ein zuverlässiges Transportprotokoll angewiesen, wofür in nahezu allen Fällen TCP verwendet wird.

Derzeit werden zwei Protokollversionen, HTTP/1.0 und HTTP/1.1, verwendet. Neuere Versionen wichtiger Webbrowser wie Chromium, Opera, Firefox und Internet Explorer sind darüber hinaus bereits kompatibel zu SPDY, der Entwicklungsvorlage für Version 2 des HTTP (HTTP/2).

### 2.1.2 Aufbau

Die Kommunikationseinheiten in HTTP zwischen Client und Server werden als Nachrichten bezeichnet, von denen es zwei unterschiedliche Arten gibt: die Anfrage (englisch Request) vom Client an den Server und die Antwort (englisch Response) als Reaktion darauf vom Server zum Client.

Jede Nachricht besteht dabei aus zwei Teilen, dem Nachrichtenkopf (englisch Message Header, kurz: Header oder auch HTTP-Header genannt) und dem Nachrichtenrumpf (englisch Message Body, kurz: Body). Der Nachrichtenkopf enthält Informationen über den Nachrichtenrumpf wie etwa verwendete Kodierungen oder den Inhaltstyp, damit dieser vom Empfänger korrekt interpretiert werden kann (→ Hauptartikel: Liste der HTTP-Headerfelder). Der Nachrichtenrumpf enthält schließlich die Nutzdaten.

### 2.1.3 Funktionsweise

Beispiel einer Transaktion, ausgeführt mit Telnet

Wenn auf einer Webseite der Link zur URL `http://www.example.net/infotext.html` aktiviert wird, so wird an den Rechner mit dem Hostnamen `www.example.net` die Anfrage gerichtet, die Ressource `/infotext.html` zurückzusenden.

Der Name `www.example.net` wird dabei zuerst über das DNS-Protokoll in eine IP-Adresse umgesetzt. Zur Übertragung wird über TCP auf den Standard-Port 80 des HTTP-Servers eine HTTP-GET-Anforderung gesendet.

Anfrage:

```
GET /infotext.html HTTP/1.1
Host: www.example.net
```

Enthält der Link Zeichen, die in der Anfrage nicht erlaubt sind, werden diese %-kodiert. Zusätzliche Informationen wie Angaben über den Browser, zur gewünschten Sprache etc. können über den Header (Kopfzeilen) in jeder HTTP-Kommunikation übertragen werden. Mit dem „Host“-Feld lassen sich verschiedene DNS-Namen unter der gleichen IP-Adresse unterscheiden. Unter HTTP/1.0 ist es optional, unter HTTP/1.1 jedoch erforderlich. Sobald der Header mit einer Leerzeile (beziehungsweise zwei aufeinanderfolgenden Zeilenenden) abgeschlossen wird, sendet der Rechner, der einen Web-Server (an Port 80) betreibt, seinerseits eine HTTP-Antwort zurück. Diese besteht aus den Header-Informationen des Servers, einer Leerzeile und dem tatsächlichen Inhalt der Nachricht, also dem Dateinhalt der `infotext.html`-Datei. Übertragen werden normalerweise Dateien in Seitenbeschreibungssprachen wie (X)HTML und alle ihre Ergänzungen, zum Beispiel Bilder, Stylesheets (CSS), Skripte (JavaScript) usw., die meistens von einem Browser in einer lesbaren Darstellung miteinander verbunden werden. Prinzipiell kann jede Datei in jedem beliebigen Format übertragen werden, wobei die „Datei“ auch dynamisch generiert werden kann und nicht auf dem Server als physische Datei vorhanden zu sein braucht (zum Beispiel bei Anwendung von CGI, SSI, JSP, PHP oder ASP.NET). Jede Zeile im Header wird durch den Zeilenumbruch `<CR><LF>` abgeschlossen. Die Leerzeile nach dem Header darf nur aus `<CR><LF>`, ohne eingeschlossenes Leerzeichen, bestehen.

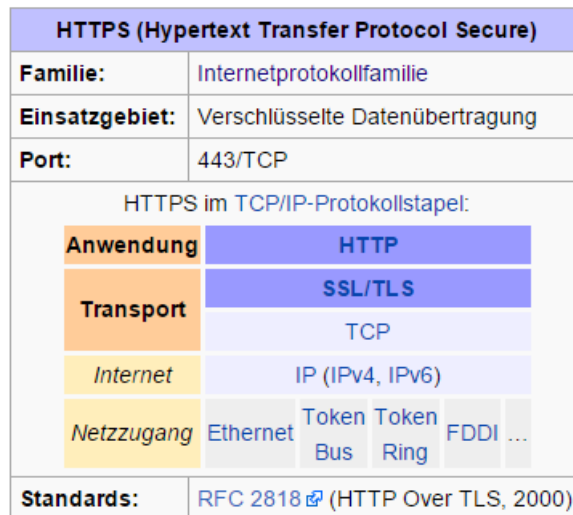
Antwort:

```
HTTP/1.1 200 OK
Server: Apache/1.3.29 (Unix) PHP/4.3.4
Content-Length: 123456 (Größe von infotext.html in Byte)
```

Content-Language: de (nach RFC 3282 sowie RFC 1766)  
Connection: close  
Content-Type: text/html

Der Server sendet eine Fehlermeldung sowie einen Fehlercode zurück, wenn die Information aus irgendeinem Grund nicht gesendet werden kann, allerdings werden auch dann Statuscodes verwendet, wenn die Anfrage erfolgreich war, in dem Falle (meistens) 200 OK. Der genaue Ablauf dieses Vorgangs (Anfrage und Antwort) ist in der HTTP-Spezifikation festgelegt.

## 2.2 Hypertext Transfer Protocol Secure



HyperText Transfer Protocol Secure (HTTPS, englisch für sicheres Hypertext-Übertragungsprotokoll) ist ein Kommunikationsprotokoll im World Wide Web, um Daten abhörsicher zu übertragen.

Technisch definiert es als URI-Schema eine zusätzliche Schicht zwischen HTTP und TCP. HTTPS wurde von Netscape entwickelt und zusammen mit SSL 1.0 erstmals 1994 mit deren Browser veröffentlicht.

### 2.2.1 Nutzen

HTTPS wird zur Herstellung von Vertraulichkeit und Integrität in der Kommunikation zwischen Webserver und Webbrowser (Client) im World Wide Web verwendet. Dies wird u. a. durch Verschlüsselung und Authentifizierung erreicht.

Ohne Verschlüsselung sind Daten, die über das Internet übertragen werden, für jeden, der Zugang zum entsprechenden Netz hat, als Klartext lesbar. Mit der zunehmenden Verbreitung von offenen (d. h. unverschlüsselten) WLANs nimmt die Bedeutung von HTTPS zu, da damit die Inhalte unabhängig vom Netz verschlüsselt werden können.

Die Authentifizierung dient dazu, dass beide Seiten der Verbindung beim Aufbau der Kommunikation die Identität des Verbindungspartners überprüfen können. Dadurch sollen bspw. Phishing, aber auch Man-in-the-Middle-Angriffe verhindert werden.

### 2.2.2 Technik

Syntaktisch ist HTTPS identisch mit dem Schema für HTTP, die zusätzliche Verschlüsselung der Daten geschieht mittels SSL/TLS: Unter Verwendung des SSL-Handshake-Protokolls findet zunächst eine geschützte Identifikation und Authentifizierung der Kommunikationspartner statt. Anschließend wird mit Hilfe asymmetrischer Verschlüsselung oder des Diffie-Hellman-Schlüsselaustauschs ein gemeinsamer symmetrischer Sitzungsschlüssel ausgetauscht. Dieser wird schließlich zur Verschlüsselung der Nutzdaten verwendet.

Der Standard-Port für HTTPS-Verbindungen ist 443.

Neben den Server-Zertifikaten können auch signierte Client-Zertifikate nach X.509.3 erstellt werden. Das ermöglicht eine Authentifizierung der Clients gegenüber dem Server, wird jedoch selten eingesetzt.

Eine ältere Protokollvariante von HTTPS war S-HTTP.

### 2.2.3 Client-Verarbeitung

Mit der Entwicklung von HTTPS durch Netscape wurde das Protokoll und die anwenderseitige Client-Software schon früh in Webbrowser integriert. Damit ist meist keine weitere Installation gesonderter Software notwendig.

SSL Symbol.png

Eine HTTPS-Verbindung wird durch eine https-URL angewählt und durch das SSL-Logo angezeigt – beim Internet Explorer 6 ein Schloss-Icon in der Statusleiste, bei Mozilla zusätzlich in der Adresszeile, die bei Firefox, aktuellen Opera- und Internet-Explorer-7-Browsern zusätzlich gelb hinterlegt wird, bei Apple Safari 3.0 durch ein kleines Schloss-Symbol in der obersten rechten Ecke des Browserfensters.

### 2.2.4 Varianten der HTTPS-Anwahl

Die Entscheidung, ob eine sichere HTTPS- statt einer HTTP-Verbindung genutzt wird, kann unterschiedlich erfolgen:

Serverseitig wird ausschließlich HTTPS zugelassen, wie meist bei Online-Banking; teils wird dabei eine angewählte http-Adresse automatisch in https umgewandelt.

Der Login wird über HTTPS erzwungen, dann wird ein HTTP-Cookie im Browser gesetzt und, um Rechenzeit zu sparen, der weitere Dienst unverschlüsselt abgewickelt; z. B. bei SourceForge oder eBay.

Login per http-Adresse, die aber vom Anwender manuell in „https...“ geändert werden kann, um eine Verschlüsselung zu bewirken; z. B. bei GMX; teils auch über einen Link „Sicheres Login“ o. ä.

Clientseitiges Browser-Add-on (z. B. für Firefox und Chrome „HTTPS Everywhere“) welches http-Anfragen durch https-Anfragen ersetzt, falls der Server das Protokoll unterstützt. Z. B. würde der Zugriff auf „<http://de.wikipedia.org/wiki/Wikipedia:Hauptseite>“ automatisch umgeleitet auf „<https://de.wikipedia.org/wiki/Wikipedia:Hauptseite>“.

Nach Anwahl der HTTPS-Adresse soll der Client-Browser dem Anwender zuerst das Zertifikat anzeigen, sofern es nicht automatisch über bereits akzeptierte Zertifikate überprüft werden kann. Dieser entscheidet nun, gegebenenfalls nach Prüfung über die angegebenen Links, ob er dem Zertifikat für diese Sitzung vertraut, ggf. es auch permanent speichert. Andernfalls wird die HTTPS-Verbindung nicht hergestellt („Diese Seite verlassen“ bei Firefox bzw. „Klicken Sie hier um diese Seite zu verlassen.“ beim Internet Explorer).

### 2.2.5 Vorinstallierte Zertifikate

Um diese für Unkundige eventuell irritierende Abfrage zu vermeiden, wurde mit der Zeit eine Reihe von Root-Zertifikaten von den Browserherstellern akzeptiert, die schon bei der Installation eingetragen werden. Webseiten, die entsprechende Zertifikate haben, werden dann, ebenso wie davon abgeleitete Unter-Zertifikate, bei Aufruf ohne Nachfrage akzeptiert. Ob ein Root-Zertifikat dem Browser bekannt ist, hängt von der Browser-Version ab; zudem wird die Liste der Zertifikate teils auch online im Rahmen der Systemaktualisierung auf den neuesten Stand gebracht, so bei Microsoft Windows.

Mit dem Internet Explorer 7 hat Microsoft, kurz danach auch Mozilla mit dem Firefox 3, die Warnung bei nicht eingetragenen Zertifikaten verschärft: Erschien vorher nur ein Pop-up „Sicherheitshinweis“, das nach Name, Quelle und Laufzeit des Zertifikats differenzierte, so wird nun der Inhalt der Webseite ausgeblendet und eine Warnung angezeigt, mit der Empfehlung, die Seite nicht zu benutzen. Um diese sehen zu können, muss der Anwender dann explizit eine „Ausnahme hinzufügen“. Ein nicht im Browser eingetragenes Zertifikat wird damit für Massenanwendungen zunehmend untauglich.

Die Frage, welche Zertifikate in die Browser aufgenommen werden, hat in der Open-Source-Community fallweise zu längeren Diskussionen geführt, so zwischen CAcert, einem Anbieter kostenloser Zertifikate, und der Mozilla Foundation, siehe CAcert (Vertrauenswürdigkeit).

## 2.3 File Transfer Protocol

FTP (File Transfer Protocol)	
<b>Familie:</b>	Internetprotokollfamilie
<b>Einsatzgebiet:</b>	Datenübertragung, Dateiverwaltung
<b>Port:</b>	20/TCP DATA Port, 21/TCP Control Port
FTP im TCP/IP-Protokollstapel:	
<b>Anwendung</b>	<b>FTP</b>
<i>Transport</i>	TCP
<i>Internet</i>	IP (IPv4, IPv6)
<i>Netzzugang</i>	Ethernet Token Bus Token Ring FDDI ...
<b>Standards:</b>	RFC 354 <a href="#">↗</a> (1972)

Das File Transfer Protocol (FTP, englisch für Dateiübertragungsprotokoll) ist ein im RFC 959 von 1985 spezifiziertes Netzwerkprotokoll zur Übertragung von Dateien über IP-Netzwerke. FTP ist in der Anwendungsschicht (Schicht 7) des OSI-Schichtenmodells angesiedelt. Es wird benutzt, um Dateien vom Server zum Client (Herunterladen), vom Client zum Server (Hochladen) oder clientgesteuert zwischen zwei FTP-Servern zu übertragen (File Exchange Protocol). Außerdem können mit FTP Verzeichnisse angelegt und ausgelesen sowie Verzeichnisse und Dateien umbenannt oder gelöscht werden.

Das FTP verwendet für die Steuerung und Datenübertragung jeweils separate Verbindungen: Eine FTP-Sitzung beginnt, indem vom Client zum Control Port des Servers (der Standard-Port dafür ist Port 21) eine TCP-Verbindung aufgebaut wird. Über diese Verbindung werden Befehle zum Server gesendet. Der Server antwortet auf jeden Befehl mit einem Statuscode, oft mit einem angehängten, erklärenden Text. Die meisten Befehle sind allerdings erst nach einer erfolgreichen Authentifizierung zulässig.

### 2.3.1 Verbindungsarten

Zum Senden und Empfangen von Dateien sowie zur Übertragung von Verzeichnislisten (der Standard-Port dafür ist Port 21) wird pro Vorgang jeweils eine separate TCP-Verbindung verwendet. FTP kennt für den Aufbau solcher Verbindungen zwei Modi:

#### 2.3.2 Aktives FTP

Beim aktiven FTP (auch „Active Mode“) öffnet der Client einen zufälligen Port und teilt dem Server diesen sowie die eigene IP-Adresse mittels des PORT- oder des EPRT-Kommandos mit. Dies ist typischerweise ein Port des Clients, der jenseits von 1023 liegt, kann aber auch ein anderer Server sein, der seinerseits in den Passive Mode geschaltet wurde, also auf eine Verbindung wartet (so genanntes FXP). Die Datenübertragung auf der Server-Seite erfolgt dabei über Port 20. Die Kommunikation mit Befehlen erfolgt ausschließlich auf dem Control Port. Man spricht auch von der Steuerung „Out of Band“. Somit bleibt es möglich, dass während der Datenübertragung die Partner noch immer miteinander kommunizieren können.

#### 2.3.3 Passives FTP

Beim passiven FTP (auch „Passive Mode“) sendet der Client ein PASV- oder ein EPSV-Kommando, der Server öffnet einen Port und übermittelt diesen mitsamt IP-Adresse an den Client. Hier wird auf der Client-Seite ein Port jenseits 1023 verwendet und auf der Server-Seite der vorher an den Client übermittelte Port. Diese Technik wird eingesetzt, wenn der Server keine Verbindung zum Client aufbauen kann. Dies ist beispielsweise der Fall, wenn der Client sich hinter einem Router befindet, der die Adresse des Clients mittels NAT umschreibt, oder wenn eine Firewall das Netzwerk des Clients vor Zugriffen von außen abschirmt.

### 2.3.4 Öffentliche FTP-Server

Viele FTP-Server, vor allem Server von Universitäten, Fachhochschulen und Mirrors, bieten sogenanntes Anonymous FTP an. Solche FTP-Server werden auch als Pub (v. engl. public ‚öffentlich‘) bezeichnet. Hier ist zum Einloggen neben den realen Benutzerkonten ein spezielles Benutzerkonto, typischerweise „anonymous“ und/oder „ftp“, vorgesehen, für das kein (oder ein beliebiges) Passwort angegeben werden muss. Früher gehörte es zum „guten Ton“, bei anonymem FTP seine eigene, gültige E-Mail-Adresse als Passwort anzugeben. Die meisten Webbrowser tun dies heute nicht mehr, da es aus Spamschutz-Gründen nicht zu empfehlen ist.

### 2.3.5 FTP-Software

Für das Datenübertragungsverfahren wird ein FTP-Client benötigt. In vielen aktuellen Browsern ist ein FTP-Client meist bereits integriert. Ein Beispiel für die Syntax einer FTP-Adressierung im Browser ist:

[ftp://\[ftp\\_username\[:ftp\\_PWD\]@\]Servername\[:Port\]](ftp://[ftp_username[:ftp_PWD]@]Servername[:Port])

Der Client baut die TCP-Verbindung zum Control Port eines Servers auf. Über diese Verbindung wird über FTP-Kommandos der Datenaustausch zwischen Client und Server gesteuert. Davon zu unterscheiden sind die Kommandos für den zum Betriebssystem gehörenden Terminal-Client „ftp“, siehe auch FTP-Terminal-Client.

Daneben ist WebFTP ein von Webservern angebotener Dienst, der den Zugriff auf FTP-Server auch über HTTP ermöglicht. Die Darstellung erfolgt dabei innerhalb eines Webbrowsers. Eine Installation von Client-Software auf einem lokalen Rechner entfällt dadurch.

Eine Free/Libre Open Source Software zur Dateiübertragung mittels FTP ist FileZilla.

### 2.3.6 Sicherheit

Um Verschlüsselung und Authentifizierung zu nutzen, kann Transport Layer Security eingesetzt werden (FTP über SSL, kurz FTPS). Nach der Authentifizierung des Hosts und der Verschlüsselung durch TLS kann FTP die Authentifizierung des Client mittels Benutzername und Kennwort durchführen, wenn der Client sich nicht bereits mit einem Zertifikat über TLS authentifiziert hat.

Außerdem existiert mit dem SSH File Transfer Protocol (SFTP) eine auf SSH aufbauende Alternative zu FTP für Dateiverwaltung und -übertragung, bei dem nur der schon laufende sshd-Daemon genutzt und somit keine weitere Software auf Serverseite benötigt wird.



## 2.4 Simple Mail Transfer Protocol

SMTP (Simple Mail Transfer Protocol)	
<b>Familie:</b>	Internetprotokollfamilie
<b>Einsatzgebiet:</b>	Einspeisung von E-Mail (Mail Submission), Abholung von E-Mails eventuell über mehrere Stationen (Mail Transfer)
<b>Ports:</b>	25/TCP 465/TCP (Nur mit SSL/TLS und inzwischen veraltet - Siehe SMTPS) 587/TCP (Nur und bevorzugt für Einlieferung durch Mail-Clients)
SMTP im TCP/IP-Protokollstapel:	
<b>Anwendung</b>	<b>SMTP</b>
<i>Transport</i>	TCP
<i>Internet</i>	IP (IPv4, IPv6)
<i>Netzzugang</i>	Ethernet Token Bus Token Ring FDDI ...
<b>Standard:</b>	RFC 5321 <a href="#">↗</a>

Das Simple Mail Transfer Protocol (SMTP, zu deutsch etwa Einfaches E-Mail-Transportprotokoll) ist ein Protokoll der Internetprotokollfamilie, das zum Austausch von E-Mails in Computernetzen dient. Es wird dabei vorrangig zum Einspeisen und zum Weiterleiten von E-Mails verwendet. Zum Abholen von Nachrichten kommen andere, spezialisierte Protokolle wie POP3 oder IMAP zum Einsatz. SMTP-Server nehmen traditionell Verbindungen auf Port 25 („smtp“) entgegen.

Neuere Server benutzen auch Port 587, um ausschließlich von authentifizierten Benutzern Mails entgegenzunehmen („mail submission agent“). Durch eine klare Trennung eigener und fremder Benutzer sollen Konfigurationsprobleme und damit Spam vermieden werden (→ SMTP-Relay-Server). Außerdem kann aufgrund der unterschiedlichen Ports eine einfache Firewallregel verwendet werden, um unkontrolliert abgehende Spammnachrichten aus dem eigenen Netzwerk zu blockieren, ohne dass Verbindungen zu externen SMTP-Servern vollständig ausgeschlossen werden.

### 2.4.1 Geschichte

Vorgänger von SMTP waren im Arpanet das Mail Box Protocol (RFC 278) vom Juli 1971 und FTP Mail (RFC 458) vom Februar 1973. Mit der Entstehung des Internets aus dem ARPANET um 1980 schlug Jonathan Postel vor, die Abhängigkeit des E-Mail-Verkehrs vom FTP-Dienst abzukoppeln (RFC 772), und veröffentlichte 1982 SMTP unter RFC 821. In den frühen 1980er Jahren wurde es eine Ergänzung zu UUCP, das vor allem für den E-Mail-Verkehr periodisch verbundener Rechner genutzt wurde. SMTP wurde der Standard für Rechner, die ständig am Netz waren.

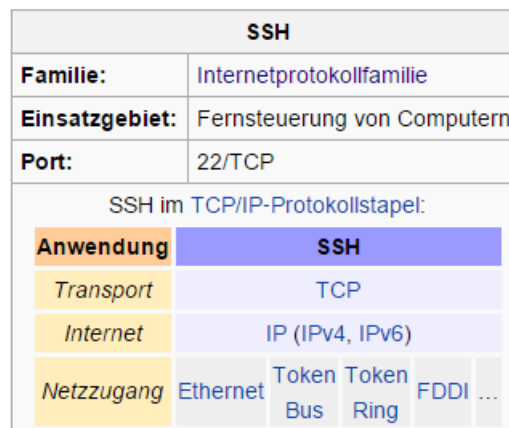
Einer der ersten Mail Transfer Agents, der SMTP implementierte und weitere Verbreitung erlangte, war sendmail. Inzwischen gibt es unzählige Programme, die SMTP als Client oder Server unterstützen, darunter weit verbreitete SMTP-Server wie Postfix, qmail, exim, usw. Da viele Programmierframeworks wie .NET oder Java bereits SMTP-Klassen eingebaut haben, ist die Entwicklung auch nur noch mit geringem Aufwand verbunden.

SMTP begann als reines ASCII-Protokoll, so dass damit keine Binärdateien übertragen werden konnten. Erst Standards wie MIME (Multipurpose Internet Mail Extensions) schufen diese Möglichkeit durch ein Kodieren der Binärdateien in ASCII.

## 2.4.2 Verfahren

Die Abwicklung des SMTP-Verfahrens wird meist für den Anwender unsichtbar durch sein Mailprogramm vorgenommen, den sogenannten Mail User Agent (MUA). Dieses Programm verbindet sich mit einem SMTP-Server, dem Mail Submission Agent (MSA), der die Mail über ggf. weitere SMTP-Server, sogenannte Mail Transfer Agents (MTA), zum Ziel transportiert. Da SMTP als Protokoll zum Transport von lokal erstellten Mails zwischen Servern konzipiert wurde, übernahm dabei ursprünglich ein einzelner Server auf Port 25 („smtp“) die Rolle von MSA und MTA. Der dedizierte Port 587 („submission“) für MSAs wurde erst 1998 eingeführt, um den unterschiedlichen Anforderungen beider Aufgaben gerecht zu werden: Ein MSA akzeptiert ausdrücklich nur Nachrichten berechtigter Nutzer und bereitet sie vor der Einspeisung in das Mailsystem gegebenenfalls standardkonform auf. Wegen der nahen Verwandtschaft beider Dienste wird die Funktionalität von MSA und MTA üblicherweise immer noch von nur einem Programm, das dann auf beiden Ports Verbindungen annimmt, bereitgestellt.

## 2.5 Secure Shell



Secure Shell oder SSH bezeichnet sowohl ein Netzwerkprotokoll als auch entsprechende Programme, mit deren Hilfe man auf eine sichere Art und Weise eine verschlüsselte Netzwerkverbindung mit einem entfernten Gerät herstellen kann. Häufig wird diese Methode verwendet, um lokal eine entfernte Kommandozeile verfügbar zu machen, das heißt, auf einer lokalen Konsole werden die Ausgaben der entfernten Konsole ausgegeben und die lokalen Tastatureingaben werden an den entfernten Rechner gesendet. Genutzt werden kann dies beispielsweise zur Fernwartung eines in einem entfernten Rechenzentrum stehenden Servers. Die neuere Protokoll-Version SSH-2 bietet weitere Funktionen wie Datenübertragung per SFTP.

Die IANA hat dem Protokoll den TCP-Port 22 zugeordnet.

### 2.5.1 Geschichte

Die erste Version des Protokolls (jetzt SSH-1 genannt) wurde 1995 von Tatu Ylönen als Reaktion auf die Nachfrage nach drop-in replacements für Berkeley Services unter Unix einschließlich der Befehle rsh (remote shell), rcp (remote copy) und rlogin (remote login) entwickelt. Er veröffentlichte seine Implementierung 1995 als Freeware, die daraufhin schnell an Popularität gewann; Ende des Jahres 1995 zählte man bereits 20.000 Benutzer in fünfzig Ländern.

Im Dezember gründete Tatu Ylönen die Firma SSH Communications Security, um SSH zu vermarkten und weiterzuentwickeln. Die Originalsoftware enthielt ursprünglich Open-Source-Quellcode, entwickelte sich aber im Laufe der Zeit immer mehr zu proprietärer Software.

Nachdem einige Schwachstellen in der Integritätsprüfung von SSH-1 bekannt geworden waren, wurde 1996 mit SSH-2 eine überarbeitete Version des Protokolls entwickelt. Sie ist inkompatibel zu SSH-1. Dabei wurde unter anderem das Protokoll in verschiedene Einzelteile aufgegliedert und somit die Verwendung sicherer Verschlüsselungs- und Authentifikations-Algorithmen ermöglicht. Damit wurde die Schwachstelle beseitigt. Derzeit gilt das Protokoll als sicher.

1999 wurde der Wunsch nach einer freien Implementierung von SSH laut, und aus der letzten freien Version der Originalimplementierung entwickelte sich das separate OpenSSH-Projekt. Spätestens seit dieser Zeit existiert das SSH-Protokoll in zwei Implementierungen: Als Open-Source-Software (OpenSSH) und als proprietäre Software

(Produktname: SSH Tectia), entwickelt und vertrieben von der Firma SSH Communications Security, also den Original-Entwicklern rund um Ylönen.

2005, also zehn Jahre nach der Original-Entwicklung, ging die Firma SSH Communications Security mit der Generation 3 (SSH G3) an die Öffentlichkeit. Diese Protokollversion unterstützt die Verwendung des umstrittenen proprietären Algorithmus CryptiCore. Die anderen, etablierten Verschlüsselungsalgorithmen werden weiterhin unterstützt. 2006 wurde dieses Protokoll (Version 2) von der IETF als Internetstandard vorgeschlagen. Eine Zertifizierung nach FIPS-Standard 140-2 besteht bereits länger.

### **2.5.2 Verwendung**

Eine X11-Verbindung wird über SSH weitergeleitet

SSH ermöglicht eine sichere, authentifizierte und verschlüsselte Verbindung zwischen zwei Rechnern über ein unsicheres Netzwerk. Dadurch dient es unter anderem als Ersatz für die Vorgänger rlogin, telnet und rsh; diese übertragen jeglichen Netzverkehr, darunter auch die Passwörter, unverschlüsselt.

## 3 Webserver

### 3.1 Apache HTTP Server

Der Apache HTTP Server [əˈpætʃi] ist ein quelloffenes und freies Produkt der Apache Software Foundation und der meistbenutzte Webserver im Internet.

#### Geschichte

Eine Gruppe von acht Entwicklern begann 1994 den Webserver NCSA HTTPd zu erweitern. Dies waren im Einzelnen: Brian Behlendorf, Roy T. Fielding, Rob Hartill, David Robinson, Cliff Skolnick, Randy Terbush, Robert S. Thau und Andrew Wilson mit Unterstützung von Eric Hagberg, Frank Peters und Nicolas Pioch.

Sie gaben dem Ergebnis ihrer Arbeit den Namen Apache HTTP Server und veröffentlichten diesen im April 1995. Er war das Gründungsprojekt der Apache Software Foundation.

#### Eigenschaften und Funktionen

Neben Unix und Linux unterstützt Apache Win32, NetWare sowie eine Vielzahl weiterer Betriebssysteme. In Version 2.0 wurde die Stabilität und Geschwindigkeit des Servers – vor allem auf Nicht-Unix-Systemen – erheblich verbessert: Die Bibliothek Apache Portable Runtime (APR) stellt eine Verallgemeinerung wichtiger Systemaufrufe zur Verfügung, sodass die individuellen Stärken des jeweiligen Betriebssystems ausgenutzt werden können. Hinzu kommen verschiedene Multiprocessing-Module (MPM), die je nach Plattform unterschiedliche Lösungen für die gleichzeitige Bedienung mehrerer Client-Anfragen anbieten: Beispielsweise setzt das MPM prefork für klassische Unix-Systeme auf Forking von Prozessen, während mpm\_winnt für die unter Windows empfehlenswerteren Threads optimiert ist.

Der Apache-Webserver ist modular aufgebaut: Durch entsprechende Module kann er beispielsweise die Kommunikation zwischen Browser und Webserver verschlüsseln (mod\_ssl), als Proxyserver eingesetzt werden (mod\_proxy) oder komplexe Manipulationen von HTTP-Kopfdaten (mod\_headers) und URLs (mod\_rewrite) durchführen.

Der Apache bietet die Möglichkeit, mittels serverseitiger Skriptsprachen Webseiten dynamisch zu erstellen. Häufig verwendete Skriptsprachen sind PHP, Perl oder Ruby. Weitere Sprachen sind Python, JavaScript (z. B. V8CGI), Lua, Tcl und .NET (mit ASP.NET oder Mono). Diese sind kein Bestandteil des Webserver, sondern müssen ebenfalls entweder als Module eingebunden werden oder über das CGI angesprochen werden. Über das bei der Apache-Installation enthaltene mod\_include kann Server Side Includes (SSI) ausgeführt werden. Damit ist es möglich, einfache dynamische Webseiten zu erstellen und den Verwaltungsaufwand von statischen Webseiten zu minimieren.

Der Apache HTTP Server ist, wie alle Programme der Apache Software Foundation, eine freie Software. Derzeit wird noch die stabile Version 2.2.x unterstützt und somit beispielsweise mit Sicherheitsupdates versorgt. Die Apache-Entwickler empfehlen die Version 2.4.x für den Produktiveinsatz.

#### Namensherkunft

Der Name wurde aus Respekt vor dem nordamerikanischen Indianerstamm der Apachen gewählt. Nicht korrekt ist, dass der Name eine Umdeutung von „a patchy server“ sei, was so viel wie „ein zusammengeflackter Server“ bedeutet. Diese Deutung entstand durch den Umstand, dass der Apache HTTP Server ursprünglich eine gepatchte Erweiterung des alten NCSA HTTP Servers war.

#### Distributionen

Der Apache HTTP Server ist in fast allen Linux-Distributionen und in Mac OS X standardmäßig enthalten. Eine beliebte Entwicklungs-Distribution für Windows, Linux und Mac OS X ist XAMPP.

## 3.2 Internet Information Services (IIS)

Internet Information Services (IIS) (vormals Internet Information Server) ist eine Dienstplattform des Unternehmens Microsoft für PCs und Server. Über sie können Dokumente und Dateien im Netzwerk zugänglich gemacht werden. Als Kommunikationsprotokolle kommen hierbei HTTP, HTTPS, FTP, SMTP, POP3, WebDAV und andere zum Einsatz. Über IIS können ASP- oder .NET-Applikationen (ASP.NET) ausgeführt werden, sowie – mit den passenden installierbaren ISAPI-Filtern – auch PHP und JSP.

### 3.2.1 Betriebssysteme

IIS-Dienste können auf folgenden Microsoft-eigenen Betriebssystemen eingesetzt werden: Windows NT Server, Windows 2000 Server, Microsoft Windows Server 2003, Windows Server 2008 und Windows Server 2012. Die aktuelle Version ist IIS 8.5 (Stand 10/2013).

Bei Windows 2000 Professional und Windows XP Professional werden eingeschränkte IIS-Dienste in den jeweiligen Versionen als optionale Komponente mitgeliefert. Hier ist die Anzahl gleichzeitiger Verbindungen auf höchstens 10 beschränkt und es kann nur eine Website (die „Standardwebsite“) eingerichtet werden. Vorgesehen ist der Einsatz als reine Test- und Entwicklungsumgebung. Diese IIS Express-Version ist auch in allen Varianten von Visual Studio 2012 enthalten.

IIS 7.0 ist nicht nur in Windows Server 2008 enthalten, sondern auch in den Business-, Enterprise- und Ultimate-Versionen von Windows Vista. Eine eingeschränkte Version (maximal 3 Verbindungen) ist in Vista Home Premium enthalten.

### 3.2.2 Versionen

IIS 1.0 gab es als Download für Windows NT 3.51.

IIS 2.0 ist in Windows NT 4.0 integriert.

IIS 3.0 wurde mit Service Pack 2 unter Windows NT 4.0 installiert. Es war die erste Version mit den Active Server Pages.

IIS 4.0 wurde als Download für Windows NT 4.0 zusammen mit dem Option Pack angeboten. Der Option Pack bot viele weitere Technologien wie den Microsoft Transaction Server, den Microsoft Index Server, den Certificate Server und Site Server Express. Mit Ausnahme des letzten sind diese ganzen Produkte direkt in IIS 5.0 unter Windows 2000 eingeflossen.

IIS 5.0 ist in Windows 2000 integriert. Ab sofort heißt das Produkt „Microsoft Internet Information Services“ (statt „Microsoft Internet Information Server“).

IIS 5.1 ist in Windows XP Professional und Windows MCE integriert. (32-Bit-Versionen)

IIS 6.0 ist in Windows Server 2003 und in Windows XP x64 Edition integriert. Er wurde von Grund auf neu konzipiert, um Sicherheitsprobleme zu lösen. So muss jetzt jede Web Server Extension wie ASP, ASP .NET, Internet Printing, Server Side Includes (SSI) etc. explizit angeschaltet werden, damit nicht benutzte Features keine potentiellen Sicherheitslücken anbieten.

IIS 7.0 ist in Windows Vista und in Windows Server 2008 integriert.

IIS 7.5 ist in Windows Server 2008 R2 und in Windows 7 integriert

IIS 8.0 ist in Windows Server 2012 und in Windows 8 integriert.

IIS 8.5 ist in Windows Server 2012 R2 und in Windows 8.1 integriert.

### 3.3 XAMPP



XAMPP ist eine Zusammenstellung von freier Software – vorwiegend im Umfeld des LAMP-Systems. XAMPP ermöglicht das einfache Installieren und Konfigurieren des Webserver Apache mit der Datenbank MySQL bzw. SQLite und den Skriptsprachen Perl und PHP (mit PEAR). Das X steht hierbei für die verschiedenen Betriebssysteme, auf denen es eingesetzt werden kann. XAMPP enthält zusätzlich andere nützliche Werkzeuge wie den FTP-Server ProFTPD oder FileZilla Server, den Mailserver Mercury, phpMyAdmin, Webalizer und OpenSSL. Seit Version 1.7.4 beinhaltet die Windows-Variante zusätzlich auch Apache Tomcat 7, der die Ausführung von JavaServer Pages und Java Servlets ermöglicht.

#### 3.3.1 Eigenschaften und Funktionen

Ziel von XAMPP ist es, eine besonders einfache Installation zu erreichen. Für Windows-Systeme gibt es eine Version mit Installationsroutine, für die anderen unterstützten Betriebssysteme (siehe unten) eine Version mit ausführlicher Installationsanleitung. Mit wenigen Mausklicks erhält man Server-Werkzeuge, die alleine teilweise recht lange Konfigurationszeiten benötigen würden.

XAMPP ist nicht für den Einsatz als Produktivsystem (z. B. als öffentlicher Webserver) gedacht, sondern für Entwickler, die möglichst schnell ein kompaktes Testsystem aufsetzen möchten. Dies erklärt auch die bewusst in Kauf genommenen Einschränkungen in Hinblick auf die Sicherheit von XAMPP. Die Missachtung dieser Warnung führte z. B. zum Patras-Hack bei der Bundespolizei im Juli 2011.

Unter Linux installiert sich XAMPP komplett in das Verzeichnis /opt. Da die meisten Distributionen Apache, PHP und MySQL standardmäßig installieren (z. B. nach /usr/bin oder /usr/sbin), ist es ratsam, diese Pakete vor einer Installation von XAMPP komplett zu deinstallieren, um Konfusionen zu vermeiden.

#### 3.3.2 Varianten

XAMPP ist für folgende Betriebssysteme erhältlich:

Linux (früher als LAMP bekannt)

Solaris – noch im Beta-Entwicklungsstadium

Mac OS X

Windows (Windows 2000 und höher - früher als WAMPP bekannt)

Ursprünglich wurden die jeweiligen Versionen in Abhängigkeit zum zugedachten Betriebssystem LAMPP (Linux), MAMPP (MAC) bzw. WAMPP (Windows) genannt. Aufgrund der möglichen Begriffsverwirrung und der Etablierung eines einheitlichen Produktnamens werden alle Pakete zukünftig nur noch als XAMPP bezeichnet. Die Umstellung der Software erfolgt aus Stabilitätsgründen sukzessive.

### 3.3.3 Lizenz

XAMPP an sich ist unter der GNU General Public License freigegeben, ebenso der größte Teil der mitgelieferten Software. In Abhängigkeit zur verwendeten Betriebssystem-Version ist jedoch auch Software enthalten, die anderen Lizenzen unterliegt.

## 3.4 Microsoft WebMatrix



Microsoft WebMatrix ist ein von Microsoft zur Verfügung gestelltes Entwicklungswerkzeug für Webseiten auf unterschiedlichen Plattformen und Frameworks. Der Benutzer hat die Möglichkeit, seine Webseiten von Grund auf neu zu entwickeln, existierende Webseiten zu bearbeiten, vorgefertigte Templates zu nutzen oder über die Microsoft Web App Gallery diverse kostenlose Open-Source-Frameworks wie WordPress, Joomla, Drupal, DotNetNuke zu nutzen. Die Installation und Verwaltung dieser diversen Content-Management-, Blog- und E-Shop-Systeme funktioniert dabei komplett innerhalb von WebMatrix. WebMatrix umfasst IIS Developer Express (einen Entwicklungs-Webserver), ASP.NET (ein Web-Framework) und Microsoft SQL Server Compact (eine eingebettete Datenbank).

Unterstützte Frameworks und Sprachen

In Version 1.0 unterstützt WebMatrix folgende Frameworks:

- Acquia Drupal
- AtomSite
- Azopho
- BlogEngine.NET
- Composite C1
- Core Ecommerce
- dasBlog
- DotNetNuke
- Droptings
- Gallery
- Gallery Server Pro
- Joomla
- Kartris
- Kentico CMS

Kooboo  
Mayando  
MODx  
mojoPortal  
MonoX  
Moodle  
MY Web Pages  
N2 CMS  
nopCommerce  
nService  
phpBB  
Piwik  
Resource Blender  
ScewTurn Wiki  
SilverStripe  
Sitefinity  
Subtext  
SugarCRM  
TangoCMS  
Tiki  
Umbraco  
WordPress  
YetAnotherForum

Mit Webmatrix können Webseiten mit Hilfe von HTML, CSS, JavaScript, PHP und ASP.NET erstellt werden. Zusätzlich wird die sogenannte Razor-Syntax für ASP.NET Webseiten unterstützt.

### **3.4.1 Features**

Webprojekte, die mit WebMatrix erstellt wurden, können direkt in Microsofts Entwicklungsumgebung Visual Studio bearbeitet werden.

Das Programm beinhaltet eine kleine, eingebettete Datenbank namens SQL Server Compact. Für die Verwendung ist keine zusätzliche Installation auf dem Webserver erforderlich. Die Migration auf Microsoft SQL Server wird unterstützt.

WebMatrix liefert eine Option für automatische SEO-Berichte mit. Dieser bietet Hinweise dazu, wie Webseiten besser gestaltet werden können. Es erlaubt das Publizieren von Webseiten mit Hilfe von FTP, FTPS und WebDeploy.